

LNCS 15303

Apostolos Antonacopoulos ·
Subhasis Chaudhuri · Rama Chellappa ·
Cheng-Lin Liu · Saumik Bhattacharya ·
Umapada Pal (Eds.)

Pattern Recognition

27th International Conference, ICPR 2024
Kolkata, India, December 1–5, 2024
Proceedings, Part III

3
Part III

ICPR
2024
INDIA



 Springer

MOREMEDIA 

Lecture Notes in Computer Science

15303


Founding Editors

Gerhard Goos
Juris Hartmanis

Editorial Board Members

Elisa Bertino, *Purdue University, West Lafayette, IN, USA*

Wen Gao, *Peking University, Beijing, China*

Bernhard Steffen , *TU Dortmund University, Dortmund, Germany*

Moti Yung , *Columbia University, New York, NY, USA*

The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), has established itself as a medium for the publication of new developments in computer science and information technology research, teaching, and education.


LNCS enjoys close cooperation with the computer science R & D community, the series counts many renowned academics among its volume editors and paper authors, and collaborates with prestigious societies. Its mission is to serve this international community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings. LNCS commenced publication in 1973.


Apostolos Antonacopoulos ·
Subhasis Chaudhuri · Rama Chellappa ·
Cheng-Lin Liu · Saumik Bhattacharya ·
Umapada Pal
Editors


Pattern Recognition

27th International Conference, ICPR 2024
Kolkata, India, December 1–5, 2024
Proceedings, Part III

Editors

Apostolos Antonacopoulos 
University of Salford
Salford, UK

Rama Chellappa 
Johns Hopkins University
Baltimore, MD, USA

Saumik Bhattacharya 
IIT Kharagpur
Kharagpur, India

Subhasis Chaudhuri 
Indian Institute of Technology Bombay
Mumbai, India

Cheng-Lin Liu 
Chinese Academy of Sciences
Beijing, China

Umapada Pal 
Indian Statistical Institute Kolkata
Kolkata, India

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-3-031-78121-6

ISBN 978-3-031-78122-3 (eBook)

<https://doi.org/10.1007/978-3-031-78122-3>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

President's Address

On behalf of the Executive Committee of the International Association for Pattern Recognition (IAPR), I am pleased to welcome you to the 27th International Conference on Pattern Recognition (ICPR 2024), the main scientific event of the IAPR.

After a completely digital ICPR in the middle of the COVID pandemic and the first hybrid version in 2022, we can now enjoy a fully back-to-normal ICPR this year. I look forward to hearing inspirational talks and keynotes, catching up with colleagues during the breaks and making new contacts in an informal way. At the same time, the conference landscape has changed. Hybrid meetings have made their entrance and will continue. It is exciting to experience how this will influence the conference. Planning for a major event like ICPR must take place over a period of several years. This means many decisions had to be made under a cloud of uncertainty, adding to the already large effort needed to produce a successful conference. It is with enormous gratitude, then, that we must thank the team of organizers for their hard work, flexibility, and creativity in organizing this ICPR. ICPR always provides a wonderful opportunity for the community to gather together. I can think of no better location than Kolkata to renew the bonds of our international research community.

Each ICPR is a bit different owing to the vision of its organizing committee. For 2024, the conference has six different tracks reflecting major themes in pattern recognition: Artificial Intelligence, Pattern Recognition and Machine Learning; Computer and Robot Vision; Image, Speech, Signal and Video Processing; Biometrics and Human Computer Interaction; Document Analysis and Recognition; and Biomedical Imaging and Bioinformatics. This reflects the richness of our field. ICPR 2024 also features two dozen workshops, seven tutorials, and 15 competitions; there is something for everyone. Many thanks to those who are leading these activities, which together add significant value to attending ICPR, whether in person or virtually. Because it is important for ICPR to be as accessible as possible to colleagues from all around the world, we are pleased that the IAPR, working with the ICPR organizers, is continuing our practice of awarding travel stipends to a number of early-career authors who demonstrate financial need. Last but not least, we are thankful to the Springer LNCS team for their effort to publish these proceedings.

Among the presentations from distinguished keynote speakers, we are looking forward to the three IAPR Prize Lectures at ICPR 2024. This year we honor the achievements of Tin Kam Ho (IBM Research) with the IAPR's most prestigious King-Sun Fu Prize "for pioneering contributions to multi-classifier systems, random decision forests, and data complexity analysis". The King-Sun Fu Prize is given in recognition of an outstanding technical contribution to the field of pattern recognition. It honors the memory of Professor King-Sun Fu who was instrumental in the founding of IAPR, served as its first president, and is widely recognized for his extensive contributions to the field of pattern recognition.

The Maria Petrou Prize is given to a living female scientist/engineer who has made substantial contributions to the field of Pattern Recognition and whose past contributions, current research activity and future potential may be regarded as a model to both aspiring and established researchers. It honours the memory of Professor Maria Petrou as a scientist of the first rank, and particularly her role as a pioneer for women researchers. This year, the Maria Petrou Prize is given to Guoying Zhao (University of Oulu), “for contributions to video analysis for facial micro-behavior recognition and remote bio-signal reading (RPPG) for heart rate analysis and face anti-spoofing”.

The J.K. Aggarwal Prize is given to a young scientist who has brought a substantial contribution to a field that is relevant to the IAPR community and whose research work has had a major impact on the field. Professor Aggarwal is widely recognized for his extensive contributions to the field of pattern recognition and for his participation in IAPR's activities. This year, the J.K. Aggarwal Prize goes to Xiaolong Wang (UC San Diego) “for groundbreaking contributions to advancing visual representation learning, utilizing self-supervised and attention-based models to establish fundamental frameworks for creating versatile, general-purpose pattern recognition systems”.

During the conference we will also recognize 21 new IAPR Fellows selected from a field of very strong candidates. In addition, a number of Best Scientific Paper and Best Student Paper awards will be presented, along with the Best Industry Related Paper Award and the Piero Zamperoni Best Student Paper Award. Congratulations to the recipients of these very well-deserved awards!

I would like to close by again thanking everyone involved in making ICPR 2024 a tremendous success; your hard work is deeply appreciated. These thanks extend to all who chaired the various aspects of the conference and the associated workshops, my ExCo colleagues, and the IAPR Standing and Technical Committees. Linda O’Gorman, the IAPR Secretariat, deserves special recognition for her experience, historical perspective, and attention to detail when it comes to supporting many of the IAPR’s most important activities. Her tasks became so numerous that she recently got support from Carolyn Buckley (layout, newsletter), Ugur Halici (ICPR matters), and Rosemary Stramka (secretariat). The IAPR website got a completely new design. Ed Sobczak has taken care of our web presence for so many years already. A big thank you to all of you!

This is, of course, the 27th ICPR conference. Knowing that ICPR is organized every two years, and that the first conference in the series (1973!) pre-dated the formal founding of the IAPR by a few years, it is also exciting to consider that we are celebrating over 50 years of ICPR and at the same time approaching the official IAPR 50th anniversary in 2028: you’ll get all information you need at ICPR 2024. In the meantime, I offer my thanks and my best wishes to all who are involved in supporting the IAPR throughout the world.

September 2024

Arjan Kuijper
President of the IAPR

Preface

It is our great pleasure to welcome you to the proceedings of the 27th International Conference on Pattern Recognition (ICPR 2024), held in Kolkata, India. The city, formerly known as ‘Calcutta’, is the home of the fabled Indian Statistical Institute (ISI), which has been at the forefront of statistical pattern recognition for almost a century. Concepts like the Mahalanobis distance, Bhattacharyya bound, Cramer–Rao bound, and Fisher–Rao metric were invented by pioneers associated with ISI. The first ICPR (called IJCPD then) was held in 1973, and the second in 1974. Subsequently, ICPR has been held every other year. The International Association for Pattern Recognition (IAPR) was founded in 1978 and became the sponsor of the ICPR series. Over the past 50 years, ICPR has attracted huge numbers of scientists, engineers and students from all over the world and contributed to advancing research, development and applications in pattern recognition technology.

ICPR 2024 was held at the Biswa Bangla Convention Centre, one of the largest such facilities in South Asia, situated just 7 kilometers from Kolkata Airport (CCU). According to ChatGPT “Kolkata is often called the ‘Cultural Capital of India’. The city has a deep connection to literature, music, theater, and art. It was home to Nobel laureate Rabindranath Tagore, and the Bengali film industry has produced globally renowned filmmakers like Satyajit Ray. The city boasts remarkable colonial architecture, with landmarks like Victoria Memorial, Howrah Bridge, and the Indian Museum (the oldest and largest museum in India). Kolkata’s streets are dotted with old mansions and buildings that tell stories of its colonial past. Walking through the city can feel like stepping back into a different era. Finally, Kolkata is also known for its street food.”

ICPR 2024 followed a two-round paper submission format. We received a total of 2135 papers (1501 papers in round-1 submissions, and 634 papers in round-2 submissions). Each paper, on average, received 2.84 reviews, in single-blind mode. For the first-round papers we had a rebuttal option available to authors.

In total, 945 papers (669 from round-1 and 276 from round-2) were accepted for presentation, resulting in an acceptance rate of 44.26%, which is consistent with previous ICPR events. At ICPR 2024 the papers were categorized into six tracks: Artificial Intelligence, Machine Learning for Pattern Analysis; Computer Vision and Robotic Perception; Image, Video, Speech, and Signal Analysis; Biometrics and Human-Machine Interaction; Document and Media Analysis; and Biomedical Image Analysis and Informatics.

The main conference ran over December 2–5, 2024. The main program included the presentation of 188 oral papers (19.89% of the accepted papers), 757 poster papers and 12 competition papers (out of 15 submitted). A total 10 oral sessions were held concurrently in four meeting rooms with a total of 40 oral sessions. In total 24 workshops and 7 tutorials were held on December 1, 2024.

The plenary sessions included three prize lectures and three invited presentations. The prize lectures were delivered by Tin Kam Ho (IBM Research, USA; King Sun

Fu Prize winner), Xiaolong Wang (University of California, San Diego, USA; J.K. Aggarwal Prize winner), and Guoying Zhao (University of Oulu, Finland; Maria Petrou Prize winner). The invited speakers were Timothy Hospedales (University of Edinburgh, UK), Venu Govindaraju (University at Buffalo, USA), and Shuicheng Yan (Skywork AI, Singapore).

Several best paper awards were presented in ICPR: the Piero Zamperoni Award for the best paper authored by a student, the BIRPA Best Industry Related Paper Award, and the Best Paper Awards and Best Student Paper Awards for each of the six tracks of ICPR 2024.

The organization of such a large conference would not be possible without the help of many volunteers. Our special gratitude goes to the Program Chairs (Apostolos Antonacopoulos, Subhasis Chaudhuri, Rama Chellappa and Cheng-Lin Liu), for their leadership in organizing the program. Thanks to our Publication Chairs (Ananda S. Chowdhury and Wataru Ohyama) for handling the overwhelming workload of publishing the conference proceedings. We also thank our Competition Chairs (Richard Zanibbi, Lianwen Jin and Laurence Likforman-Sulem) for arranging 12 important competitions as part of ICPR 2024. We are thankful to our Workshop Chairs (P. Shivakumara, Stephanie Schuckers, Jean-Marc Ogier and Prabir Bhattacharya) and Tutorial Chairs (B.B. Chaudhuri, Michael R. Jenkin and Guoying Zhao) for arranging the workshops and tutorials on emerging topics. ICPR 2024, for the first time, held a Doctoral Consortium. We would like to thank our Doctoral Consortium Chairs (Véronique Eglin, Dan Lopresti and Mayank Vatsa) for organizing it.

Thanks go to the Track Chairs and the meta reviewers who devoted significant time to the review process and preparation of the program. We also sincerely thank the reviewers who provided valuable feedback to the authors.

Finally, we acknowledge the work of other conference committee members, like the Organizing Chairs and Organizing Committee Members, Finance Chairs, Award Chair, Sponsorship Chairs, and Exhibition and Demonstration Chairs, Visa Chair, Publicity Chairs, and Women in ICPR Chairs, whose efforts made this event successful. We also thank our event manager Alpcord Network for their help.

We hope that all the participants found the technical program informative and enjoyed the sights, culture and cuisine of Kolkata.

October 2024

Umapada Pal
Josef Kittler
Anil Jain

Organization

General Chairs

Umapada Pal
Josef Kittler
Anil Jain

Indian Statistical Institute, Kolkata, India
University of Surrey, UK
Michigan State University, USA

Program Chairs

Apostolos Antonacopoulos
Subhasis Chaudhuri
Rama Chellappa
Cheng-Lin Liu

University of Salford, UK
Indian Institute of Technology, Bombay, India
Johns Hopkins University, USA
Institute of Automation, Chinese Academy of
Sciences, China

Publication Chairs

Ananda S. Chowdhury
Wataru Ohyama

Jadavpur University, India
Tokyo Denki University, Japan

Competition Chairs

Richard Zanibbi
Lianwen Jin
Laurence Likforman-Sulem

Rochester Institute of Technology, USA
South China University of Technology, China
Télécom Paris, France

Workshop Chairs

P. Shivakumara
Stephanie Schuckers
Jean-Marc Ogier
Prabir Bhattacharya

University of Salford, UK
Clarkson University, USA
Université de la Rochelle, France
Concordia University, Canada

Tutorial Chairs

B. B. Chaudhuri	Indian Statistical Institute, Kolkata, India
Michael R. Jenkin	York University, Canada
Guoying Zhao	University of Oulu, Finland

Doctoral Consortium Chairs

Véronique Eglin	CNRS, France
Daniel P. Lopresti	Lehigh University, USA
Mayank Vatsa	Indian Institute of Technology, Jodhpur, India

Organizing Chairs

Saumik Bhattacharya	Indian Institute of Technology, Kharagpur, India
Palash Ghosal	Sikkim Manipal University, India

Organizing Committee

Santanu Phadikar	West Bengal University of Technology, India
SK Md Obaidullah	Aliah University, India
Sayantari Ghosh	National Institute of Technology Durgapur, India
Himadri Mukherjee	West Bengal State University, India
Nilamadhaba Tripathy	Clarivate Analytics, USA
Chayan Halder	West Bengal State University, India
Shibaprasad Sen	Techno Main Salt Lake, India

Finance Chairs

Kaushik Roy	West Bengal State University, India
Michael Blumenstein	University of Technology Sydney, Australia

Awards Committee Chair

Arpan Pal	Tata Consultancy Services, India
-----------	----------------------------------

Sponsorship Chairs

P. J. Narayanan	Indian Institute of Technology, Hyderabad, India
Yasushi Yagi	Osaka University, Japan
Venu Govindaraju	University at Buffalo, USA
Alberto Bel Bimbo	Università di Firenze, Italy

Exhibition and Demonstration Chairs

Arjun Jain	FastCode AI, India
Agnimitra Biswas	National Institute of Technology, Silchar, India

International Liaison, Visa Chair

Balasubramanian Raman	Indian Institute of Technology, Roorkee, India
-----------------------	------------------------------------------------

Publicity Chairs

Dipti Prasad Mukherjee	Indian Statistical Institute, Kolkata, India
Bob Fisher	University of Edinburgh, UK
Xiaojun Wu	Jiangnan University, China

Women in ICPR Chairs

Ingela Nystrom	Uppsala University, Sweden
Alexandra B. Albu	University of Victoria, Canada
Jing Dong	Institute of Automation, Chinese Academy of Sciences, China
Sarbani Palit	Indian Statistical Institute, Kolkata, India

Event Manager

Alpcord Network

Track Chairs – Artificial Intelligence, Machine Learning for Pattern Analysis

Larry O’Gorman	Nokia Bell Labs, USA
Dacheng Tao	University of Sydney, Australia
Petia Radeva	University of Barcelona, Spain
Susmita Mitra	Indian Statistical Institute, Kolkata, India
Jiliang Tang	Michigan State University, USA

Track Chairs – Computer and Robot Vision

C. V. Jawahar	International Institute of Information Technology (IIIT), Hyderabad, India
João Paulo Papa	São Paulo State University, Brazil
Maja Pantic	Imperial College London, UK
Gang Hua	Dolby Laboratories, USA
Junwei Han	Northwestern Polytechnical University, China

Track Chairs – Image, Speech, Signal and Video Processing

P. K. Biswas	Indian Institute of Technology, Kharagpur, India
Shang-Hong Lai	National Tsing Hua University, Taiwan
Hugo Jair Escalante	INAOE, CINVESTAV, Mexico
Sergio Escalera	Universitat de Barcelona, Spain
Prem Natarajan	University of Southern California, USA

Track Chairs – Biometrics and Human Computer Interaction

Richa Singh	Indian Institute of Technology, Jodhpur, India
Massimo Tistarelli	University of Sassari, Italy
Vishal Patel	Johns Hopkins University, USA
Wei-Shi Zheng	Sun Yat-sen University, China
Jian Wang	Snap, USA

Track Chairs – Document Analysis and Recognition

Xiang Bai	Huazhong University of Science and Technology, China
David Doermann	University at Buffalo, USA
Josep Lladós	Universitat Autònoma de Barcelona, Spain
Mita Nasipuri	Jadavpur University, India

Track Chairs – Biomedical Imaging and Bioinformatics

Jayanta Mukhopadhyay	Indian Institute of Technology, Kharagpur, India
Xiaoyi Jiang	Universität Münster, Germany
Seong-Whan Lee	Korea University, Korea

Metareviewers (Conference Papers and Competition Papers)

Wael Abd-Almageed	University of Southern California, USA
Maya Aghaei	NHL Stenden University, Netherlands
Alireza Alaei	Southern Cross University, Australia
Rajagopalan N. Ambasmudram	Indian Institute of Technology, Madras, India
Suyash P. Awate	Indian Institute of Technology, Bombay, India
Inci M. Baytas	Bogazici University, Turkey
Aparna Bharati	Lehigh University, USA
Brojeshwar Bhowmick	Tata Consultancy Services, India
Jean-Christophe Burie	University of La Rochelle, France
Gustavo Carneiro	University of Surrey, UK
Chee Seng Chan	Universiti Malaya, Malaysia
Sumohana S. Channappayya	Indian Institute of Technology, Hyderabad, India
Dongdong Chen	Microsoft, USA
Shengyong Chen	Tianjin University of Technology, China
Jun Cheng	Institute for Infocomm Research, A*STAR, Singapore
Albert Clapés	University of Barcelona, Spain
Oscar Dalmau	Center for Research in Mathematics, Mexico

Tyler Derr	Vanderbilt University, USA
Abhinav Dhall	Indian Institute of Technology, Ropar, India
Bo Du	Wuhan University, China
Yuxuan Du	University of Sydney, Australia
Ayman S. El-Baz	University of Louisville, USA
Francisco Escolano	University of Alicante, Spain
Siamac Fazli	Nazarbayev University, Kazakhstan
Jianjiang Feng	Tsinghua University, China
Gernot A. Fink	TU Dortmund University, Germany
Alicia Fornes	CVC, Spain
Junbin Gao	University of Sydney, Australia
Yan Gao	Amazon, USA
Yongsheng Gao	Griffith University, Australia
Caren Han	University of Melbourne, Australia
Ran He	Institute of Automation, Chinese Academy of Sciences, China
Tin Kam Ho	IBM, USA
Di Huang	Beihang University, China
Kaizhu Huang	Duke Kunshan University, China
Donato Impedovo	University of Bari, Italy
Julio Jacques	University of Barcelona and Computer Vision Center, Spain
Lianwen Jin	South China University of Technology, China
Wei Jin	Emory University, USA
Danilo Samuel Jodas	São Paulo State University, Brazil
Manjunath V. Joshi	DA-IICT, India
Jayashree Kalpathy-Cramer	Massachusetts General Hospital, USA
Dimosthenis Karatzas	Computer Vision Centre, Spain
Hamid Karimi	Utah State University, USA
Baiying Lei	Shenzhen University, China
Guoqi Li	Chinese Academy of Sciences, and Peng Cheng Lab, China
Laurence Likforman-Sulem	Institut Polytechnique de Paris/Télécom Paris, France
Aishan Liu	Beihang University, China
Bo Liu	Bytedance, USA
Chen Liu	Clarkson University, USA
Cheng-Lin Liu	Institute of Automation, Chinese Academy of Sciences, China
Hongmin Liu	University of Science and Technology Beijing, China
Hui Liu	Michigan State University, USA

Jing Liu	Institute of Automation, Chinese Academy of Sciences, China
Li Liu	University of Oulu, Finland
Qingshan Liu	Nanjing University of Posts and Telecommunications, China
Adrian P. Lopez-Monroy	Centro de Investigacion en Matematicas AC, Mexico
Daniel P. Lopresti	Lehigh University, USA
Shijian Lu	Nanyang Technological University, Singapore
Yong Luo	Wuhan University, China
Andreas K. Maier	FAU Erlangen-Nuremberg, Germany
Davide Maltoni	University of Bologna, Italy
Hong Man	Stevens Institute of Technology, USA
Lingtong Min	Northwestern Polytechnical University, China
Paolo Napoletano	University of Milano-Bicocca, Italy
Kamal Nasrollahi	Milestone Systems, Aalborg University, Denmark
Marcos Ortega	University of A Coruña, Spain
Shivakumara Palaiahnakote	University of Salford, UK
P. Jonathon Phillips	NIST, USA
Filiberto Pla	University Jaume I, Spain
Ajit Rajwade	Indian Institute of Technology, Bombay, India
Shanmuganathan Raman	Indian Institute of Technology, Gandhinagar, India
Imran Razzak	UNSW, Australia
Beatriz Remeseiro	University of Oviedo, Spain
Gustavo Rohde	University of Virginia, USA
Partha Pratim Roy	Indian Institute of Technology, Roorkee, India
Sanjoy K. Saha	Jadavpur University, India
Joan Andreu Sánchez	Universitat Politècnica de València, Spain
Claudio F. Santos	UFSCar, Brazil
Shin'ichi Satoh	National Institute of Informatics, Japan
Stephanie Schuckers	Clarkson University, USA
Srirangaraj Setlur	University at Buffalo, SUNY, USA
Debdoot Sheet	Indian Institute of Technology, Kharagpur, India
Jun Shen	University of Wollongong, Australia
Li Shen	JD Explore Academy, China
Chen Shengyong	Zhejiang University of Technology and Tianjin University of Technology, China
Andy Song	RMIT University, Australia
Akihiro Sugimoto	National Institute of Informatics, Japan
Qianru Sun	Singapore Management University, Singapore
Arijit Sur	Indian Institute of Technology, Guwahati, India
Estefania Talavera	University of Twente, Netherlands

Wei Tang	University of Illinois at Chicago, USA
Joao M. Tavares	Universidade do Porto, Portugal
Jun Wan	NLPR, CASIA, China
Le Wang	Xi'an Jiaotong University, China
Lei Wang	Australian National University, Australia
Xiaoyang Wang	Tencent AI Lab, USA
Xinggang Wang	Huazhong University of Science and Technology, China
Xiao-Jun Wu	Jiangnan University, China
Yiding Yang	Bytedance, China
Xiwen Yao	Northwestern Polytechnical University, China
Xu-Cheng Yin	University of Science and Technology Beijing, China
Baosheng Yu	University of Sydney, Australia
Shiqi Yu	Southern University of Science and Technology, China
Xin Yuan	Westlake University, China
Yibing Zhan	JD Explore Academy, China
Jing Zhang	University of Sydney, Australia
Lefei Zhang	Wuhan University, China
Min-Ling Zhang	Southeast University, China
Wenbin Zhang	Florida International University, USA
Jiahuan Zhou	Peking University, China
Sanping Zhou	Xi'an Jiaotong University, China
Tianyi Zhou	University of Maryland, USA
Lei Zhu	Shandong Normal University, China
Pengfei Zhu	Tianjin University, China
Wangmeng Zuo	Harbin Institute of Technology, China

Reviewers (Competition Papers)

Liangcai Gao	Da-Han Wang
Mingxin Huang	Yang Xue
Lei Kang	Wentao Yang
Wenhui Liao	Jiixin Zhang
Yuliang Liu	Yiwu Zhong
Yongxin Shi	

Reviewers (Conference Papers)

Aakanksha Aakanksha
 Aayush Singla
 Abdul Muqet
 Abhay Yadav
 Abhijeet Vijay Nandedkar
 Abhimanyu Sahu
 Abhinav Rajvanshi
 Abhisek Ray
 Abhishek Shrivastava
 Abhra Chaudhuri
 Aditi Roy
 Adriano Simonetto
 Adrien Maglo
 Ahmed Abdulkadir
 Ahmed Boudissa
 Ahmed Hamdi
 Ahmed Rida Sekkat
 Ahmed Sharafeldeen
 Aiman Farooq
 Aishwarya Venkataramanan
 Ajay Kumar
 Ajay Kumar Reddy Poreddy
 Ajita Rattani
 Ajoy Mondal
 Akbar K.
 Akbar Telikani
 Akshay Agarwal
 Akshit Jindal
 Al Zadid Sultan Bin Habib
 Albert Clapés
 Alceu Britto
 Alejandro Peña
 Alessandro Ortis
 Alessia Auriemma Citarella
 Alexandre Stenger
 Alexandros Sopasakis
 Alexia Toumpa
 Ali Khan
 Alik Pramanick
 Alireza Alaei
 Alper Yilmaz
 Aman Verma
 Amit Bhardwaj

Amit More
 Amit Nandedkar
 Amitava Chatterjee
 Amos L. Abbott
 Amrita Mohan
 Anand Mishra
 Ananda S. Chowdhury
 Anastasia Zakharova
 Anastasios L. Kesidis
 Andras Horvath
 Andre Gustavo Hochuli
 André P. Kelm
 Andre Wyzykowski
 Andrea Bottino
 Andrea Lagorio
 Andrea Torsello
 Andreas Fischer
 Andreas K. Maier
 Andreu Girbau Xalabarder
 Andrew Beng Jin Teoh
 Andrew Shin
 Andy J. Ma
 Aneesh S. Chivukula
 Ángela Casado-García
 Anh Quoc Nguyen
 Anindya Sen
 Anirban Saha
 Anjali Gautam
 Ankan Bhattacharyya
 Ankit Jha
 Anna Scius-Bertrand
 Annalisa Franco
 Antoine Doucet
 Antonino Staiano
 Antonio Fernández
 Antonio Parziale
 Anu Singha
 Anustup Choudhury
 Anwesan Pal
 Anwasha Sengupta
 Archisman Adhikary
 Arjan Kuijper
 Arnab Kumar Das

Arnav Bhavsar	Bin-Bin Jia
Arnav Varma	Binbin Yong
Arpita Dutta	Bindita Chaudhuri
Arshad Jamal	Bindu Madhavi Tummala
Artur Jordao	Binh M. Le
Arunkumar Chinnaswamy	Bi-Ru Dai
Aryan Jadon	Bo Huang
Aryaz Baradarani	Bo Jiang
Ashima Anand	Bob Zhang
Ashis Dhara	Bowen Liu
Ashish Phophalia	Bowen Zhang
Ashok K. Bhateja	Boyang Zhang
Ashutosh Vaish	Boyu Diao
Ashwani Kumar	Boyun Li
Asifuzzaman Lasker	Brian M. Sadler
Atefeh Khoshkhahtinat	Bruce A. Maxwell
Athira Nambiar	Bryan Bo Cao
Attilio Fiandrotti	Buddhika L. Semage
Avandra S. Hemachandra	Bushra Jalil
Avik Hati	Byeong-Seok Shin
Avinash Sharma	Byung-Gyu Kim
B. H. Shekar	Caihua Liu
B. Uma Shankar	Cairong Zhao
Bala Krishna Thunakala	Camille Kurtz
Balaji Tk	Carlos A. Caetano
Balázs Pálffy	Carlos D. Martá-Nez-Hinarejos
Banafsheh Adami	Ce Wang
Bang-Dang Pham	Cevahir Cigla
Baochang Zhang	Chakravarthy Bhagvati
Baodi Liu	Chandrakanth Vipparla
Bashirul Azam Biswas	Changchun Zhang
Beiduo Chen	Changde Du
Benedikt Kottler	Changkun Ye
Beomseok Oh	Changxu Cheng
Berkay Aydin	Chao Fan
Berlin S. Shaheema	Chao Guo
Bertrand Kerautret	Chao Qu
Bettina Finzel	Chao Wen
Bhavana Singh	Chayan Halder
Bibhas C. Dhara	Che-Jui Chang
Bilge Günsel	Chen Feng
Bin Chen	Chenan Wang
Bin Li	Cheng Yu
Bin Liu	Chenghao Qian
Bin Yao	Cheng-Lin Liu

Chengxu Liu
Chenru Jiang
Chensheng Peng
Chetan Ralekar
Chih-Wei Lin
Chih-Yi Chiu
Chinmay Sahu
Chintan Patel
Chintan Shah
Chiranjoy Chattopadhyay
Chong Wang
Choudhary Shyam Prakash
Christophe Charrier
Christos Smailis
Chuanwei Zhou
Chun-Ming Tsai
Chunpeng Wang
Ciro Russo
Claudio De Stefano
Claudio F. Santos
Claudio Marrocco
Connor Levenson
Constantine Dovrolis
Constantine Kotropoulos
Dai Shi
Dakshina Ranjan Kisku
Dan Anitei
Dandan Zhu
Daniela Pamplona
Danli Wang
Danqing Huang
Daoan Zhang
Daqing Hou
David A. Clausi
David Freire Obregon
David Münch
David Pujol Perich
Davide Marelli
De Zhang
Debalina Barik
Debapriya Roy (Kundu)
Debashis Das
Debashis Das Chakladar
Debi Prosad Dogra
Debraj D. Basu
Decheng Liu
Deen Dayal Mohan
Deep A. Patel
Deepak Kumar
Dengpan Liu
Denis Coquenat
Désiré Sidibé
Devesh Walawalkar
Dewan Md. Farid
Di Ming
Di Qiu
Di Yuan
Dian Jia
Dianmo Sheng
Diego Thomas
Diganta Saha
Dimitri Bulatov
Dimpy Varshni
Dingcheng Yang
Dipanjan Das
Dipanjoyoti Paul
Divya Biligere Shivanna
Divya Saxena
Divya Sharma
Dmitrii Matveichev
Dmitry Minskiy
Dmitry V. Sorokin
Dong Zhang
Donghua Wang
Donglin Zhang
Dongming Wu
Dongqiangzi Ye
Dongqing Zou
Dongrui Liu
Dongyang Zhang
Dongzhan Zhou
Douglas Rodrigues
Duarte Folgado
Duc Minh Vo
Duoxuan Pei
Durai Arun Pannir Selvam
Durga Bhavani S.
Eckart Michaelsen
Elena Goyanes
Élodie Puybareau

Emanuele Vivoli
Emna Ghorbel
Enrique Naredo
Enyu Cai
Eric Patterson
Ernest Valveny
Eva Blanco-Mallo
Eva Breznik
Evangelos Sartinas
Fabio Solari
Fabiola De Marco
Fan Wang
Fangda Li
Fangyuan Lei
Fangzhou Lin
Fangzhou Luo
Fares Bougourzi
Farman Ali
Fatiha Mokdad
Fei Shen
Fei Teng
Fei Zhu
Feiyan Hu
Felipe Gomes Oliveira
Feng Li
Fengbei Liu
Fenghua Zhu
Fillipe D. M. De Souza
Flavio Piccoli
Flavio Prieto
Florian Kleber
Francesc Serratosa
Francesco Bianconi
Francesco Castro
Francesco Ponzio
Francisco Javier Hernández López
Frédéric Rayar
Furkan Osman Kar
Fushuo Huo
Fuxiao Liu
Fu-Zhao Ou
Gabriel Turinici
Gabrielle Flood
Gajjala Viswanatha Reddy
Gaku Nakano
Galal Binamakhshen
Ganesh Krishnasamy
Gang Pan
Gangyan Zeng
Gani Rahmon
Gaurav Harit
Gennaro Vessio
Genoveffa Tortora
George Azzopardi
Gerard Ortega
Gerardo E. Altamirano-Gomez
Gernot A. Fink
Gibran Benitez-Garcia
Gil Ben-Artzi
Gilbert Lim
Giorgia Minello
Giorgio Fumera
Giovanna Castellano
Giovanni Puglisi
Giulia Orrù
Giuliana Ramella
Gökçe Uludoğan
Gopi Ramena
Gorthi Rama Krishna Sai Subrahmanyam
Gourav Datta
Gowri Srinivasa
Gozde Sahin
Gregory Randall
Guanjie Huang
Guanjun Li
Guanwen Zhang
Guanyu Xu
Guanyu Yang
Guanzhou Ke
Guhnoo Yun
Guido Borghi
Guilherme Brandão Martins
Guillaume Caron
Guillaume Tochon
Guocai Du
Guohao Li
Guoqiang Zhong
Guorong Li
Guotao Li
Gurman Gill

Haechang Lee
Haichao Zhang
Haidong Xie
Haifeng Zhao
Haimei Zhao
Hainan Cui
Haixia Wang
Haiyan Guo
Hakime Ozturk
Hamid Kazemi
Han Gao
Hang Zou
Hanjia Lyu
Hanjoo Cho
Hanqing Zhao
Hanyuan Liu
Hanzhou Wu
Hao Li
Hao Meng
Hao Sun
Hao Wang
Hao Xing
Hao Zhao
Haoan Feng
Haodi Feng
Haofeng Li
Haoji Hu
Haojie Hao
Haojun Ai
Haopeng Zhang
Haoran Li
Haoran Wang
Haorui Ji
Haoxiang Ma
Haoyu Chen
Haoyue Shi
Harald Koestler
Harbinder Singh
Harris V. Georgiou
Hasan F. Ates
Hasan S. M. Al-Khaffaf
Hatef Otroshi Shahreza
Hebeizi Li
Heng Zhang
Hengli Wang
Hengyue Liu
Hertog Nugroho
Hieyong Jeong
Himadri Mukherjee
Hoai Ngo
Hoda Mohaghegh
Hong Liu
Hong Man
Hongcheng Wang
Hongjian Zhan
Hongxi Wei
Hongyu Hu
Hoseong Kim
Hossein Ebrahimnezhad
Hossein Malekmohamadi
Hrishav Bakul Barua
Hsueh-Yi Sean Lin
Hua Wei
Huafeng Li
Huali Xu
Huaming Chen
Huan Wang
Huang Chen
Huanran Chen
Hua-Wen Chang
Huawen Liu
Huayi Zhan
Hugo Jair Escalante
Hui Chen
Hui Li
Huichen Yang
Huiqiang Jiang
Huiyuan Yang
Huizi Yu
Hung T. Nguyen
Hyeongyu Kim
Hyeonjeong Park
Hyeonjun Lee
Hymalai Bello
Hyung-Gun Chi
Hyunsoo Kim
I-Chen Lin
Ik Hyun Lee
Ilan Shimshoni
Imad Eddine Toubal

Imran Sarker
Inderjot Singh Saggu
Indrani Mukherjee
Indranil Sur
Ines Rieger
Ioannis Pierros
Irina Rabaev
Ivan V. Medri
J. Rafid Siddiqui
Jacek Komorowski
Jacopo Bonato
Jacson Rodrigues Correia-Silva
Jaekoo Lee
Jaime Cardoso
Jakob Gawlikowski
Jakub Nalepa
James L. Wayman
Jan Čech
Jangho Lee
Jani Boutellier
Javier Gurrola-Ramos
Javier Lorenzo-Navarro
Jayasree Saha
Jean Lee
Jean Paul Barddal
Jean-Bernard Hayet
Jean-Philippe G. Tarel
Jean-Yves Ramel
Jenny Benois-Pineau
Jens Bayer
Jerin Geo James
Jesús Miguel García-Gorrostieta
Jia Qu
Jiahong Chen
Jiaji Wang
Jian Hou
Jian Liang
Jian Xu
Jian Zhu
Jianfeng Lu
Jianfeng Ren
Jiangfan Liu
Jianguo Wang
Jiangyan Yi
Jiangyong Duan
Jianhua Yang
Jianhua Zhang
Jianhui Chen
Jianjia Wang
Jianli Xiao
Jianqiang Xiao
Jianwu Wang
Jianxin Zhang
Jianxiong Gao
Jianxiong Zhou
Jianyu Wang
Jianzhong Wang
Jiaru Zhang
Jiashu Liao
Jiaxin Chen
Jiaxin Lu
Jiaxing Ye
Jiaxuan Chen
Jiaxuan Li
Jiayi He
Jiayin Lin
Jie Ou
Jiehua Zhang
Jiejie Zhao
Jignesh S. Bhatt
Jin Gao
Jin Hou
Jin Hu
Jin Shang
Jing Tian
Jing Yu Chen
Jingfeng Yao
Jinglun Feng
Jingtong Yue
Jingwei Guo
Jingwen Xu
Jingyuan Xia
Jingzhe Ma
Jinhong Wang
Jinjia Wang
Jinlai Zhang
Jinlong Fan
Jinming Su
Jinrong He
Jintao Huang

Jinwoo Ahn
Jinwoo Choi
Jinyang Liu
Jinyu Tian
Jionghao Lin
Jiuding Duan
Jiwei Shen
Jiyang Pan
Jiyoun Kim
João Papa
Johan Debayle
John Atanbori
John Wilson
John Zhang
Jónathan Heras
Joohi Chauhan
Jorge Calvo-Zaragoza
Jorge Figueroa
Jorma Laaksonen
José Joaquim De Moura Ramos
Jose Vicent
Joseph Damilola Akinyemi
Josiane Zerubia
Juan Wen
Judit Szücs
Juepeng Zheng
Juha Roning
Jumana H. Alsubhi
Jun Cheng
Jun Ni
Jun Wan
Junghyun Cho
Junjie Liang
Junjie Ye
Junlin Hu
Juntong Ni
Junxin Lu
Junxuan Li
Junyaup Kim
Junyeong Kim
Jürgen Seiler
Jushang Qiu
Juyang Weng
Jyostna Devi Bodapati
Jyoti Singh Kirar
Kai Jiang
Kaiqiang Song
Kalidas Yeturu
Kalle Åström
Kamalakar Vijay Thakare
Kang Gu
Kang Ma
Kanji Tanaka
Karthik Seemakurthy
Kaushik Roy
Kavisha Jayathunge
Kazuki Uehara
Ke Shi
Keigo Kimura
Keiji Yanai
Kelton A. P. Costa
Kenneth Camilleri
Kenny Davila
Ketan Atul Bapat
Ketan Kotwal
Kevin Desai
Keyu Long
Khadiga Mohamed Ali
Khakon Das
Khan Muhammad
Kilho Son
Kim-Ngan Nguyen
Kishan Kc
Kishor P. Upla
Klaas Dijkstra
Komal Bharti
Konstantinos Triaridis
Kostas Ioannidis
Koyel Ghosh
Kripabandhu Ghosh
Krishnendu Ghosh
Kshitij S. Jadhav
Kuan Yan
Kun Ding
Kun Xia
Kun Zeng
Kunal Banerjee
Kunal Biswas
Kunchi Li
Kurban Ubul

Lahiru N. Wijayasingha
Laines Schmalwasser
Lakshman Mahto
Lala Shakti Swarup Ray
Lale Akarun
Lan Yan
Lawrence Amadi
Lee Kang Il
Lei Fan
Lei Shi
Lei Wang
Leonardo Rossi
Lequan Lin
Levente Tamas
Li Bing
Li Li
Li Ma
Li Song
Lia Morra
Liang Xie
Liang Zhao
Lianwen Jin
Libing Zeng
Lidia Sánchez-González
Lidong Zeng
Lijun Li
Likang Wang
Lili Zhao
Lin Chen
Lin Huang
Linfei Wang
Ling Lo
Lingchen Meng
Lingheng Meng
Lingxiao Li
Lingzhong Fan
Liqi Yan
Liqiang Jing
Lisa Gutzeit
Liu Ziyi
Liushuai Shi
Liviú-Daniel Stefan
Liyuan Ma
Liyun Zhu
Lizuo Jin

Longteng Guo
Lorena Álvarez Rodríguez
Lorenzo Putzu
Lu Leng
Lu Pang
Lu Wang
Luan Pham
Luc Brun
Luca Guarnera
Luca Piano
Lucas Alexandre Ramos
Lucas Goncalves
Lucas M. Gago
Luigi Celona
Luis C. S. Afonso
Luis Gerardo De La Fraga
Luis S. Luevano
Luis Teixeira
Lunke Fei
M. Hassaballah
Maddimsetti Srinivas
Mahendran N.
Mahesh Mohan M. R.
Maiko Lie
Mainak Singha
Makoto Hirose
Malay Bhattacharyya
Mamadou Dian Bah
Man Yao
Manali J. Patel
Manav Prabhakar
Manikandan V. M.
Manish Bhatt
Manjunath Shantharamu
Manuel Curado
Manuel Günther
Manuel Marques
Marc A. Kastner
Marc Chaumont
Marc Cheong
Marc Lalonde
Marco Cotogni
Marcos C. Santana
Mario Molinara
Mariofanna Milanova

Markus Bauer
Marlon Becker
Mårten Wadenbäck
Martin G. Ljungqvist
Martin Kämpel
Martina Pastorino
Marwan Turki
Masashi Nishiyama
Masayuki Tanaka
Massimo O. Spata
Matteo Ferrara
Matthew D. Dawkins
Matthew Gadd
Matthew S. Watson
Maura Pintor
Max Ehrlich
Maxim Popov
Mayukh Das
Md Baharul Islam
Md Sajid
Meghna Kapoor
Meghna P. Ayyar
Mei Wang
Meiqi Wu
Melissa L. Tijink
Meng Li
Meng Liu
Meng-Luen Wu
Mengnan Liu
Mengxi China Guo
Mengya Han
Michaël Clément
Michal Kawulok
Mickael Coustaty
Miguel Domingo
Milind G. Padalkar
Ming Liu
Ming Ma
Mingchen Feng
Mingde Yao
Minghao Li
Mingjie Sun
Ming-Kuang Daniel Wu
Mingle Xu
Mingyong Li
Mingyuan Jiu
Minh P. Nguyen
Minh Q. Tran
Minheng Ni
Minsu Kim
Minyi Zhao
Mirko Paolo Barbato
Mo Zhou
Modesto Castrillón-Santana
Mohamed Amine Mezghich
Mohamed Dahmane
Mohamed Elsharkawy
Mohamed Yousuf
Mohammad Hashemi
Mohammad Khalooei
Mohammad Khateri
Mohammad Mahdi Dehshibi
Mohammad Sadil Khan
Mohammed Mahmoud
Moises Diaz
Monalisha Mahapatra
Monidipa Das
Mostafa Kamali Tabrizi
Mridul Ghosh
Mrinal Kanti Bhowmik
Muchao Ye
Mugalodi Ramesha Rakesh
Muhammad Rameez Ur Rahman
Muhammad Suhaib Kanroo
Muming Zhao
Munender Varshney
Munsif Ali
Na Lv
Nader Karimi
Nagabhushan Somraj
Nakkwan Choi
Nakul Agarwal
Nan Pu
Nan Zhou
Nancy Mehta
Nand Kumar Yadav
Nandakishor Nandakishor
Nandyala Hemachandra
Nanfeng Jiang
Narayan Hegde

Narayan Ji Mishra	Palash Ghosal
Narayan Vetrekar	Pallav Dutta
Narendra D. Londhe	Paolo Rota
Nathalie Girard	Paramanand Chandramouli
Nati Ofir	Paria Mehrani
Naval Kishore Mehta	Parth Agrawal
Nazmul Shahadat	Partha Basuchowdhuri
Neeti Narayan	Patrick Horain
Neha Bhargava	Pavan Kumar
Nemanja Djuric	Pavan Kumar Anasosalu Vasu
Newlin Shebiah R.	Pedro Castro
Ngo Ba Hung	Peipei Li
Nhat-Tan Bui	Peipei Yang
Niaz Ahmad	Peisong Shen
Nick Theisen	Peiyu Li
Nicolas Passat	Peng Li
Nicolas Ragot	Pengfei He
Nicolas Sidere	Pengrui Quan
Nikolaos Mitianoudis	Pengxin Zeng
Nikolas Ebert	Pengyu Yan
Nilah Ravi Nair	Peter Eisert
Nilesh A. Ahuja	Petra Gomez-Krämer
Nilkanta Sahu	Pierrick Bruneau
Nils Murrugarra-Llerena	Ping Cao
Nina S. T. Hirata	Pingping Zhang
Ninad Aithal	Pintu Kumar
Ning Xu	Pooja Kumari
Ningzhi Wang	Pooja Sahani
Niraj Kumar	Prabhu Prasad Dev
Nirmal S. Punjabi	Pradeep Kumar
Nisha Varghese	Pradeep Singh
Norio Tagawa	Pranjal Sahu
Obaidullah Md Sk	Prasun Roy
Oguzhan Ulucan	Prateek Keserwani
Olfa Mechi	Prateek Mittal
Oliver Tüselmann	Praveen Kumar Chandaliya
Orazio Pontorno	Praveen Tirupattur
Oriol Ramos Terrades	Pravin Nair
Osman Akin	Preeti Gopal
Ouadi Beya	Preety Singh
Ozge Mercanoglu Sincan	Prem Shanker Yadav
Pabitra Mitra	Prerana Mukherjee
Padmanabha Reddy Y. C. A.	Prerna A. Mishra
Palaash Agrawal	Prianka Dey
Palaiahnakote Shivakumara	Priyanka Mudgal

Qc Kha Ng
Qi Li
Qi Ming
Qi Wang
Qi Zuo
Qian Li
Qiang Gan
Qiang He
Qiang Wu
Qiangqiang Zhou
Qianli Zhao
Qiansen Hong
Qiao Wang
Qidong Huang
Qihua Dong
Qin Yuke
Qing Guo
Qingbei Guo
Qingchao Zhang
Qingjie Liu
Qinhong Yang
Qiushi Shi
Qixiang Chen
Quan Gan
Quanlong Guan
Rachit Chhaya
Radu Tudor Ionescu
Rafal Zdunek
Raghavendra Ramachandra
Rahimul I. Mazumdar
Rahul Kumar Ray
Rajib Dutta
Rajib Ghosh
Rakesh Kumar
Rakesh Paul
Rama Chellappa
Rami O. Skaik
Ramon Aranda
Ran Wei
Ranga Raju Vatsavai
Ranganath Krishnan
Rasha Friji
Rashmi S.
Razaib Tariq
Rémi Giraud
René Schuster
Renlong Hang
Renrong Shao
Renu Sharma
Reza Sadeghian
Richard Zanibbi
Rimon Elias
Rishabh Shukla
Rita Delussu
Riya Verma
Robert J. Ravier
Robert Sablatnig
Robin Strand
Rocco Pietrini
Rocio Diaz Martin
Rocio Gonzalez-Diaz
Rohit Venkata Sai Dulam
Romain Giot
Romi Banerjee
Ru Wang
Ruben Machucho
Ruddy Théodose
Ruggero Pintus
Rui Deng
Rui P. Paiva
Rui Zhao
Ruifan Li
Ruigang Fu
Ruikun Li
Ruirui Li
Ruixiang Jiang
Ruwei Jiang
Rushi Lan
Rustam Zhumagambetov
S. Amutha
S. Divakar Bhat
Sagar Goyal
Sahar Siddiqui
Sahbi Bahroun
Sai Karthikeya Vemuri
Saibal Dutta
Saihui Hou
Sajad Ahmad Rather
Saksham Aggarwal
Sakthi U.

Salimeh Sekeh
Samar Bouazizi
Samia Boukir
Samir F. Harb
Samit Biswas
Samrat Mukhopadhyay
Samriddha Sanyal
Sandika Biswas
Sandip Purnapatra
Sanghyun Jo
Sangwoo Cho
Sanjay Kumar
Sankaran Iyer
Sanket Biswas
Santanu Roy
Santosh D. Pandure
Santosh Ku Behera
Santosh Nanabhau Palaskar
Santosh Prakash Chouhan
Sarah S. Alotaibi
Sasanka Katreddi
Sathyanarayanan N. Aakur
Saurabh Yadav
Sayan Rakshit
Scott McCloskey
Sebastian Bunda
Sejuti Rahman
Selim Aksoy
Sen Wang
Seraj A. Mostafa
Shanmuganathan Raman
Shao-Yuan Lo
Shaoyuan Xu
Sharia Arfin Tanim
Shehreen Azad
Sheng Wan
Shengdong Zhang
Shengwei Qin
Shenyuan Gao
Sherry X. Chen
Shibaprasad Sen
Shigeaki Namiki
Shiguang Liu
Shijie Ma
Shikun Li
Shinichiro Omachi
Shirley David
Shishir Shah
Shiv Ram Dubey
Shiva Baghel
Shivanand S. Gornale
Shogo Sato
Shotaro Miwa
Shreya Ghosh
Shreya Goyal
Shuai Su
Shuai Wang
Shuai Zheng
Shuaifeng Zhi
Shuang Qiu
Shuhei Tarashima
Shujing Lyu
Shuliang Wang
Shun Zhang
Shunming Li
Shunxin Wang
Shuping Zhao
Shuquan Ye
Shuwei Huo
Shuyue Lan
Shyi-Chyi Cheng
Si Chen
Siddarth Ravichandran
Sihan Chen
Siladitya Manna
Silambarasan Elkana Ebinazer
Simon Benaïchouche
Simon S. Woo
Simone Caldarella
Simone Milani
Simone Zini
Sina Lotfian
Sitao Luan
Sivaselvan B.
Siwei Li
Siwei Wang
Siwen Luo
Siyu Chen
Sk Aziz Ali
Sk Md Obaidullah

Sneha Shukla
 Snehasis Banerjee
 Snehasis Mukherjee
 Snigdha Sen
 Sofia Casarin
 Soheila Farokhi
 Soma Bandyopadhyay
 Son Minh Nguyen
 Son Xuan Ha
 Sonal Kumar
 Sonam Gupta
 Sonam Nahar
 Song Ouyang
 Sotiris Kotsiantis
 Souhaila Djaffal
 Soumen Biswas
 Soumen Sinha
 Soumitri Chattopadhyay
 Souvik Sengupta
 Spiros Kostopoulos
 Sreeraj Ramachandran
 Sreya Banerjee
 Srikanta Pal
 Srinivas Arukonda
 Stephane A. Guinard
 Su O. Ruan
 Subhadip Basu
 Subhajit Paul
 Subhankar Ghosh
 Subhankar Mishra
 Subhankar Roy
 Subhash Chandra Pal
 Subhayu Ghosh
 Sudip Das
 Sudipta Banerjee
 Suhas Pillai
 Sujit Das
 Sukalpa Chanda
 Sukhendu Das
 Suklav Ghosh
 Suman K. Ghosh
 Suman Samui
 Sumit Mishra
 Sungho Suh
 Sunny Gupta

Suraj Kumar Pandey
 Surendrabikram Thapa
 Suresh Sundaram
 Sushil Bhattacharjee
 Susmita Ghosh
 Swakkhar Shatabda
 Syed Ms Islam
 Syed Tousiful Haque
 Taegyeong Lee
 Taihui Li
 Takashi Shibata
 Takeshi Oishi
 Talha Ahmad Siddiqui
 Tanguy Gernot
 Tangwen Qian
 Tanima Bhowmik
 Tanpia Tasnim
 Tao Dai
 Tao Hu
 Tao Sun
 Taoran Yi
 Tapan Shah
 Taveena Lotey
 Teng Huang
 Tengqi Ye
 Teresa Alarcon
 Tetsuji Ogawa
 Thanh Phuong Nguyen
 Thanh Tuan Nguyen
 Thattapon Surasak
 Thibault Napol on
 Thierry Bouwmans
 Thinh Truong Huynh Nguyen
 Thomas De Min
 Thomas E. K. Zielke
 Thomas Swearingen
 Tianatahina Jimmy Francky Randrianasoa
 Tianheng Cheng
 Tianjiao He
 Tianyi Wei
 Tianyuan Zhang
 Tianyue Zheng
 Tiecheng Song
 Tilottama Goswami
 Tim B chner

Tim H. Langer	Wataru Ohyama
Tim Raven	Wee Kheng Leow
Ting kai Liu	Wei Chen
Tingting Yao	Wei Cheng
Tobias Meisen	Wei Hua
Toby P. Breckon	Wei Lu
Tong Chen	Wei Pan
Tonghua Su	Wei Tian
Tran Tuan Anh	Wei Wang
Tri-Cong Pham	Wei Wei
Trishna Saikia	Wei Zhou
Trung Quang Truong	Weidi Liu
Tuan T. Nguyen	Weidong Yang
Tuan Vo Van	Weijun Tan
Tushar Shinde	Weimin Lyu
Ujjwal Karn	Weinan Guan
Ukrit Watchareeruetai	Weining Wang
Uma Mudenagudi	Weiqiang Wang
Umarani Jayaraman	Weiwei Guo
V. S. Malemath	Weixia Zhang
Vallidevi Krishnamurthy	Wei-Xuan Bao
Ved Prakash	Weizhong Jiang
Venkata Krishna Kishore Kolli	Wen Xie
Venkata R. Vavilthota	Wenbin Qian
Venkatesh Thirugnana Sambandham	Wenbin Tian
Verónica Maria Vasconcelos	Wenbin Wang
Véronique Ve Eglin	Wenbo Zheng
Víctor E. Alonso-Pérez	Wenhan Luo
Vinay Palakkode	Wenhao Wang
Vinayak S. Nageli	Wen-Hung Liao
Vincent J. Whannou De Dravo	Wenjie Li
Vincenzo Conti	Wenkui Yang
Vincenzo Gattulli	Wenwen Si
Vineet Padmanabhan	Wenwen Yu
Vishakha Pareek	Wenwen Zhang
Viswanath Gopalakrishnan	Wenwu Yang
Vivek Singh Baghel	Wenxi Li
Vivekraj K.	Wenxi Yue
Vladimir V. Arlazarov	Wenxue Cui
Vu-Hoang Tran	Wenzhuo Liu
W. Sylvia Lilly Jebarani	Widhiyo Sudiyono
Wachirawit Ponghiran	Willem Dijkstra
Wafa Khlif	Wolfgang Fuhl
Wang An-Zhi	Xi Zhang
Wanli Xue	Xia Yuan

Xianda Zhang
Xiang Zhang
Xiangdong Su
Xiang-Ru Yu
Xiangtai Li
Xiangyu Xu
Xiao Guo
Xiao Hu
Xiao Wu
Xiao Yang
Xiaofeng Zhang
Xiaogang Du
Xiaoguang Zhao
Xiaoheng Jiang
Xiaohong Zhang
Xiaohua Huang
Xiaohua Li
Xiao-Hui Li
Xiaolong Sun
Xiaosong Li
Xiaotian Li
Xiaoting Wu
Xiaotong Luo
Xiaoyan Li
Xiaoyang Kang
Xiaoyi Dong
Xin Guo
Xin Lin
Xin Ma
Xinchi Zhou
Xingguang Zhang
Xingjian Leng
Xingpeng Zhang
Xingzheng Lyu
Xinjian Huang
Xinqi Fan
Xinqi Liu
Xinqiao Zhang
Xinrui Cui
Xizhan Gao
Xu Cao
Xu Ouyang
Xu Zhao
Xuan Shen
Xuan Zhou

Xuchen Li
Xuejing Lei
Xuelu Feng
Xueting Liu
Xuewei Li
Xueyi X. Wang
Xugong Qin
Xu-Qian Fan
Xuxu Liu
Xu-Yao Zhang
Yan Huang
Yan Li
Yan Wang
Yan Xia
Yan Zhuang
Yanan Li
Yanan Zhang
Yang Hou
Yang Jiao
Yang Liping
Yang Liu
Yang Qian
Yang Yang
Yang Zhao
Yangbin Chen
Yangfan Zhou
Yanhui Guo
Yanjia Huang
YanJun Zhu
Yanming Zhang
Yanqing Shen
Yaoming Cai
Yaoxin Zhuo
Yaoyan Zheng
Yaping Zhang
Yaqian Liang
Yarong Feng
Yasmina Benmabrouk
Yasufumi Sakai
Yasutomo Kawanishi
Yazeed Alzahrani
Ye Du
Ye Duan
Yechao Zhang
Yeong-Jun Cho

Yi Huo
Yi Shi
Yi Yu
Yi Zhang
Yibo Liu
Yibo Wang
Yi-Chieh Wu
Yifan Chen
Yifei Huang
Yihao Ding
Yijie Tang
Yikun Bai
Yimin Wen
Yinan Yang
Yin-Dong Zheng
Yinfeng Yu
Ying Dai
Yingbo Li
Yiqiao Li
Yiqing Huang
Yisheng Lv
Yisong Xiao
Yite Wang
Yizhe Li
Yong Wang
Yonghao Dong
Yong-Hyuk Moon
Yongjie Li
Yongqian Li
Yongqiang Mao
Yongxu Liu
Yongyu Wang
Yongzhi Li
Youngha Hwang
Yousri Kessentini
Yu Wang
Yu Zhou
Yuan Tian
Yuan Zhang
Yuanbo Wen
Yuanxin Wang
Yubin Hu
Yubo Huang
Yuchen Ren
Yucheng Xing
Yuchong Yao
Yuecong Min
Yuewei Yang
Yufei Zhang
Yufeng Yin
Yugen Yi
Yuhang Ming
Yujia Zhang
Yujun Ma
Yukiko Kenmochi
Yun Hoyeoung
Yun Liu
Yunhe Feng
Yunxiao Shi
Yuru Wang
Yushun Tang
Yusuf Osmanlioglu
Yusuke Fujita
Yuta Nakashima
Yuwei Yang
Yuwu Lu
Yuxi Liu
Yuya Obinata
Yuyao Yan
Yuzhi Guo
Zaipeng Xie
Zander W. Blasingame
Zedong Wang
Zeliang Zhang
Zexin Ji
Zhanxiang Feng
Zhaofei Yu
Zhe Chen
Zhe Cui
Zhe Liu
Zhe Wang
Zhekun Luo
Zhen Yang
Zhenbo Li
Zhenchun Lei
Zhenfei Zhang
Zheng Liu
Zheng Wang
Zhengming Yu
Zhengyin Du

Zhengyun Cheng
Zhenshen Qu
Zhenwei Shi
Zhenzhong Kuang
Zhi Cai
Zhi Chen
Zhibo Chu
Zhicun Yin
Zhida Huang
Zhida Zhang
Zhifan Gao
Zhihang Ren
Zhihang Yuan
Zhihao Wang
Zhihua Xie
Zhihui Wang
Zhikang Zhang
Zhiming Zou
Zhiqi Shao
Zhiwei Dong
Zhiwei Qi
Zhixiang Wang
Zhixuan Li
Zhiyu Jiang
Zhiyuan Yan
Zhiyuan Yu
Zhiyuan Zhang
Zhong Chen
Zhongwei Teng
Zhongzhan Huang
Zhongzhi Yu
Zhuan Han
Zhuangzhuang Chen
Zhuo Liu
Zhuo Su
Zhuojun Zou
Zhuoyue Wang
Ziang Song
Zicheng Zhang
Zied Mnasri
Zifan Chen
Žiga Babnik
Zijing Chen
Zikai Zhang
Ziling Huang
Zilong Du
Ziqi Cai
Ziqi Zhou
Zi-Rui Wang
Zirui Zhou
Ziwen He
Ziyao Zeng
Ziyi Zhang
Ziyue Xiang
Zonglei Jing
Zongyi Xu

Contents – Part III

Deep Multi-order Context-Aware Kernel Network for Multi-label Classification	1
<i>Mingyuan Jiu, Hailong Zhu, and Hichem Sahbi</i>	
Classifier Enhanced Deep Learning Model for Erythroblast Differentiation with Limited Data	18
<i>Buddhadev Goswami, Adithya B. Somaraj, Prantar Chakrabarti, Ravindra Gudi, and Nirmal Punjabi</i>	
PiExtract: An End-to-End Data Extraction Pipeline for Pie-Charts	31
<i>Muhammad Suhaib Kanroo, Hadia Showkat Kawoosa, Joy Dhar, and Puneet Goyal</i>	
Machine Learning Solutions for Predicting Bankruptcy in Indian Firms	47
<i>Chaithra, Priyanshu Sharma, and Biju R. Mohan</i>	
Efficient Object Detection via Fine-Grained Regularization with Global Initialization	65
<i>Binhan Chen, Qiaojun Wu, Song Chen, and Yi Kang</i>	
On Trace of PGD-Like Adversarial Attacks	81
<i>Mo Zhou and Vishal M. Patel</i>	
CAB-KWS : Contrastive Augmentation: An Unsupervised Learning Approach for Keyword Spotting in Speech Technology	98
<i>Weinan Dai, Yifeng Jiang, Yuanjing Liu, Jinkun Chen, Xin Sun, and Jinglei Tao</i>	
Deep Learning in Automated Worm Identification and Tracking for C. Elegans Mating Behaviour Analysis	113
<i>Chukwuma Hilary Akpu, Hong Wei, and Xia Hong</i>	
Interactive-Time Text-Guided Editing of 3D Face	129
<i>Yeon-Jeong Lee, Yeong-Hun Song, Sang Wook Yoo, and Joon-Kyung Seong</i>	
Unlearning Vision Transformers Without Retaining Data via Low-Rank Decompositions	147
<i>Samuele Poppi, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara</i>	

gWaveNet: Classification of Gravity Waves from Noisy Satellite Data Using Custom Kernel Integrated Deep Learning Method	164
<i>Seraj Al Mahmud Mostafa, Omar Faruque, Chenxi Wang, Jia Yue, Sanjay Purushotham, and Jianwu Wang</i>	
Neural-Code PIFu: High-Fidelity Single Image 3D Human Reconstruction via Neural Code Integration	181
<i>Ruizhi Liu, Paolo Remagnino, and Hubert P. H. Shum</i>	
Sea-ShipNet: Detect Any Ship in SAR Images	196
<i>Qinglin Zhang, Donghai Guan, Weiwei Yuan, and Mingqiang Wei</i>	
Semantic Correlation Adaptation for Union-Set Multi-label Image Recognition	210
<i>Xinyu Wang, Tao Pu, Dongyu Zhang, and Liang Lin</i>	
FedSC: Federated Generalized Face Anti-Spoofing via Shuffled Codebook	226
<i>Shuai Yang, Mei Wang, Weihong Deng, and Jiani Hu</i>	
LoHoSC: Low Order High Order Style Consistency for Syn-to-Real Domain Generalized Semantic Segmentation	243
<i>Sudhakar Kumawat and Hajime Nagahara</i>	
Incorporating Spatial Locality Into Self-attention for Training Vision Transformer on Small-Scale Datasets	259
<i>Yuki Igaue, Takio Kurita, and Hiroaki Aizawa</i>	
Cross-Domain Calibration and Boundary Denoising Network for Weakly Supervised Semantic Segmentation	275
<i>Zhoufeng Liu, Bingrui Li, Shumin Ding, Jiangtao Xi, and Chunlei Li</i>	
EFLLD-NET: Enhancing Few-Shot Learning with Local Descriptors	289
<i>Guangtong Lu, Weidong Du, and Fanzhang Li</i>	
Using Multiscale Information for Improved Optimization-Based Image Attribution	303
<i>Aniket Singh and Anoop Namboodiri</i>	
Split-DNN Computing for Video Analytics	322
<i>Nagabhushan Eswara, Jaroslaw Sydir, V. Srinivasa Somayazulu, Parual Datta, Nilesh Ahuja, and Omesh Tickoo</i>	
Task-Aware Local Descriptors Reconstruction Network for Few-Shot Find-Grained Image Classification	339
<i>Jianchang Tan, Xiangqian Ding, and Shusong Yu</i>	

TRIGS: Trojan Identification from Gradient-Based Signatures 356
*Mohamed Hussein, Sudharshan Subramaniam Janakiraman,
and Wael AbdAlmageed*

Multifaceted Anchor Nodes Attack on Graph Neural Networks:
A Budget-Efficient Approach 372
Huanzhang Zhu, Shaoxin Li, and Lingyang Chu

Causal Attentive Group Recommendation 391
Liancheng Xu, Xiaoqi Wu, Xiaoxiang Wang, and Xinhua Wang

E^2 DAS: An Efficient Equivariant Dynamic Aggregation Saliency Model
for Omnidirectional Images 407
*Nana Zhang, Qian Liu, Dandan Zhu, Kun Zhu, Guangtao Zhai,
and Xiaokang Yang*

FewConv: Efficient Variant Convolution for Few-Shot Image Generation 424
Si-Hao Liu, Cong Hu, Xiao-Ning Song, Jia-Sheng Chen, and Xiao-Jun Wu

FixPix: Fixing Bad Pixels using Deep Learning 441
Sreetama Sarkar, Xinan Ye, Gourav Datta, and Peter A. Beerel

Real-World Coarse to Fine-Grained Source-Free Multidomain Adaptation 456
Anoushka Banerjee and Ananth Ganesh

Author Index 473



Deep Multi-order Context-Aware Kernel Network for Multi-label Classification

Mingyuan Jiu^{1,2,3(✉)}, Hailong Zhu¹, and Hichem Sahbi⁴

¹ School of Computer and Artificial Intelligence,
Zhengzhou University, Zhengzhou, China
iemyjiu@zzu.edu.cn

² Engineering Research Center of Intelligent Swarm Systems, Ministry of Education,
Zhengzhou University, Zhengzhou, China

³ National Supercomputing Center in Zhengzhou, Zhengzhou, China

⁴ Sorbonne University, CNRS, LIP6, 75005 Paris, France

Abstract. Multi-label classification is a challenging task in pattern recognition. Many deep learning methods have been proposed and largely enhanced classification performance. However, most of the existing sophisticated methods ignore context in the models' learning process. Since context may provide additional cues to the learned models, it may significantly boost classification performances. In this work, we make full use of context information (namely geometrical structure of images) in order to learn better context-aware similarities (a.k.a. kernels) between images. We reformulate context-aware kernel design as a feed-forward network that outputs explicit kernel mapping features. Our obtained context-aware kernel network further leverages multiple orders of patch neighbors within different distances, resulting into a more discriminating Deep Multi-order Context-aware Kernel Network (DMCKN) for multi-label classification. We evaluate the proposed method on the challenging Corel5K and NUS-WIDE benchmarks, and empirical results show that our method obtains competitive performances against the related state-of-the-art, and both quantitative and qualitative performances corroborate its effectiveness and superiority for multi-label image classification.

Keywords: Multi-label classification · Context-aware kernel · Deep learning · Deep unfolding

1 Introduction

Multi-label image classification is a challenging task in pattern recognition. It aims at identifying the presence of objects, scenes, or concepts by assigning multiple labels to images. This task is crucial for parsing and understanding visual information, significantly enhancing machine cognition of complex visual scenes.

This work is supported by grants from the National Natural Science Foundation of China (No. 62272422, U22B2051).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15303, pp. 1–17, 2025.
https://doi.org/10.1007/978-3-031-78122-3_1

Multi-label classification can also be applied to many scenarios, such as human attribute recognition, scene understanding, image tagging, labeling, and so on. However, multi-label classification encounters many challenges [1], primarily due to the complexity and diversity of the image contents. Compared to single-label classification [5], this task requires the model to simultaneously recognize all relevant objects and concepts in an image, and annotate them accurately, demanding higher requisite on the model’s discrimination and generalization ability. The relationships among objects in an image can also be exceedingly complex, including, but not limited to, exclusivity, dependency, and hierarchical relationships. Additionally, long tail label distributions further increases the difficulty of multi-label classification.

Recently, the rapid development of deep learning techniques [32], especially the introduction of Transformer [7] and attention mechanism [31], along with label relationship learning through Graph Convolutional Networks (GCN) [4, 26], has significantly advanced multi-label classification performance. These edge-cutting methods, that learn intricate dependencies between pixels, regions, or labels, have significantly improved recognition and classification performance in complex scenes. Despite these advancements, challenges remain in order to fully leverage contextual information and structural relationships among objects within images.

It is well known that appropriately leveraging contextual information into a learning model can enhance performances [27, 30]. Following this line, this work proposes a novel multi-label classification framework that learns rich contextual information within images through structure-aware perception. Based on a deep understanding of the importance of context and multi-layer deep networks, our framework effectively captures complex and fine-grained relationships in images. This is achieved by learning these complex relationships as a part of a designed sophisticated kernel function. The latter allows to obtain a significant gain in accuracy and robustness of multi-label classification. Considering the aforementioned issues, the main contributions of this work include:

- A novel multi-label classification framework that combines contextual information through a deep multi-order context-aware kernel network (DMCKN), resulting in more discriminative features;
- An end-to-end framework that learns the geometrical relationships between image regions with increasing contextual ranges;
- Extensive experiments on several benchmarks which show that our method obtains very competitive results and significantly outperforms different baselines as well as the related approaches.

2 Related Work

2.1 Multi-label Classification

The study of multi-label classification has attracted increasing attention in recent years. Initial efforts primarily focused on generating region proposals through

object detection techniques for label prediction [35]. Subsequent region-based work delved into modeling spatial dependencies among objects. For instance, Wang et al. [34] proposed a model utilizing spatial transformer layers and Long Short-Term Memory (LSTM) units in order to capture spatial dependencies between different object areas in images. Chen et al. [2] explored semantic interactions between labels by leveraging label co-occurrence. Wu et al. [38] used graph-matching techniques to simultaneously explore spatial associations between instances, semantic dependencies of labels, and the feasibility of instance-label matching.

Recently, many studies have dedicated efforts to capturing relationships between labels. Sequence-based methods analyze and learn semantic associations between label vectors by using Recurrent Neural Networks (RNN), while graph-based approaches capture and utilize label dependencies through Graph Convolutional Networks (GCN). For example, Chen et al. [4] mapped complex label relationship graphs into series of independent label classifiers. Moreover, Wang et al. [33] constructed label graphs by analyzing label co-occurrence information in the data for label representation learning.

The emergence of Vision Transformers (ViT) [6] has introduced a new direction for multi-label classification. Lanchantin et al. [13] developed a framework based on transformer encoders in order to capture complex dependencies between visual features and labels, while Liu et al. [17] explored the use of label embeddings to directly query the presence of labels in images using transformer decoders.

Although the aforementioned multi-label classification methods have made significant progress, particularly in considering spatial dependencies and interactions between labels, there is, however, still a lack of in-depth utilization of structural and contextual information within images. Therefore, our work focuses on further exploring these aspects, aiming to capture rich contextual information by learning and leveraging geometric relationships in images at multiple orders and ranges. Our proposed approach is intended to empower the multi-label classification model with more discriminative image representations.

2.2 Context-Aware Models

The concept of ‘‘Context Awareness’’ has been extensively studied across multiple fields, particularly in computer vision [9, 14, 27], where its applications span a wide extent of applications including object detection and recognition, scene understanding, image segmentation, multi-label classification, etc. Early work on context information primarily focus on integrating local features within images with their surrounding contextual information for object recognition and scene classification. Since the advancement of machine learning techniques, such as random forests and support vector machines (SVM), context-dependent scene modeling has been a major bottleneck in enhancing performances in different classification tasks. For instance, Torralba et al. [21] explored methods to improve object detection accuracy through the use of scene context.

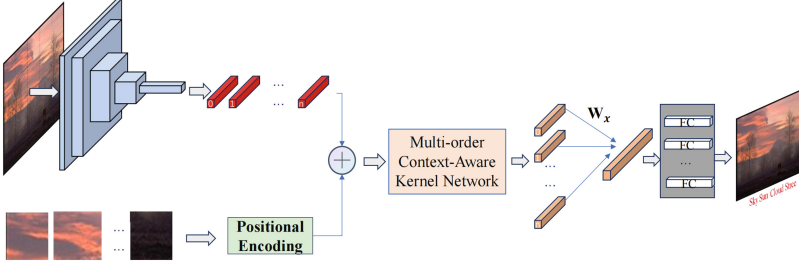


Fig. 1. Deep Multi-order Context-aware Kernel Network framework.

With the advancement of deep learning, context modeling in computer vision has led to a major breakthrough. For instance, the VGG network [29] captures multi-level features of images through convolutional layers, while the Faster R-CNN [22] utilizes a region proposal network to precisely focus on key parts of images, enhancing detection performance. Furthermore, GCN [19, 26, 28] and ViT [6] enhance the processing of context information by capturing relationships between nodes in graphs and by employing global self-attention mechanisms, respectively. Additionally, multi-modal context modeling with the Bilinear Attention Networks (BAN) [12] offers rich scene knowledge for other tasks, namely visual question-answering.

In this paper, we model and learn contextual relationships between image regions at multiple ranges and orders. This modeling leads to better image representations and to conceptually a different approach compared to the related work. This approach is based on learning multiple order similarity kernels whose underlying unfolded networks allow to capture both content and geometric structure (context) in the learned image representations. Our structural relationships in the unfolded networks are dynamically learned end-to-end.

3 Method

In this section, we consider a multi-order neighborhood system that allows integrating contextual information of image regions, and also defining rich and more discriminative image representations. A given image is segmented into a regular grid of cells, with each cell being described with (i) visual features obtained through a pretrained model, and (ii) positional features obtained by encoding their location in images. For each cell, the integrated features are then fed to our proposed deep multi-order context-aware kernel network, which updates cell features by incorporating their first and higher-order neighbors. The overall structure of the network is shown in Fig. 1.

3.1 Context-Aware Kernel Map

For simplicity, we define $\{\mathcal{I}_p\}_{p=1}^P$ as a set of labeled training images, Y_k^P is a binary variable standing for the membership of a given image \mathcal{I}_p to the class

$k \in \{1, \dots, K\}$. $\mathcal{S}_p = \{\mathbf{x}_1^p, \dots, \mathbf{x}_n^p\}$ corresponds to a set of non-overlapping cells extracted from a regular grid in \mathcal{I}_p ; without a loss of generality, n is constant for all images.

The similarity between any two images \mathcal{I}_p and \mathcal{I}_q can be measured by using a convolution kernel:

$$\mathcal{K}(\mathcal{I}_p, \mathcal{I}_q) = \sum_{i,j} \kappa(\mathbf{x}_i^p, \mathbf{x}_j^q), \quad (1)$$

here κ is a positive definite elementary kernel, such as linear, polynomial, and Gaussian, or their linear combinations. These elementary kernels primarily focus on the visual content of the cells within images, ignoring their contextual relationships.

In order to obtain a more relevant similarity, we define a learned context-aware kernel κ (or equivalently its Gram matrix, denoted as \mathbf{K} , where $[\mathbf{K}]_{\mathbf{x}_i, \mathbf{x}_j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ among all cells $\mathcal{X} = \bigcup_p \mathcal{S}_p$). The kernel matrix \mathbf{K} is obtained as [11, 25]

$$\min_{\mathbf{K}} \text{tr}(-\mathbf{K}\mathbf{S}^\top) - \alpha \sum_{c=1}^C \text{tr}(\mathbf{K}\mathbf{P}_c \mathbf{K}^\top \mathbf{P}_c^\top) + \frac{\beta}{2} \|\mathbf{K}\|_2^2, \quad (2)$$

here $\alpha \geq 0$, $\beta > 0$, \mathbf{S} is the similarity matrix of data in \mathcal{X} without context information, \top denotes matrix transpose, and tr denotes the trace operator. The set of matrices $\{\mathbf{P}_c\}_{c=1}^C$ defines the neighborhood relationships between cells (in practice $C = 4$, corresponding to the four directions: up, down, left and right). Specifically, for a given cell \mathbf{x} , if there exists an immediate neighbor \mathbf{x}' in direction c , then $[\mathbf{P}_c]_{\mathbf{x}, \mathbf{x}'} \neq 0$; otherwise $[\mathbf{P}_c]_{\mathbf{x}, \mathbf{x}'} = 0$, $\forall \mathbf{x}' \in \mathcal{X}$. In Eq. (2), the leftmost part is a fidelity criterion that provides high kernel values for visually similar cells pairs $\{(\mathbf{x}_i, \mathbf{x}_j)\}_{ij}$, the second term strengthens or weakens the kernel values between these pairs based on the similarity of their neighborhood, and the right-hand side term acts as a regularizer controlling the smoothness of the learned kernel solution.

One may show that the minimization of Eq. (2) leads to the following recursive solution:

$$\mathbf{K}^{(t+1)} = \mathbf{S} + \gamma \sum_{c=1}^C \mathbf{P}_c \mathbf{K}^{(t)} \mathbf{P}_c^\top, \quad (3)$$

where $\gamma = \frac{\alpha}{\beta}$ controls the impact of context, guaranteeing that learning converges to a stable solution. Equation (3) can be written using an explicit kernel mapping form as

$$\Phi^{(t+1)} = \left(\Phi^{(0)\top} \gamma^{1/2} \mathbf{P}_1 \Phi^{(t)\top} \dots \gamma^{1/2} \mathbf{P}_c \Phi^{(t)\top} \right)^\top, \quad (4)$$

where the matrix $\Phi^{(0)}$ represents either an exact or an approximate kernel mapping of \mathbf{S} , and the matrix Φ is the kernel mapping of \mathbf{K} that needs to be estimated. This update procedure can be reformulated using a fixed feed-forward network structure; each layer in this network corresponds to an iteration: the input layer $\Phi^{(0)}$ is extracted through a pre-trained visual model (such

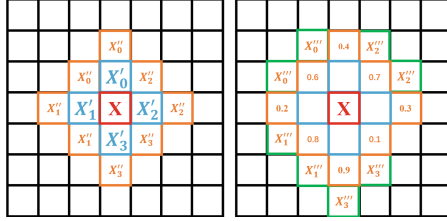


Fig. 2. Multi-order neighborhood system. The left side shows the first-order and second-order neighborhoods. On the right, the third-order neighborhood is built from the second-order neighborhood based on the transition probabilities.

as ResNet101, TresnetL, etc.), and the intermediate hidden layers $\{\Phi^{(t)}\}_t$ are updated through the neighborhood matrices $\{\mathbf{P}_c\}_c$ and the output of the previous layers according to Eq. (3), whilst the final output layer (denoted as $\phi_{\mathcal{K}}(\mathcal{S}_p)$) corresponds to the high-order context-aware features of the image \mathcal{I}_p which are generated by iteratively aggregating multi-order contextual information as described subsequently.

3.2 High-Order Context

In the previous section, we discussed how the first-order neighborhood is leveraged in kernel-based representation learning. For scenes with short range contextual relationships (particularly when individual small objects are considered), first-order neighborhoods are enough. However, for scenes with wider range contextual relationships (for instance when multiple objects co-exist), first-order neighborhoods are not sufficient. As suggested by [10], wider range contexts are involved by extending the scope of first-order neighborhood but it may also significantly increase the dimensionality of the learned representations, and hence the computational cost, and may also potentially introduce excessive noise. Therefore, our proposed contribution relies on higher-order neighborhoods, using random walk and self-attention mechanisms that aggregates more relevant context from a wider range, thus avoiding excessive noise and reducing unnecessary computational burden.

Random walk is able to expand low-order neighborhoods to high-order ones, effectively filtering out noisy image areas. In other words, by moving stochastically through cell neighbors, transitions can be adjusted according to contextual information to balance the exploration of neighboring cells and the exploitation of local ones. This approach prevents getting stuck in noisy cells, and it allows to capture more meaningful contextual information. Multiple independent iterations of random walk tend to retain cells that are frequently visited and more relevant to the content of the central cell, gradually filtering out noisy image areas and enhancing the overall quality of the learned representations.

Formally, given a cell \mathbf{x} , we define its first-order (c -typed) neighborhood $\{\mathcal{N}_c^{(1)}(\mathbf{x})\}$ through the set of matrices $\{\mathbf{P}_c\}_{c=1}^C$ where values of c refer to dif-

ferent types of neighborhoods. For any order $p \geq 2$, the p -th (higher) order neighborhood of \mathbf{x} is recursively defined as

$$\mathcal{N}_c^{(p)}(\mathbf{x}) = \bigcup_{\mathbf{x}' \in \mathcal{N}_c^{(p-1)}(\mathbf{x})} \mathcal{N}_c^{(p-1)}(\mathbf{x}') \quad \text{with } \mathbf{x}' \neq \mathbf{x}. \quad (5)$$

The attention score between \mathbf{x} and any \mathbf{x}'' in $\mathcal{N}_c^{(p)}(\mathbf{x})$ is obtained as

$$\text{score}(\phi(\mathbf{x}), \phi(\mathbf{x}'')) = \text{softmax} \left(\frac{W_q \phi(\mathbf{x})(W_k \phi(\mathbf{x}''))^\top}{\sqrt{d}} \right), \quad (6)$$

where $\phi(\mathbf{x})$ refers to the features of the target cell \mathbf{x} , and $\phi(\mathbf{x}'')$ denotes the features of a cell \mathbf{x}'' within the neighborhood $\mathcal{N}_c^{(p)}(\mathbf{x})$. W_q and W_k are learnable parameter matrices and d corresponds to the dimensionality of the keys.

In order to consider the transition probabilities from the first order neighborhood to the second-order one, we evaluate the probability $p_c^{(p)}(\mathbf{x}, \mathbf{x}'')$ (with $p = 2$) by employing an exponential function to the attention scores and then normalizing the values by summing all the scores in the second-order neighborhood $\mathcal{N}_c^{(2)}(\mathbf{x})$ for the target cell \mathbf{x} and the direction c so that their sum equates 1, i.e.,

$$p_c^{(2)}(\mathbf{x}, \mathbf{x}'') = \frac{\exp(\text{score}(\phi(\mathbf{x}), \phi(\mathbf{x}''))) }{\sum_{z \in \mathcal{N}_c^{(2)}(\mathbf{x})} \exp(\text{score}(\phi(\mathbf{x}), \phi(z)))}. \quad (7)$$

Here $p_c^{(2)}(\mathbf{x}, \mathbf{x}'')$ defines the probability of a random walk from the first-order to the second-order context for cell \mathbf{x} in direction c .

Subsequently, using the aforementioned transition probabilities, we obtain a better estimate of the features while taking into account the second-order neighborhood as

$$\phi_{c,p}(\mathbf{x}) = \sum_{\mathbf{x}'' \in \mathcal{N}_c^{(p)}(\mathbf{x})} p_c^{(p)}(\mathbf{x}, \mathbf{x}'') (W_v \phi(\mathbf{x}')), \quad \text{with } p = 2, \quad (8)$$

where W_v is a learnable parameter matrix used to transform the features; it is multiplied by the transition probabilities to form the second-order (c -typed) contextual features $\phi_{c,2}(\cdot)$.

In order to build higher-order contexts, we employ an iterative method similar to that used for the second-order context, as shown in Fig. 2. However, it is important to note that higher-order contexts involve information with larger distances, and the relevance between a given central cell and the cells in the higher-order neighborhood decreases. Therefore, instead of using all first-order neighboring cells to construct second-order contexts, only a part of cells is chosen in order to build higher-order neighborhoods using transition probabilities.

3.3 Deep Multi-order Context-Aware Kernel Networks

In this section, we build context-aware kernel mapping using the underlying multi-layered networks. This process is achieved iteratively. Below, we detail each step of the network's construction procedure.

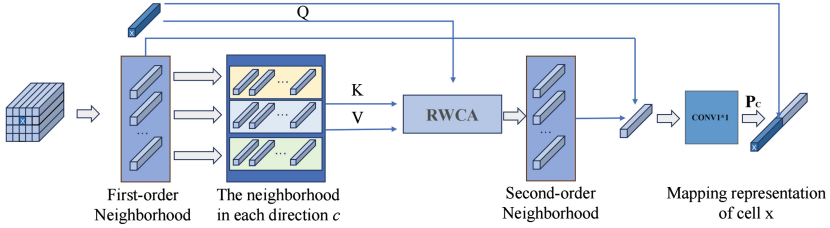


Fig. 3. Details of the Deep Multi-order Context-aware Kernel Network. “RWCA” is the abbreviation of Random Walk and Context Awareness.

In order to capture both content and context of a given cell \mathbf{x} at the t -th layer, we define the multi-order representation as the concatenation of all the orders of contextual features through different directions $c \in \{1, \dots, C\}$ as

$$\phi_c^{(t)}(\mathbf{x}) = \left(\phi_{c,1}^{(t)}(\mathbf{x})^\top \quad \phi_{c,2}^{(t)}(\mathbf{x})^\top \quad \dots \right)^\top, \quad (9)$$

then the representation at the $(t + 1)$ -th layer for a given cell \mathbf{x} is obtained by integrating all the directions as

$$\Phi^{(t+1)} = \left(\Phi^{(t)\top} \quad \gamma^{1/2} \mathbf{P}_1 \Phi_1^{(t)\top} \quad \dots \quad \gamma^{1/2} \mathbf{P}_C \Phi_C^{(t)\top} \right)^\top, \quad (10)$$

being $\phi_c^{(t)}(\mathbf{x})$ a column of $\Phi_c^{(t)}$, and similarly $\phi^{(t)}(\mathbf{x})$ a column of $\Phi^{(t)}$. The details of deep multi-order context-aware kernel network are shown in Fig. 3.

Equation (11) details how, at each layer, the kernel value $[\mathbf{K}^{(t)}]_{\mathbf{x}_i, \mathbf{x}_j}$ between two cells \mathbf{x}_i and \mathbf{x}_j is evaluated by unfolding the map of the kernel as

$$[\mathbf{K}^{(t)}]_{\mathbf{x}_i, \mathbf{x}_j} = \phi^{(t)}(\dots(\phi^{(1)}(\phi^{(0)}(\mathbf{x}_i)))) \cdot \phi^{(t)}(\dots(\phi^{(1)}(\phi^{(0)}(\mathbf{x}_j))))), \quad (11)$$

It’s worth noticing that the dimensionality of $\phi(\mathbf{x})$ increases with deeper networks and concatenation of multi-order contextual features. To address the resulting computational challenge, we introduce 1×1 convolutions at each layer of the context-aware kernel map network for dimensionality reduction. To effectively preserve essential features, we implement a layerwise dimensionality reduction using [15] as

$$\psi_c^{(t)}(\mathbf{x}) = C_t(\phi_c^{(t)}(\mathbf{x})). \quad (12)$$

Here $\phi_c^{(t)}(\mathbf{x})$ and $\psi_c^{(t)}(\mathbf{x})$ respectively refer to the contextual features before and after undergoing dimensionality reduction, and $C_t(\cdot)$ stands for the convolution operation.

Subsequently, Eq. (13) redefines the similarity between two images \mathcal{I}_p and \mathcal{I}_q using this kernel construction (following Eq. (1))

$$\mathcal{K}(\mathcal{I}_p, \mathcal{I}_q) = \sum_{\mathbf{x}_i \in \mathcal{S}_p} \phi^{(t)}(\dots(\phi^{(0)}(\mathbf{x}_i))) \cdot \sum_{\mathbf{x}_j \in \mathcal{S}_q} \phi^{(t)}(\dots(\phi^{(0)}(\mathbf{x}_j))). \quad (13)$$

Eq. (13) reveals the inner product between two recursive kernel mappings, with each one corresponding to an unfolded multi-layered neural network whose feature mappings capture broader contexts as the depth of this network increases. The network’s structure is similar to common deep learning architectures, yet distinct in that the network depth and the number of units per layer are dynamically determined based on the dimensions of the kernel mappings and the number of iterations prior to convergence.

When considering the limit of Eq. (3) as $\tilde{\mathbf{K}}$ and the underlying map in Eq. (4) as $\tilde{\phi}(\cdot)$, the convolution kernel \mathcal{K} between two given images \mathcal{I}_p and \mathcal{I}_q can be expressed as

$$\mathcal{K}(\mathcal{I}_p, \mathcal{I}_q) = \langle \tilde{\phi}_{\mathcal{K}}(\mathcal{S}_p), \tilde{\phi}_{\mathcal{K}}(\mathcal{S}_q) \rangle, \quad (14)$$

$$\tilde{\phi}_{\mathcal{K}}(\mathcal{S}_p) = \sum_{\mathbf{x}_i \in \mathcal{S}_p} w_i \tilde{\phi}(\mathbf{x}_i), \quad (15)$$

being $\{w_i\}_i$ learnable parameters. Hence, each constellation of cells in a given image \mathcal{I}_p can be represented by a deep explicit kernel map $\tilde{\phi}_{\mathcal{K}}(\mathcal{S}_p)$ that aggregates the representation of all the cells in \mathcal{I}_p . In order to explore the full potential of Eq. (14), we consider an end-to-end framework that learns the neighborhood system $\{\mathbf{P}_c\}_c$ within images.

3.4 End-To-End Supervised Learning

We train our context-aware kernel map network (end-to-end) and particularly its underlying contextual parameters, for the task of multi-label classification. Considering N training images $\{\mathcal{I}_p\}_{p=1}^N$ and their category labels Y_p^k , where $Y_p^k = 1$ if \mathcal{I}_p belongs to the k^{th} category, and $Y_p^k = -1$ otherwise. In our kernel map network, we use a fully connected layer for classification. To address class imbalance problem, we consider a grouped training strategy based on label co-occurrence, by training a classification layer for each group and by weighting the underlying losses in order to obtain the total loss; the latter is defined as

$$\min_{\{W_g\}, \{\mathbf{P}_c\}} \frac{1}{2} \sum_{g=1}^G \|W_g\|_2^2 + \sum_{g=1}^G C_g \sum_{p=1}^{N_g} \mathcal{L}_g(W_g \phi_{\mathcal{K}}(\mathcal{S}_p), Y_{p,g}^k), \quad (16)$$

here W_g is the weight matrix for the fully connected layer, $\{\mathbf{P}_c\}_c$ are the learnable context matrices, N_g the size of each group, and C_g corresponds to a hyper-parameter. The total loss includes the weighted sum of cross-entropy losses \mathcal{L}_g and ℓ_2 regularization across category groups. Error backpropagation and gradient descent algorithm are used to update the parameters.

4 Experiment

4.1 Implementation Details

We evaluate our framework on the Core15K and NUS-WIDE benchmarks, which is trained in 200 epochs with an AdamW optimizer, a batch size of 128, and a

Table 1. Comparisons (in percentages) of different methods with ours in terms of Recall (R), Precision (P), and F1 Score (F1) on the Corel5K dataset.

Method	Backbone	CL	R	P	F1
FT DMN+SVM [10]	–	no	38.1	23.4	28.9
CNN-R [20]	–	no	41.3	32.0	36.0
3-layer DKN+SVM [8]	–	no	43.2	25.6	32.1
LNR+2PKNN [40]	–	yes	46.1	44.2	44.9
DCKN [10]	ResNet101	yes	44.4	33.4	38.1
Q2L-TResL [17]	TResNetL	no	48.1	43.5	45.7
DMCKN (ours)	TResNetL	4*5	47.8	43.4	45.5
	TResNetL	8*10	48.3	43.9	45.9
	Cvt-w24	8*10	49.1	45.2	47.0

maximum learning rate of 0.0001. An early stopping strategy is used, with data augmentation techniques such as RandAugment and Cutout, and exponential moving average applied to model parameters with a decay rate of 0.9997.

4.2 Results on Corel5K

The Corel5k dataset comprises 4999 images annotated with 260 concepts. It is split into 4500 training and 500 testing images, with each test image potentially labeled with up to 5 keywords. Performance metrics include average precision (P), recall (R), and F1-score (F1). Images are resized to 400×500 pixels and divided into 4×5 and 8×10 cell configurations for analysis. We use Resnet101, TresnetL [24], and Cvt [36], pre-trained on ImageNet to extract visual features. To address category imbalance, we select all positive instances and three times random subset of negative ones.

Table 1 compares the performance of our model (DMCKN) against other models on the Corel5K dataset. ‘‘CL’’ stands for context learning. The best and second-best performances for each metric are highlighted in red and blue, respectively. The results demonstrate that models with context significantly outperform those without context. Under two different scene configurations (4×5 and 8×10), DMCKN shows superior performance in both configurations. Furthermore, performance on the latest Cvt network architecture further validate our model’s robustness to different backbone networks.

Ablation study. We conducted an in-depth ablation studies to assess the impact of various modules on the performance, focusing on five core components: context awareness, group fully connected layers, context-aware distance, network depth, and the random walk strategy. These experiments adopted ResNet101, pre-trained on the ImageNet dataset, as the feature extractor.

Firstly, we evaluate the effectiveness of the context awareness module and group fully connected layers by comparing model’s performance with and

Table 2. Ablation study of the context-aware module and group fully connected layer on Corel5k dataset.

Method	CA	LG	R	P	F1
Baseline	✗	✗	45.9	38.3	41.7
Ours	✓	✗	47.1	40.2	43.3
	✗	✓	46.3	38.9	42.3
	✓	✓	47.5	40.9	43.9

Table 3. Ablation study of network depth and context-awareness levels: analysis of recall (R), precision (P), and F1 Score (F1) on Corel5k dataset (R/P/F1).

	One-layer	Two-layers	Three-layers
SC	47.1/39.8/43.1	47.5/40.9/43.9	47.9/41.7/44.5
TC	47.4/40.2/43.5	47.9/41.3/44.3	48.3/42.2/45.0

without these modules. Keeping other configurations fixed, we adjust the context-aware distance (second-order and third-order neighborhoods) to explore the specific impact of different context-aware distances on performance. Additionally, we evaluate the model’s performance at different network depths to investigate the impact of network depth on performance. Finally, we also study the role of different random walk strategies in mitigating model noise by using different random walk strategies.

Table 2 shows the results of ablation studies on the context awareness module (CA) and the group fully connected layer (LG) in our model. The results show that the context awareness module increases the performance by 1.2/1.9/1.6, and the group fully connected layer obtained performance gain of 0.4/0.6/0.6. When both are used, there is a significant performance enhancement of 1.6/2.6/2.2, validating the effectiveness of our modules.

Table 3 demonstrates the impact of second-order context awareness (SC) and third-order context awareness (TC) on model performance across different network depths (one-layer, two-layers, and three-layers). The results indicate a gradual improvement in model’s performance with increasing network depth. This suggests that deeper network structures can lead to better performance improvements when considering more profound levels of context information.

Table 4 presents the impact of different random walk strategies (RWG) on model’s performance. Here, ✗ indicates that the random walk strategy was not used to construct higher-order neighborhoods, while different thresholds (*thres*) are investigated for the transition probability of the random walk, used to exclude some unrelated cells, in other words, when $p < thres$, the corresponding cell is dropped. The results demonstrate that the random walk strategy effectively suppresses noise during context aggregation, thereby enhancing model’s performance. Among the strategies, the model performs best with a threshold of 0.67, achieving R, P, and F1 scores of 47.96, 41.32, and 44.39, respectively.

Table 4. Ablation study on the random walk strategy in the Corel5k dataset. *thres* shows the probability threshold.

Method	RWS	R	P	F1
DMCKN(Ours)	X	46.59	39.26	42.61
	<i>thres</i> = 0	48.11	40.98	44.26
	<i>thres</i> = 0.62	48.03	41.21	44.35
	<i>thres</i> = 0.67	47.96	41.32	44.39
	<i>thres</i> = 0.70	47.91	41.33	44.37

Table 5. Comparison with state-of-the-art methods on the NUS-WIDE dataset, where numbers in red indicate the best performance and numbers in blue represent the second-best performance.

Method	Backbone	cells	mAP	CF1	OF1
MS-CMA [39]	ResNet101		61.4	60.5	73.8
SRN [41]	ResNet101		62.0	58.5	73.4
ICME [4]	ResNet101		62.8	60.7	74.1
ASL [23]	ResNet101		65.2	63.6	75.0
Q2L-R101	ResNet101		65.0	63.1	75.0
ML-SGM [37]	ResNet101		64.6	62.4	72.5
SST [3]	ResNet101		63.5	59.6	73.2
SADCL [18]	ResNet101		65.9	63.0	75.0
DMCKN (ours)	ResNet101	4*5	65.4	63.9	74.2
	ResNet101	8*10	66.3	64.6	74.8
Focal loss [16]	TresNetL		64.0	62.9	74.7
ASL	TresNetL		65.2	63.6	75.0
Q2L-TResL	TresNetL		66.3	64.0	75.0
DMCKN (ours)	TresNetL	4*5	66.9	64.5	75.8
	TresNetL	8*10	67.8	65.1	76.5
MITr-l	MITr-l(22k)		66.3	65.0	75.8
Q2L-CvT [17]	CvT-w24		70.1	67.6	76.3
DMCKN (ours)	CvT-w24	4*5	69.4	68.2	76.1
	CvT-w24	8*10	69.7	68.9	76.6

4.3 Results on NUS-WIDE

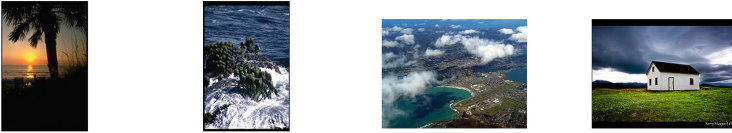
The NUS-WIDE dataset is a widely used benchmark for multi-label image classification, comprising 269,648 Flickr images with 5,018 labels and manually annotated with 81 specific concepts, on average 2.4 concepts per image. According to the official division, 161,789 images are used for training and 107,859 for testing, with small-size images selected for our experiments.

To assess the model’s performance on the NUS-WIDE dataset, we employed metrics such as mean Average Precision (mAP), Composite F1 Score (CF1), and Overall F1 Score (OF1), where higher scores stand for better performance. We resized all the images to 400×500 pixels and segmented context structures based on 4×5 and 8×10 cell grids. For the feature extraction, the pre-trained Resnet101, Tresnet, and CvT models on the ImageNet dataset are used.

Table 5 shows the quantitative results of the proposed method compared to other state-of-the-art on the NUS-WIDE dataset, showing superior performance. With Resnet101, ours achieved improvements of 0.4 and 1.6 in mean Average Precision (mAP) and Composite F1 Score (CF1), respectively. With TresnetL, our model obtained gains of 1.6, 1.1, and 1.5 in mAP, CF1, and Overall F1 Score (OF1), respectively. Further exploration of the potential of our method—employing the latest backbone network (i.e. CvT-w24)—allows us to reach extra gains of 1.3 in CF1 and 0.3 in OF1.



Fig. 4. Image instances of the initial and learned context of higher-order domains on the Corel5K dataset (upper half) and the NUS-WIDE dataset (lower half). From the left to right column: the original images, the initial multi-order neighborhood system, the learned different levels of neighborhoods on the central cell, the impacts of different cells. Warmer color stands for higher impact.



	FC	SC	TC		FC	SC	TC		FC	SC	TC		FC	SC	TC
sky	*	*	*	water	*	*	*	lake				window	*	*	*
sun	*	*	*	sea	*	*	*	grass	*	*	*	building	*	*	*
water	*	*	*	palm	*	*	*	water	*	*	*	grass	*	*	*
tree		*	*	waves	*	*	*	clouds	*	*	*	clouds	*	*	*
palm				coast	*	*	*	sky	*	*	*	sky	*	*	*

Fig. 5. Comparison of image instances of predicted labels and actual labels including FC (First-Order Context), SC (Second-Order Context), and TC (Third-Order Context), the left two images are from the Corel5K dataset and the right two images are from the NUS-WIDE dataset.

4.4 Visualization of Context Impact and Label Prediction

Fig. 4 visualizes the learning effects of our context-aware kernel network on the Corel5K dataset (upper half) and the NUS-WIDE dataset (lower half). The learned multi-order neighbor relationships enhance the focus on visually similar neighboring cells, being capable of capturing more specific and rich contextual information. The final column demonstrates that the network gives higher attention to cells containing both prominent and smaller targets.

Figure 5 shows the prediction results of our network. By learning a multi-order neighborhood system, we more precisely identify the detailed features of targets and effectively capture the overall features of images through the integration of contextual information at various levels, significantly enhancing the accuracy of label predictions.

5 Conclusion

In this work, we introduce a deep multi-order context-aware kernel network to enhance the multi-label image classification task. By leveraging deep contextual modeling, this approach captures intrinsic structural relationships and external connections, significantly improving classification performance. Our framework aggregates multi-order contextual information, providing more refined feature representations for multi-label learning. Experimental results on the Corel5K and NUS-WIDE datasets validate the effectiveness of our method. Future work will focus on modeling label dependencies within our framework and exploring multi-scale approaches for global image representation. We plan to iteratively merge cells through the context-aware kernel network, which is expected to further boost performance.

References

1. Alazaidah, R., Ahmad, F.K.: Trending challenges in multi label classification. *Int. J. Adv. Comput. Sci. Appl.* **7**(10), 127–131 (2016)
2. Chen, T., Wang, Z., Li, G., Lin, L.: Recurrent attentional reinforcement learning for multi-label image recognition. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 32 (2018)
3. Chen, Z.M., Cui, Q., Zhao, B., Song, R., Zhang, X., Yoshie, O.: SST: spatial and semantic transformers for multi-label image recognition. *IEEE Trans. Image Process.* **31**, 2570–2583 (2022)
4. Chen, Z.M., Wei, X.S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5177–5186 (2019)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE (2009)
6. Dosovitskiy, A., et al.: An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929)* (2020)
7. Guo, H., Zheng, K., Fan, X., Yu, H., Wang, S.: Visual attention consistency under image transforms for multi-label image classification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 729–739 (2019)
8. Jiu, M., Sahbi, H.: Nonlinear deep kernel learning for image annotation. *IEEE Trans. Image Process.* **26**(4), 1820–1832 (2017)
9. Jiu, M., Sahbi, H.: Deep representation design from deep kernel networks. *Pattern Recogn.* **88**, 447–457 (2019)
10. Jiu, M., Sahbi, H.: Context-aware deep kernel networks for image annotation. *Neurocomputing* **474**, 154–167 (2022)
11. Jiu, M., Wolf, C., Taylor, G., Baskurt, A.: Human body part estimation from depth images via spatially-constrained deep learning. *Pattern Recogn. Lett.* **50**, 122–129 (2014)
12. Kim, J.H., Jun, J., Zhang, B.T.: Bilinear attention networks. *Adv. Neural Inf. Proce Syst.* **31** (2018)
13. Lanchantin, J., Wang, T., Ordonez, V., Qi, Y.: General multi-label image classification with transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16478–16488 (2021)
14. Li, X., Sahbi, H.: Superpixel-based object class segmentation using conditional random fields. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1101–1104. IEEE (2011)
15. Li, Y., Yang, L.: More correlations better performance: fully associative networks for multi-label image classification. In: *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 9437–9444. IEEE (2021)
16. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988 (2017)
17. Liu, S., Zhang, L., Yang, X., Su, H., Zhu, J.: Query2label: A simple transformer way to multi-label classification. *arXiv preprint [arXiv:2107.10834](https://arxiv.org/abs/2107.10834)* (2021)
18. Ma, L., Sun, D., Wang, L., Zhao, H., Luo, B.: Semantic-aware dual contrastive learning for multi-label image classification. *arXiv preprint [arXiv:2307.09715](https://arxiv.org/abs/2307.09715)* (2023)

19. Mazari, A., Sahbi, H.: Mlgn: Multi-laplacian graph convolutional networks for human action recognition. In: The British Machine Vision Conference (BMVC) (2019)
20. Murthy, V.N., Maji, S., Manmatha, R.: Automatic image annotation using deep learning representations. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pp. 603–606 (2015)
21. Oliva, A., Torralba, A.: Building the gist of a scene: the role of global image features in recognition. *Prog. Brain Res.* **155**, 23–36 (2006)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. *Adv. Neural Inf. Proce. Syst.* **28** (2015)
23. Ridnik, T., et al.: Asymmetric loss for multi-label classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 82–91 (2021)
24. Ridnik, T., Lawen, H., Noy, A., Ben Baruch, E., Sharir, G., Friedman, I.: Tresnet: high performance gpu-dedicated architecture. In: proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1400–1409 (2021)
25. Sahbi, H.: Imageclef annotation with explicit context-aware kernel maps. *Int. J. Multimedia Inf. Retrieval* **4**, 113–128 (2015)
26. Sahbi, H.: Learning laplacians in chebyshev graph convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2064–2075 (2021)
27. Sahbi, H., Li, X.: Context-based support vector machines for interconnected image annotation. In: Asian Conference on Computer Vision, pp. 214–227. Springer (2010)
28. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. *IEEE Trans. Neural Netw.* **20**(1), 61–80 (2008)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
30. Tamura, M., Ohashi, H., Yoshinaga, T.: Qpic: query-based pairwise human-object interaction detection with image-wide contextual information. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10410–10419 (2021)
31. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Proce. Syst.* **30** (2017)
32. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: Cnn-rnn: a unified framework for multi-label image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2285–2294 (2016)
33. Wang, Y., et al.: Multi-label classification with label graph superimposing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12265–12272 (2020)
34. Wang, Z., Chen, T., Li, G., Xu, R., Lin, L.: Multi-label image recognition by recurrently discovering attentional regions. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 464–472 (2017)
35. Wei, Y., et al.: Hcp: a flexible CNN framework for multi-label image classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(9), 1901–1907 (2015)
36. Wu, H., et al.: Cvt: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 22–31 (2021)
37. Wu, Y., Feng, S., Wang, Y.: Semantic-aware graph matching mechanism for multi-label image recognition. *IEEE Trans. Circuits Syst. Video Technol.* (2023)
38. Wu, Y., Liu, H., Feng, S., Jin, Y., Lyu, G., Wu, Z.: Gm-mlic: graph matching based multi-label image classification. arXiv preprint [arXiv:2104.14762](https://arxiv.org/abs/2104.14762) (2021)

39. You, R., Guo, Z., Cui, L., Long, X., Bao, Y., Wen, S.: Cross-modality attention with semantic graph embedding for multi-label classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12709–12716 (2020)
40. Zhang, W., Hu, H., Hu, H.: Neural ranking for automatic image annotation. *Multimedia Tools Appl.* **77**, 22385–22406 (2018)
41. Zhu, F., Li, H., Ouyang, W., Yu, N., Wang, X.: Learning spatial regularization with image-level supervisions for multi-label image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5513–5522 (2017)



Classifier Enhanced Deep Learning Model for Erythroblast Differentiation with Limited Data

Buddhadev Goswami¹, Adithya B. Somaraj¹, Prantar Chakrabarti²,
Ravindra Gudi³, and Nirmal Punjabi^{1,4}

¹ Koita Centre for Digital Health,
Indian Institute of Technology Bombay, Mumbai, India
{buddhadev,adithya,npunjabi}@iitb.ac.in

² Zoho Corporation, Tenkasi, India
prantar@gmail.com

³ Department of Chemical Engineering,
Indian Institute of Technology Bombay, Mumbai, India
ravigudi@iitb.ac.in

⁴ Sensing and Monitoring Foundation, Mumbai, India

Abstract. Hematological disorders, which involve a variety of malignant conditions and genetic diseases affecting blood formation, present significant diagnostic challenges. One such major challenge in clinical settings is differentiating Erythroblast from WBCs. Our approach evaluates the efficacy of various machine learning (ML) classifiers— SVM, XG-Boost, KNN, and Random Forest—using the ResNet-50 deep learning model as a backbone in detecting and differentiating erythroblast blood smear images across training splits of different sizes. Our findings indicate that the ResNet50-SVM classifier consistently surpasses other models' overall test accuracy and erythroblast detection accuracy, maintaining high performance even with minimal training data. Even when trained on just 1% (168 images per class for eight classes) of the complete dataset, ML classifiers such as SVM achieved a test accuracy of 86.75% and an erythroblast precision of 98.9%, compared to 82.03% and 98.6% of pre-trained ResNet-50 models without any classifiers. When limited data is available, the proposed approach outperforms traditional deep learning models, thereby offering a solution for achieving higher classification accuracy for small and unique datasets, especially in resource-scarce settings.

Keywords: Blood cells · Erythroblast detection · Classifiers · SVM · Random Forest · ResNet-50

1 Introduction

Blood disorders in India present unique challenges due to a range of malignant conditions and genetic diseases affecting blood formation. These include

B. Goswami and A. B. Somaraj: Equal contribution.

beta thalassemia, hemophilia, iron deficiency anemia, leukemia, lymphoma, etc. Managing these disorders is particularly difficult in a resource-limited setting like India, where these conditions are more prevalent and socio-economically challenging compared to Western countries [1]. Patients with certain blood disorders may undergo a splenectomy to alleviate symptoms and improve quality of life by preventing excessive destruction of blood cells [3]. This procedure can increase the presence of nucleated red blood cells (NRBCs) in the blood. NRBCs or erythroblasts are immature red blood cell precursors typically confined to the bone marrow and rarely seen in healthy adults, but they may appear more frequently in post-splenectomy patients [4]. Distinguishing NRBCs from lymphocytes in blood smears is challenging due to their similar morphological features. This task is complicated by the variability in lymphocyte appearance and the presence of abnormal cells in hematological disorders. The quality of slide preparation, the microscope's resolution, and the pathologist's expertise are crucial in accurately differentiating these cells [8].

2 Related Work

Das et al. (2016), identified nucleated red blood cells in 50 blood smear images. Their method integrates multilevel thresholding for cell localization, a unique colour space transformation for enhanced contrast between nucleated cells and RBCs, and special fuzzy c-means clustering for segmentation. A random forest classifier discriminates NRBCs from WBCs with an accuracy of 99.42%, offering a significant tool for clinicians diagnosing various anemic conditions efficiently [8].

Fang et al. (2022) in their study present a novel, label-free technique for identifying rare NRBC using deep learning and single-cell Raman spectroscopy. By combining Faster RCNN and YOLOv3 for morphological detection and Raman for verification, it offers rapid, efficient NRBC screening without pre-processing [11]. Alkafrawi et al. (2023) developed an AlexNet-based Convolutional Neural Network model that classifies and counts blood cells in microscopic images with 95.08% accuracy using a dataset of 17,092 blood smear samples. This model showcases the effectiveness of deep learning in medical diagnostics. Additionally, they created a user-friendly GUI, 'Blood Cell Classifier v1.0,' to help hematologists classify blood cells efficiently, illustrating how machine learning can automate traditional manual counting methods. [4]. Rao et al. (2023) proposed EfficientNet - XGBoost framework as a novel method for segmenting and classifying white blood cells (WBCs) from 367 blood smear images. This method uses SegNet for segmentation, EfficientNet for feature extraction, and XGBoost for classification, achieving a higher rank-1 accuracy of 99.02% compared to existing techniques. [24] Chola et al. (2022) proposed that BCNet is a deep learning model aiming to improve accuracy of blood cell classification by identifying multiclass blood cells rapidly and automatically using 17,029 images. The model uses ResNet-18 as the backbone model, achieving 96.78% accuracy [6]. Nozaka et al. (2024) used ResNet models identifying immature granulocytes (IG) and

erythrocytes while screening peripheral blood smears using 6727 images. Deep learning techniques and The findings demonstrated a precision level of 97% for healthy cases and 88% for cases with immature granulocytes. The model for IG recognition, based on CNN, achieved an accuracy rate of 97% for healthy cases and 88% for IG cases [18]

While all the above papers worked on blood cell classification, none focused on data-efficient learning for detection and differentiation of NRBC v/s other WBBCs. Our study addresses this problem and compares various ML classifiers across various splits.

3 Dataset

For this study, the dataset was sourced from the Mendeley repository called ‘A dataset for microscopic peripheral blood cell(PBC) images for development of automatic recognition systems [2]. This dataset comprises excellent digital images of typical peripheral blood cells. The images were obtained using the CellaVision DM96 analyzer at the Hospital Clinic of Barcelona after conducting cell preparation and staining procedures with the Sysmex SP1000i and May Grünwald-Giemsa stain, respectively. The dataset comprises 17,092 JPEG images with dimensions of 360×363 pixels. It is organized into eight distinct categories of blood cells: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (including metamyelocytes, myelocytes, and promyelocytes), erythroblasts, and platelets as shown in Fig. 1.

This study specifically required a dataset containing erythroblasts and lymphocytes for effective differentiation. The Mendeley dataset uniquely meets this requirement, as other accessible datasets like LISC [21] or ALL-IDB [16] do not include the erythroblast class essential for our analysis.

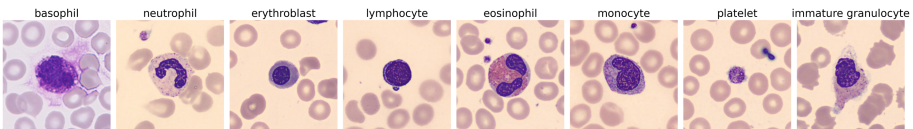


Fig. 1. Sample images from each class of the dataset.

4 Experimental Methodology

All the experiments were conducted on a single Nvidia GeForce RTX-3050 GPU device with 8 GB of RAM. We have used a batch size of 64 images, leveraging CUDA libraries for optimized performance.

4.1 Backbone Selection

A range of pre-trained convolutional neural network architectures were selected for our classification tasks, including ResNet-50 [12], VGG19 [23], ResNet-18, VGG16, and InceptionV3 [25]. The dataset was divided into 70% for training, 15% for validation, and 15% for testing. All models underwent a 15-epoch training regimen, except InceptionV3, which required 30 epochs due to its more complex structure. According to the initial evaluation results presented in Table 1, ResNet-50 demonstrated the highest testing accuracy at 98.72%, leading us to select it as our primary feature extraction backbone. Unlike the VGG models, ResNet-50 features residual connections that mitigate the vanishing gradient problem, facilitating more efficient training and enabling the construction of deeper, more effective networks [12]. As seen in Table 1, ResNet-50 provides highest accuracy with second lowest training time. ResNet-50’s optimal balance of depth and computational efficiency allows it to handle complex features more effectively than ResNet-18 while avoiding the greater computational demands of higher ResNet models. Additionally, ResNet-50’s extensive availability of pre-trained models makes it highly suitable for efficient transfer learning and deployment. In contrast to newer models like EfficientNet [24], DenseNet, and Vision Transformers (ViT), ResNet-50 remains advantageous for several reasons. EfficientNet, while highly efficient due to compound scaling, presents complexity in understanding and implementation. DenseNet’s dense connectivity enhances feature reuse and gradient flow but incurs higher memory and computational costs [15]. ViT excels in capturing long-range dependencies and performs well on large-scale datasets but demands extensive data and computational resources [10]. Despite the advancements in these newer models, ResNet-50 remains a balanced choice, offering robustness, efficiency, and accessibility for diverse real-world applications.

Table 1. Model performance comparison for backbone selection with 70:15:15 splits

Model	Test Accuracy	Precision	Recall	F1 Score	Trainable Parameters	Total Training Time
ResNet-50	98.72%	0.9892	0.9874	0.9882	25 million	23 m 24 s
VGG-16	98.67%	0.9826	0.9834	0.9832	138 million	47 m 33 s
ResNet-18	98.57%	0.9862	0.9868	0.9892	11 million	14 m 15 s
VGG-19	98.44%	0.9852	0.9864	0.9854	143 million	53 m 33 s
InceptionV3	92.38%	0.9179	0.9171	0.9186	23 million	37 m 56 s

4.2 Training on ResNet-50 Architecture

ResNet-50 architecture, tailored for image classification, was implemented using the PyTorch library for building the model, and torch-vision was used for data pre-processing. The dataset was organized into distinct classes and divided into

training, validation, and testing. The testing set was fixed with 4000 images, and the other sets were varied as per the data split.

In this setup, the ResNet-50 model, initially trained on the ImageNet [9] dataset, was fine-tuned by adjusting its final layers for blood cell classification. The cross-entropy loss function directs the training process is given in Eq. 1

$$L(y, \hat{y}) = - \sum y \log(\hat{y}) \quad (1)$$

which is the most effective choice for multi-class problems. This loss function evaluates the model’s output by comparing it to the actual data labels. The Adam optimizer is utilized for optimization, as it is known for its efficiency in adaptive updating network weights.

Before we began the training process, the optimal learning rate was determined by gradually increasing the learning rate over a predefined range. At each iteration, the training loss was recorded, and we chose the optimal learning rate. This is the point at the middle of the steepest downward curve, just before divergence. Following this, we implemented discriminative fine-tuning [14] and assigned higher learning rates to later layers in the model, while earlier layers have progressively lower learning rates. Using the optimal learning rate identified earlier as the maximum for the final layer, we applied a one-cycle learning rate scheduler. This scheduler starts with a small initial learning rate, increases it to the maximum, and then decreases it to a final value lower than the initial rate. Then, we trained our model using k-fold cross-validation with a 5-fold configuration over 15 epochs. Each fold of the validation data is used once as a test set, while the entire training dataset is used to train the model in each epoch. The validation indices are shuffled to randomize the data selection, ensuring unbiased validation subsets. The model is then evaluated on each validation subset, with performance metrics such as loss and accuracy (both top-1 and top-5) calculated and stored. After evaluating all folds, the mean and standard deviation of these metrics are computed to assess overall model performance and consistency across different subsets. If the average validation loss of a fold is lower than previously recorded, the model’s state is saved. This process helps in selecting the model configuration that generalizes best on unseen data.

4.3 Classifiers

A hybrid methodology that combines deep learning and traditional machine learning techniques to tackle image classification tasks was adopted. The computational capabilities of PyTorch library [19], supplemented by algorithms from Scikit-learn [20] enhanced efficiency.

A pre-trained ResNet-50 model was utilized and adapted for our classification task. The pre-trained network acts as a feature extractor where the initial layers capture generic features (edges, textures) while deeper layers identify more complex patterns relevant to the specific classes in our dataset. The last fully connected layer, reformulated for our needs, transforms these features into class probabilities using the softmax function shown in Eq. 2

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2)$$

where z_i are the logits (i.e., unnormalized log probabilities) produced by the last network layer for each class, and K is the total number of classes. Throughout the training and evaluation phases, we meticulously monitored various performance metrics, such as accuracy, precision, recall, and F1-score, to fine-tune and evaluate our models. The dataset was systematically sorted into training, validation, and testing directories. Preprocessing is done, which included uniform image transformations such as resizing to 224×224 pixels, center cropping, converting images into tensor format, and normalizing based on the mean (μ) and standard deviation (σ) values derived from the ImageNet dataset [9] shown in Eq. 3.

$$\text{Normalized} = \frac{\text{Image} - \mu}{\sigma} \quad (3)$$

where $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$ (values obtained from ImageNet). These preprocessing steps were essential for preparing the data for optimal processing through neural network architectures. Additional preprocessing or image augmentation was not performed due to the nature of the data, as it is a well-curated dataset.

Our modeling strategy involved dual approaches: fine-tuning traditional machine learning models (KNN [7], SVM [13], RandomForest [17], XGBoost [5]) and adapting a deep learning model. We employed grid search [22] to optimize the hyperparameters of the traditional models, shown in Eq. 4.

$$\text{GridSearch} = \arg \max_{\theta} \left(\sum_{i=1}^n \text{Accuracy}(\theta_i) \right) \quad (4)$$

where θ represents the set of parameters over which the search is conducted, and n is the number of parameter combinations tested. This exhaustive parameter optimization ensured that our models were highly tailored to maximize accuracy on our specific dataset. Figure 2 shows our proposed Architecture for the study.

The two-step process of feature extraction followed by classifier implementation, though time-consuming, is crucial for maximizing accuracy with limited data. This method and the application of grid search for optimizing classifier parameters extend the training time. However, these steps are essential for achieving the high precision necessary in clinical applications, especially under conditions of data scarcity. Despite the potential increase in training time due to these methods, the GPU requirements do not substantially change. The computational resources required remain consistent with tasks, making our approach feasible within typical clinical research settings.

4.4 Training with Classifiers

To assess our classifiers’ performance under limited training data conditions, we devised a series of experiments with varied training set sizes, ensuring that each class was represented equally across all partitions. Table 2 provides the information above the dataset distribution.

5 Results

The performance of ResNet-50, both with and without classifiers like SVM, XG-Boost, KNN, and Random Forest, is compared in terms of test accuracy and erythroblast identification across different training dataset sizes, as shown in Table 3. Remarkably, ResNet-50-KNN outshone ResNet-50 at very low data splits in test accuracy, securing 85.88% with just 1% of data. This performance

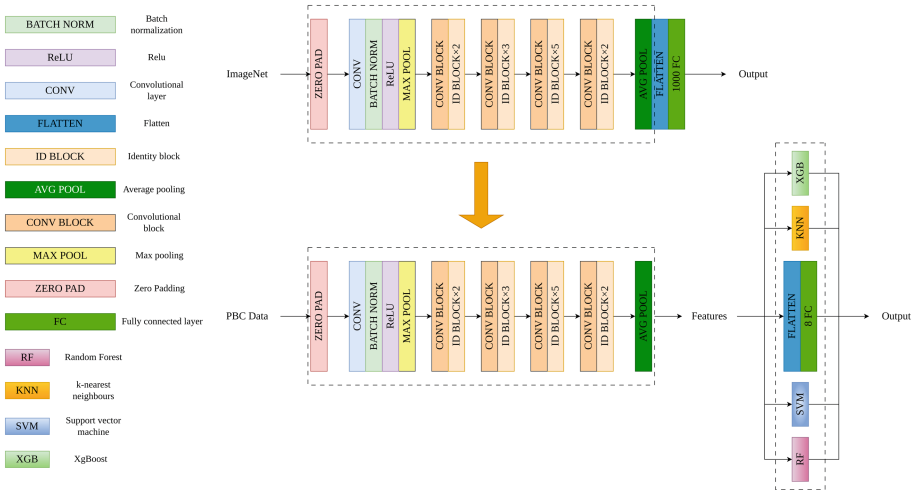


Fig. 2. Our proposed classifier enhanced ResNet-50 model architecture for the study.

Table 2. Distribution of Training, Validation, and Testing Images

Training Percentage	Training Images	Validation Images	Testing Images
1%	168	12,924	4,000
2.5%	424	12,668	4,000
5%	848	12,244	4,000
7.5%	1,273	11,819	4,000
10%	1,709	11,383	4,000
20%	3,418	9,674	4,000
30%	5,128	7,964	4,000

Table 3. Detailed Test Performance Metrics Across Various Classifiers with a ResNet-50 Backbone

Data Split	Classifier	Test Acc (%)	Erythroblast (Prec./Rec./ F1)
1% Data Split			
1%	ResNet50	82.03	0.986 / 0.857 / 0.917
1%	ResNet50-XGBoost	84.05	0.986 / 0.857 / 0.917
1%	ResNet50-KNN	85.88	0.984 / 0.819 / 0.894
1%	ResNet50-SVM	86.75	0.989 / 0.812 / 0.892
1%	ResNet50-RandomForest	84.68	0.993 / 0.763 / 0.863
2.5% Data Split			
2.5%	ResNet50	86.25	0.990 / 0.794 / 0.881
2.5%	ResNet50-XGBoost	86.82	0.988 / 0.844 / 0.910
2.5%	ResNet50-KNN	87.55	0.976 / 0.882 / 0.926
2.5%	ResNet50-SVM	88.02	0.976 / 0.900 / 0.937
2.5%	ResNet50-RandomForest	87.20	0.986 / 0.866 / 0.922
5% Data Split			
5%	ResNet50	92.99	0.979 / 0.928 / 0.953
5%	ResNet50-XGBoost	92.77	0.969 / 0.928 / 0.948
5%	ResNet50-KNN	93.05	0.949 / 0.926 / 0.937
5%	ResNet50-SVM	93.00	0.953 / 0.926 / 0.939
5%	ResNet50-RandomForest	91.90	0.981 / 0.924 / 0.952
7.5% Data Split			
7.5%	ResNet50	96.29	0.984 / 0.970 / 0.977
7.5%	ResNet50-XGBoost	95.40	0.878 / 0.980 / 0.926
7.5%	ResNet50-KNN	95.93	0.990 / 0.958 / 0.974
7.5%	ResNet50-SVM	96.07	0.988 / 0.960 / 0.974
7.5%	ResNet50-RandomForest	95.23	0.986 / 0.952 / 0.969
10% Data Split			
10%	ResNet50	96.14	0.984 / 0.972 / 0.978
10%	ResNet50-XGBoost	95.87	0.984 / 0.958 / 0.971
10%	ResNet50-KNN	96.25	0.988 / 0.958 / 0.973
10%	ResNet50-SVM	96.00	0.990 / 0.950 / 0.969
10%	ResNet50-RandomForest	95.70	0.972 / 0.964 / 0.968
20% Data Split			
20%	ResNet50	97.66	0.986 / 0.974 / 0.980
20%	ResNet50-XGBoost	96.25	0.980 / 0.958 / 0.969
20%	ResNet50-KNN	97.48	0.967 / 0.988 / 0.977
20%	ResNet50-SVM	97.30	0.956 / 0.990 / 0.973
20%	ResNet50-RandomForest	96.55	0.986 / 0.974 / 0.980
30% Data Split			
30%	ResNet50	98.36	0.982 / 0.996 / 0.989
30%	ResNet50-XGBoost	97.88	0.990 / 0.996 / 0.993
30%	ResNet50-KNN	98.45	0.980 / 0.996 / 0.988
30%	ResNet50-SVM	98.42	0.982 / 0.996 / 0.989
30%	ResNet50-RandomForest	98.00	0.976 / 0.994 / 0.985

advantage was maintained as dataset sizes increased. In the critical task of erythroblast identification, which is key for precise early detection of red blood cells, ResNet-50-KNN achieved a remarkable precision of 0.989 and a recall of 0.819 at the same minimal data size, aligning closely with ResNet-50's metrics.

As the data size expanded, all classifiers saw enhancements in erythroblast accuracy, with ResNet-50-SVM and Random Forest demonstrating notable proficiency; both achieved F1-scores exceeding 0.973 and 0.980, respectively, at a 20% data size. ResNet-50-XGBoost exhibited sturdy performance, attaining a test accuracy of 96.25% at a 20% data size. ResNet-50-KNN also displayed substantial gains as data volume grew, offering accuracy comparable to other models and erythroblast metrics.

ResNet-50-SVM outperforms the ResNet-50 model in low data and show consistent performance Fig. 3.

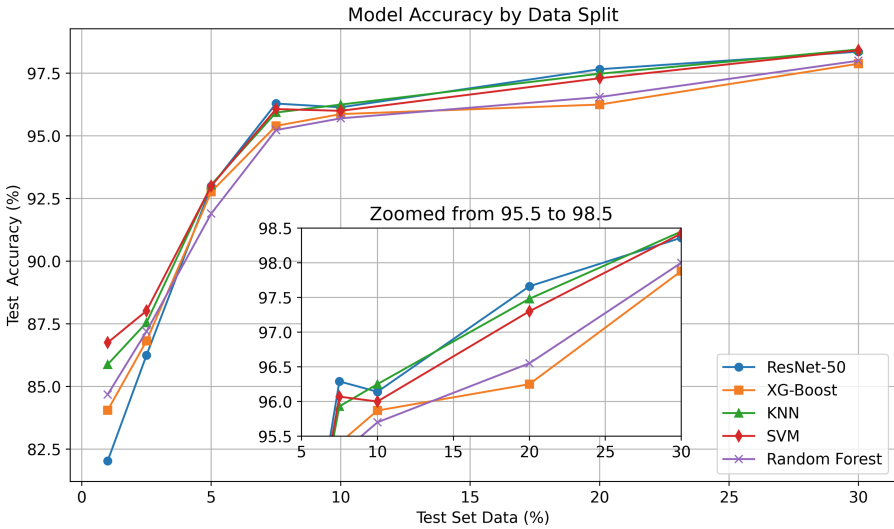


Fig. 3. Test accuracy vs. percentage of the test data for the various models. Inset shows the differences between the model for higher test data

6 Discussion

Collaboration with clinical hematologists and pathologists has improved our understanding of the capabilities and limitations of machine learning (ML) algorithms for blood cell classification. These models, notably the one with Support Vector Machine (SVM) as the classifier, achieved a test accuracy of 86.75% using only 1% of the available data. This success indicates ML's potential to enhance hematological diagnostics, which is critical for early detection and management of diseases like leukemia and anemia.

Utilizing SVM or alternative machine learning classifiers like KNN with features derived from ResNet-50 results produces superior outcomes in low training data than a standard pre-trained ResNet-50 model. ResNet-50 offers superior features that can efficiently be utilized by SVMs, enabling thorough customization of the dataset's unique characteristics through fine-tuning of hyperparameters and kernel selection. Support Vector Machines (SVMs) adaptability allows them to match the data more effectively than the fixed, fully connected layers commonly present in pre-trained models. Moreover, utilizing ResNet-50 mainly for extracting features minimizes the likelihood of overfitting, particularly in datasets with limited size. SVMs exhibit improved generalization capabilities on unfamiliar data due to their controlled training dynamics. Integrating ResNet-50's advanced deep learning capabilities with the precise adaptability of machine learning classifiers such as SVMs significantly enhances performance.

However, in conventional clinical settings, these models face challenges due to biological variability and limitations in training datasets. Performance issues often arise with images of overlapping cells, a common scenario in clinical samples. This results in frequent misclassifications and underscores the need for advanced image segmentation algorithms to isolate individual cells effectively. Additionally, the models were trained on highly cropped and zoomed images, further limiting their application to typical clinical images. Variability in staining methods and slide quality also affect model performance, as these factors can alter the appearance of cells on slides.

The model demonstrates high accuracy in detecting erythroblasts in cases where there is a solitary cell with a clearly defined ratio between the nucleus and cytoplasm in the image, as shown in Fig. 4. However, it faces difficulties in accurately categorizing erythroblasts in situations where there are multiple cells in a single image, the ratio of nucleus to cytoplasm is low, and small cells like platelets are next to red blood cells (giving the appearance of a single cell), or other cells show visible cytoplasm as shown in Fig. 5.

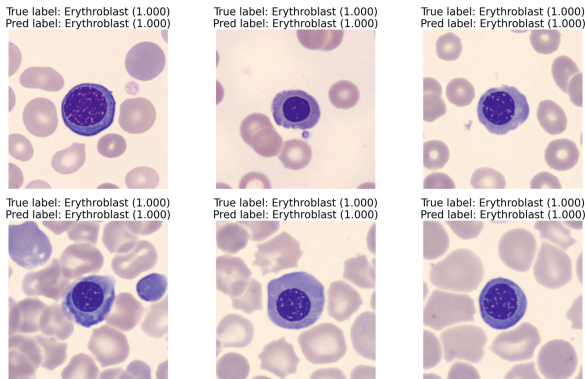


Fig. 4. Correctly predicted labels of erythroblast

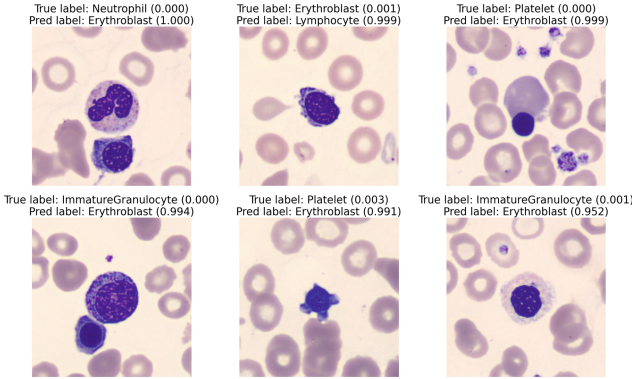


Fig. 5. Incorrect predicted labels of erythroblast

For the following work, there will be two models in conjunction. The first will segment individual cells from complex histopathology images. The proposed architecture in this study will help classify individual cells to further improve the performance. There is a need to expand the dataset diversity, and training models on comprehensive image data from patients with hematological disorders is essential. Developing a user-friendly interface that normalizes erythroblast counts relative to other white blood cells (WBCs) through ratio calculations or percentage conversions will facilitate easier comparisons across samples. These advancements will enhance model accuracy and robustness, reduce pathologists' workloads, and improve diagnostic processes, bridging the gap between theoretical precision and practical usability in clinical settings.

7 Conclusion

This study highlights the significant impact of machine learning models in hematological diagnostics. It showcases the ability of algorithms, such as SVM and ResNet-50, to accurately classify blood cells with limited images available for training, emphasizing their transformative potential. Although controlled testing environments have achieved high accuracy, real-world clinical applications face significant challenges due to biological variability and limitations in dataset diversity. The highlighted concerns encompass the potential for misclassification due to the proximity of cells in blood smears, as well as the influence of inconsistent staining quality on the model's performance. To tackle these challenges, it is necessary to improve the representativeness of the dataset and refine image processing techniques to ensure that the model performs well in different clinical settings.

Furthermore, the need to train these models with a limited amount of data brings attention to wider concerns regarding fairness in global health—specifically, the challenges imposed by limited resources in regions such as India. In such contexts, efficient models necessitating less data are crucial, as they

provide scalable solutions that rapidly adjust to various medical environments without requiring extensive computational resources.

Acknowledgements. We thank Dr Pronati Gupta, Consultant Hematopathologist, Chittaranjan National Cancer Institute, Kolkata, for her feedback.

Code and Data Availability. The code and data used in this study are available in this <https://github.com/MicroBuddha/Erythroblast.git>.

References

1. Abbas, K., Banks, J., Chandran, V., Tomeo-Reyes, I., Nguyen, K.: Classification of white blood cell types from microscope images: techniques and challenges, 17–25 (2018)
2. Acevedo, A., Merino, A., Alférez, S., Molina, Á., Boldú, L., Rodellar, J.: A dataset for microscopic peripheral blood cell images for development of automatic recognition systems **1** (2020). <https://doi.org/10.17632/snkd93bnjr.1>, <https://data.mendeley.com/datasets/snkd93bnjr/1>, publisher: Mendeley Data
3. Adams, C.D., Kessler, J.F.: Circulating nucleated red blood cells following splenectomy in a patient with congenital dyserythropoietic anemia. *Am. J. Hematol.* **38**(2), 120–123 (1991). <https://doi.org/10.1002/ajh.2830380209>
4. Alkafrawi, I.M.I., Dakhell, Z.A.: Blood cells classification using deep learning technique. In: 2022 International Conference on Engineering & MIS (ICEMIS), pp. 1–6 (2022). <https://doi.org/10.1109/ICEMIS56295.2022.9914281>, <https://ieeexplore.ieee.org/document/9914281>
5. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 16, ACM (2016). <https://doi.org/10.1145/2939672.2939785>
6. Chola, C., et al.: BCNet: a deep learning computer-aided diagnosis framework for human peripheral blood cell identification. *Diagnostics* **12**(11), 2815 (2022), <https://www.mdpi.com/2075-4418/12/11/2815>, number: 11 Publisher: Multidisciplinary Digital Publishing Institute
7. Cunningham, P., Delany, S.J.: k-nearest neighbour classifiers - a tutorial. *ACM Comput. Surv.* **54**(6), 1–25 (2021). <https://doi.org/10.1145/3459665>
8. Das, R., Ahluwalia, J., Sachdeva, M.U.S.: Hematological practice in India. *Hematol. Oncol. Clin. North Am.* **30**(2), 433–444 (2016). 10.1016/j.hoc.2015.11.009, <https://www.sciencedirect.com/science/article/pii/S0889858815001963>
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
10. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale (2021). <https://arxiv.org/abs/2010.11929>
11. Fang, T., et al.: Fast label-free recognition of NRBCs by deep-learning visual object detection and single-cell Raman spectroscopy. *Analyst* **147**(9), 1961–1967 (2022). 10.1039/D2AN00024E, <https://pubs.rsc.org/en/content/articlelanding/2022/an/d2an00024e>, publisher: The Royal Society of Chemistry
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition (2015). <https://doi.org/10.48550/arXiv.1512.03385>, <http://arxiv.org/abs/1512.03385>, arXiv:1512.03385 [cs]

13. Hearst, M., Dumais, S., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. *IEEE Intell. Syst. Appl.* **13**(4), 18–28 (1998). <https://doi.org/10.1109/5254.708428>
14. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification (2018). <https://arxiv.org/abs/1801.06146>
15. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks (2018). <https://arxiv.org/abs/1608.06993>
16. Labati, R.D., Piuri, V., Scotti, F.: All-idb: The acute lymphoblastic leukemia image database for image processing. In: 2011 18th IEEE International Conference on Image Processing, pp. 2045–2048 (2011). <https://doi.org/10.1109/ICIP.2011.6115881>
17. Louppe, G.: Understanding random forests: From theory to practice (2015)
18. NOZAKA, H., KUSHIBIKI, M., KAMATA, K., YAMAGATA, K.: Approach to recognition of immature granulocytes using deep learning in peripheral blood smear screening: the potential of AI models using a convolution neural network for blood cell morphology classification. *Japanese J. Med. Technol.* **73**(1), 69–77 (2024). <https://doi.org/10.14932/jamt.23-72>
19. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library (2019)
20. Pedregosa, F., et al.: Scikit-learn: machine learning in python (2018)
21. Rezatofghi, S.H., Soltanian-Zadeh, H.: Automatic recognition of five types of white blood cells in peripheral blood. *Comput. Med. Imaging Graph. Official J. Comput. Med. Imaging Soc.* **35**(4), 333–343 (2011). <https://doi.org/10.1016/j.compmedimag.2011.01.003>
22. Shekar, B.H., Dagnev, G.: Grid search-based hyperparameter tuning and classification of microarray cancer data. In: 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), pp. 1–8 (2019). <https://doi.org/10.1109/ICACCP.2019.8882943>
23. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (2015). <http://arxiv.org/abs/1409.1556>, [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) [cs]
24. SivaRao, B.S.S., Rao, B.S.: EfficientNet - XGBoost: an effective white-blood-cell segmentation and classification framework. *Nano Biomed. Eng.* **15**(2), 126–135 (2023). <https://doi.org/10.26599/NBE.2023.9290014>, <https://www.sciopen.com/article/10.26599/NBE.2023.9290014>
25. Szegedy, C., et al.: Going Deeper with Convolutions (2014). <https://doi.org/10.48550/arXiv.1409.4842>, <http://arxiv.org/abs/1409.4842>, [arXiv:1409.4842](https://arxiv.org/abs/1409.4842) [cs]



PiExtract: An End-to-End Data Extraction Pipeline for Pie-Charts

Muhammad Suhaib Kanroo¹, Hadia Showkat Kawoosa¹, Joy Dhar¹,
and Puneet Goyal^{1,2}

¹ Indian Institute of Technology Ropar, Rupnagar 140001, Punjab, India
puneet@iitrpr.ac.in

² NIMS Institute of Engineering and Technology, NIMS University, Jaipur 303121,
Rajasthan, India

Abstract. Charts are important non-textual elements present in documents, providing a visual representation of numerical data. Among different representations, Pie-charts are commonly employed in digital documents due to their perceptual advantages for displaying numerical data and inter-relationship information. Chart Data Extraction is a multi-stage pipeline, with each stage playing a crucial role in obtaining the raw data correctly. Prior work mostly focuses on improving the performance of one or a combination of a few sub-stages. In this work, we propose a novel end-to-end data extraction algorithm, PiExtract, to extract data from pie-charts. This proposed algorithm designs a novel Robust Fusion Attention Network (RobFA-Net) approach for chart classification tasks. This network introduces a robust fusion attention strategy to learn significant discriminative global and local information, thereby enhancing the learning model performance. In addition, our novel rule-based sector data extraction method further enhances its performance in extracting data from pie-charts. Extensive experimentation is conducted on three datasets, specifically Revision, Chagas, and FigureQA, focusing on chart classification and the FigureQA dataset for data extraction from pie-charts. Our findings demonstrate that the proposed pipeline outperforms compared to previous works, showcasing superior performance.

Keywords: Classification · Chart Data Extraction · Pie-chart · computer vision · Attention

1 Introduction

Charts provide a compact summary of this generated data and are widely used by scientific and business communities [1–4]. While charts interpret the data more intuitively and objectively, such depictions are not meant for machine consumption. Chart analysis by machines is important for indexing and reusability. People suffering from visual impairment and other cognitive diseases can also

M. S. Kanroo and H. S. Kawoosa—Both authors have equally contributed to this work.

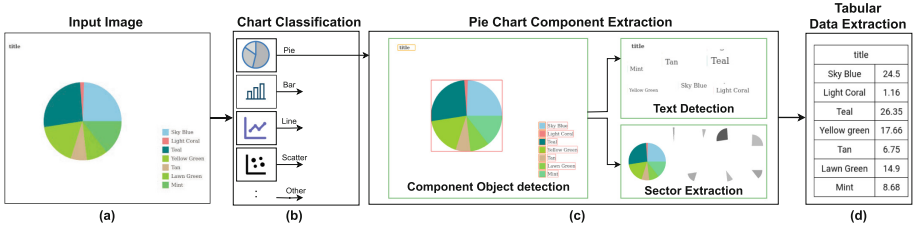


Fig. 1. Block diagram of an end-to-end data extraction pipeline for pie-charts.

benefit from the machine readability of charts. Chart Data Extraction (CDE) refers to the mechanism of extracting the raw data content stored in the charts. Data extraction from charts is a multi-stage pipeline, as shown in Fig. 1, with each stage playing a crucial role in obtaining the raw data correctly [2, 4]. The chart classification is considered the first stage in the pipeline. The next step is the component extraction, which includes text detection, text recognition, etc., in the chart. The final step in CDE is the mapping of all detected elements and subsequent data extraction of raw data.

Chart classification is the primary and important stage of the pipeline. Accurate data extraction is only possible if the chart image has been correctly classified. The complexity of the task is increased due to the wide variety of chart images for classification. Other problems include noise, inter-class similarity, and intra-class variation. While substantial research has been conducted on various chart types such as bar and line graphs [2, 5], there exists a paucity of literature [6, 7] addressing the extraction of data from pie-charts mainly due to the scarcity of datasets containing ground truth for pie chart data. Unlike other chart types, sector information is processed for efficient automated extraction of underlying numeric data embedded in pie-charts. Existing works rely on the image processing techniques of obtaining pixel count for each sector to obtain its percentage value. Since the percentage of a particular sector is dependent on the total pixel count of all sectors, inaccurate pixel count in any sector affects all the sectors. Thus, prior works enable error propagation in data extraction in pie-charts. Another approach to obtain the percentage value of each sector is by calculating the central angle of each sector. However, it is important to estimate angles precisely to capture the relative comparison correctly; however, acute angles, especially smaller ones, are difficult to estimate.

Most literature usually focuses on improving the performance of one or a combination of a few sub-stages of the pipeline like chart classification, data extraction, sector extraction, etc. [2]. We propose an end-to-end data extraction pipeline for the pie chart, where the input is an image and the output is the two-column table, one column containing the legend label and the other column containing its data value. Extensive experimentation is performed on three datasets, namely Revision [8], Chagas [9], and FigureQA [10] for chart classification and the pie-chart subset of the FigureQA dataset for data extraction. The

results demonstrate superior performance by the proposed pipeline than prior works on these datasets. The main contributions of our paper are as follows:

- We propose a novel end-to-end data extraction algorithm, PieExtract, specifically for pie-charts. The algorithm takes an image as an input and presents the output in a tabular form.
- We propose the Robust Fusion Attention Network (RobFA-Net) approach for chart classification tasks. This network introduces a global-local attention strategy called Robust Fusion Attention (RobFA) to learn discriminative global and local information, thereby enhancing the learning model performance.
- We employ YOLOv9 [11] for Object Detection and CRNN [12] for text recognition. Furthermore, we propose a novel rule-based sector extraction module for the extraction of data values from pie-charts.
- We conducted extensive experimentation on three datasets, specifically Revision [8], Chagas [9], and FigureQA [10], with a focus on chart classification and the FigureQA dataset for data extraction from pie-charts. Our findings demonstrate that the proposed pipeline outperforms previous works on these datasets, showcasing superior performance.

The rest of the paper is organized as follows. Section 2 presents the overview of the related work. Section 3 presents a Methodology of the proposed pipeline. Section 4 demonstrates various experimental results. Finally, we conclude the paper in Sect. 5.

2 Related Work

Primary research in charts has been on the classification stage of CDE. Most of the prior works use fine-tuned state-of-art models like AlexNet, MobileNet, VGG16 etc., for chart classification task [13, 14, 16, 17]. ResNet with Adam optimizer is used as a feature extraction network [7] with 512×512 input image size. DenseNet121 [18] for feature extraction is used along with the Squeeze Excitation (SE) module [19] and also performance improvement is observed by tuning hyper parameters [3]. Use of dilation mechanism in context module at back-end improved the performance of DenseNet121 [20]. XceptionNet [21] performs better than other competing models on both synthetic and real world datasets [22]. Lightweight mobilenetv2 [23] along with hyper parameter tuning displays improved performance in chart classification [24].

Text detection, recognition & role classification plays a vital role in mapping text to its corresponding numeric value. Mostly object detection techniques are utilised for detection of texts [7, 19]. Trained EAST [25] is employed for obtaining the coordinates of the text present in the chart images [26]. Features extracted from the Residual network are fed to detection head [2] for text detection. For recognising the text, widely used OCR solutions are utilised such as Tesseract OCR¹ and microsoft azure². Initially, heuristic based approach were employed

¹ <https://opensource.google.com/projects/tesseract>.

² <http://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/>.

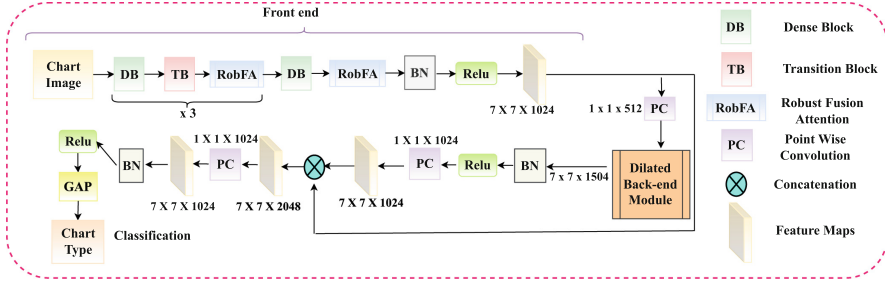


Fig. 2. An overview of the proposed Robust Fusion Attention Network (RobFA-Net) composed of Dense blocks, Transition blocks and RobFA blocks.

to recognise the text [27]. CRNN based methods are also employed to recognise the text from the charts [2]. Attention based models ensemble with CRNN [12] recognise text in an improved way [19]. Correct understanding role of each data is vital for obtaining the data correctly. Correctly obtaining the location of text is important in determining its role correctly [2]. SVM is utilised for role classification after feature vectors are obtained from geometrical properties of the text [7, 17]. Cascade RCNN predicts the class of each text in chart [26].

For data extraction from the pie chart images, pixel count of each sector plays an important role. Pre-processing of the images is performed followed by connected component analysis to determine the pixel count of each slice [6]. After detecting the legend label, color for legend mark is obtained by distance metric. Pixel count of gray scale of that color determines the percentage of the sector with that color [7]. Centre & border points in the pie chart are obtained by morphological operations followed by Canny edge detection and Hough transform. User intervention is required to correctly obtain centre and corner points [27].

3 Methodology

3.1 Chart Classification

In this section, we describe our proposed approach, RobFA-Net, tailored for pie chart classification tasks. Our method comprises two key modules: a front-end and a dilated back-end, similar to the architecture of the existing method outlined in [20]. These modules work cohesively to attain robust performance by extracting salient meaningful representations. However, our proposed approach diverges from the existing method [20] in two fundamental aspects. Firstly, we introduce a RobFA mechanism and seamlessly integrate it into the front-end module, enabling the acquisition of discriminative and meaningful global-local patterns. Secondly, we leverage multi-dilated convolution layers with the dilation factor arranged in ascending order, contrasting with the alternating dilation factor directions utilized in [20]. In the RobFA-Net framework, the front-end module serves as the first-phase feature extractor, responsible for capturing global-local

representations from the input features. Subsequently, it guides the dilated back-end module in extracting robust global-local patterns, as presented in Fig. 2. To design the front-end module, we introduce an RobFA block and incorporate four dense blocks and three transition blocks obtained from the DenseNet121 [28], serving as the backbone network in this study. These dense blocks, denoted as μ , and transition blocks, denoted as ρ , effectively capture patterns denoted as f from the original input features, $X_i \in x$, as illustrated in Eq. (1).

$$f = \mu(\rho(x_i)) \quad (1)$$

The RobFA block utilizes these patterns, denoted as f , as intermediate features and focuses on learning salient meaningful global-local information. It contributes to the performance enhancement of our proposed learning model, as demonstrated in Fig. 3. To design the dilated back-end module, we incorporate five convolution blocks consisting of multi-dilated convolution layers, batch normalization, and activation layers, all densely connected. This configuration enables the extraction of robust global-local characteristics from the discriminative and meaningful global-local patterns, as illustrated in Fig. 4. In this study, we introduce an RobFA block inspired by the Convolutional Block Attention Module (CBAM) [29] and the Channel Spatial Attention Module (CSAM) [30]. While CBAM focuses on learning meaningful spatial and channel information, CSAM aims to capture global and local information from input data. However, our proposed RobFA approach deviates from these existing methods in two key aspects. Firstly, we introduce a Global Channel Attention Module (GCAM) similar to [30], which incorporates global minimum, μ_{gmn} , global maximum, μ_{gmx} , and global average, μ_{gag} pooling layers applied to the feature map, f . Unlike using only μ_{gmx} and μ_{gag} layers, GCAM leverages all three pooling layers to learn diverse global information such as minimum, maximum, and average. Additionally, we employ two fully connected layers, δ_{sh} , for each global information to independently obtain channel-wise weights and fuse them, facilitating the learning of meaningful enhanced global information, f' , as demonstrated in Eq. (2).

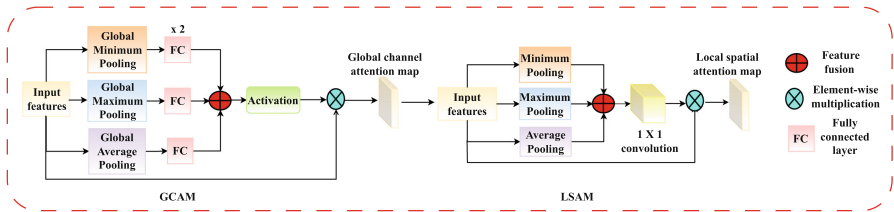


Fig. 3. Detailed diagram of the proposed RobFA block composed of Global Spatial Attention Module (GSAM) and a Local Spatial Attention Module (LSAM).

Secondly, we devise a Local Spatial Attention Module (LSAM) similar to [30]. Specifically, the LSAM utilizes minimum, μ_{mn} , maximum, μ_{mx} , and

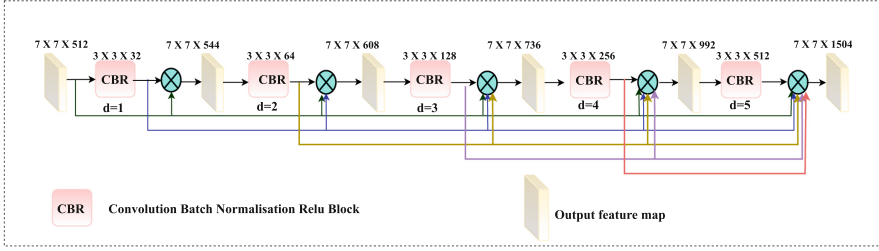


Fig. 4. Proposed Dilated Back-end Module. The dimensions of each feature map are indicated above the respective maps. The filter dimensions in the CBR are also specified, along with the dilation rate (d) in the CBR.

average, μ_{ag} , pooling layers to learn various forms of local information from the previously generated global information, f' . These diverse local information are then fused and refined using a 3×3 convolution layer, Ψ_{33} , to enhance local details. Subsequently, fusing these global and local attention outputs leads to the acquisition of robust and meaningful global-local information, f'' , as illustrated in Eq. (3).

The intuition behind incorporating a global minimum pooling layer in GCAM attention and a minimum pooling layer in LSAM attention is to reduce the issue of information loss. It is achieved by learning salient, meaningful global-local information, ultimately enhancing the learning model performance. We denote the global channel attention map as $m_c \in \mathbb{R}^{1 \times 1 \times C}$, the local spatial attention map as $m_s \in \mathbb{R}^{H \times W \times 1}$, and the intermediate representations as $f \in \mathbb{R}^{H \times W \times C}$. Consequently, the output of robust learned information based on these attention maps, f' and f'' , is derived from the proposed network utilizing m_c and m_s , as depicted in Eqs. (2–5). This learned information is then fused with intermediate representations, f , to generate discriminative learned information, f_d .

$$f' = m_c(f) \times f \tag{2}$$

$$f'' = m_s(f') \times f' \tag{3}$$

where \times denotes element-wise multiplication.

$$m_c = \theta_+(\delta_{sh}(\mu_{gmn}(f)), \delta_{sh}(\mu_{gmx}(f)), \delta_{sh}(\mu_{gag}(f))) \tag{4}$$

$$m_s = \sigma\left(\Psi_{3 \times 3}\left(\theta_+(\mu_{mx}(f'), \mu_{mn}(f'), \mu_{ag}(f'))\right)\right) \tag{5}$$

where θ_+ represents the feature fusion layer (addition), σ denotes the sigmoid function.

These discriminative, meaningful representations, denoted as f_d , serve as inputs to a point-wise convolution layer to generate feature maps, f_p . These feature maps are then used as input to the dilated back-end module. This module incorporates multi-dilated convolution layers and densely connected context

modules to enhance representation power and extract robust global-local patterns, denoted as f_r , as depicted in Eq. (6) and Fig. 4. To construct the multi-dilated convolution layers, we employ five 3×3 convolutional layers with dilation factors, denoted as $f_{(t,d)}$, where t and d signify the respective layer and dilation factors. Specifically, RobFA-Net initializes the dilation rate at one and increases it to five, contrasting with the increasing and decreasing dilation strategy employed in [20]. This approach facilitates the more effective capture of larger objects with increasing dilation rates. In RobFA-Net, there is no necessity to utilize a lower dilation rate to capture small objects, as all dense blocks in the DenseNet121 network, including the fourth ones in the front-end module, capture small objects such as markers, ticks, and symbols.

$$f_r = \theta_{\times}(f_p, f_{1,1}, f_{2,2}, f_{3,3}, f_{4,4}, f_{5,5}) \quad (6)$$

where individual output is given as $f_{1,1}$, $f_{2,2}$, $f_{3,3}$, $f_{4,4}$, and $f_{5,5}$, respectively.

In this study, we employ focal loss [31] to mitigate overfitting issues and enhance the generalization capabilities across the class of the RobFA-Net, which differ from prior existing approaches. The mathematical equation of focal loss is as follows.

$$L = - \sum_i y_i (1 - \hat{y}_i)^{\beta} \log(\hat{y}_i), \quad \beta \geq 0 \quad (7)$$

where y_i represents the ground truth label, (\hat{y}_i) denotes predicted probability, β signifies the focusing parameter, and $(1 - \hat{y}_i)^{\beta}$ specifies the modulating factor.

3.2 Pie-chart Component Extraction (PCE)

Following the initial stage of chart classification (line 1), the subsequent phase involves pie-chart Component Extraction (PCE). This step includes extracting all components within the chart, as depicted in Fig. 1. The entire process of data extraction in pie-chart is presented in Algorithm 1.

Component Object Detection (COD): The task of a COD model is to locate specific regions of interest (BB_{coord}^{label}) (line 2) in the classified image, where BB represents the bounding box, $label$ includes elements such as $\{title (t), legend preview(lp), legend label (ll), and Pie (p)\}$. The coordinates for each BB are depicted as $coord = \{b_x, b_y, b_w, b_h\}$, where b_x and b_y are the coordinates of the center of the detected box, and b_w and b_h represent the width and height of the box, respectively. The superior performance of object detection methods make them suitable for various computer vision-related tasks [32, 33]. Consequently, extensive research has focused on developing YOLO-based object detection models, which have demonstrated impressive performance [11, 36, 37]. Among the different versions of YOLO, the latest version, YOLOv9 [11], stands out for its enhanced detection accuracy. Unlike many neural networks that experience information loss due to repeated layers of feature extraction and spatial transformation, its advantage over other object detection methods lies in its ability to mitigate information loss.

This model consists of three main components: Backbone, Neck, and Head, responsible for feature extraction, feature fusion, and target detection, respectively. YOLOv9 introduces two novel techniques, Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN), specifically designed to address the information bottleneck issue, thereby enhancing both the accuracy and efficiency of object detection. Hence, we utilize the YOLOv9 algorithm for our component object detection task. PGI employs a main branch for standard processing, an auxiliary reversible branch to maintain information integrity with deeper networks, and multi-level auxiliary information to enhance learning capacity. GELAN merges the features of two existing neural network designs, CSPNet [38], and ELAN [39], thus enhancing interlayer information interaction, reducing losses, and computational complexity.

Furthermore, the loss function in YOLOv9 comprises three components: the confidence loss (Δ_{OBJ}), the classification loss (Δ_{cl}), and the positional loss of the target box and the prediction (Δ_{CIoU}) [40]. The final loss function is denoted as (Δ_{loss}):

$$\Delta_{loss} = \Delta_{CIoU} + \Delta_{cl} + \Delta_{OBJ}. \quad (8)$$

Algorithm 1: PieExtract: An end-to-end data extraction algorithm for pie-charts

```

Input :  $Img_{Chart}$ 
Output: " $||$ " = { $Seq_{it}$ ,  $Data\_Value$  (%) }
1  $Img_{pie} \leftarrow RobFA-Net (Img_{chart})$ 
2  $BB_{coord}^{label} \leftarrow COD(Img_{pie})$ 
3  $Seq_{text} \leftarrow TR(BB_{coord}^{text})$ 
4 for each  $B$  do
5    $BB_{coord}^{lp} \leftrightarrow BB_{coord}^{ll}$  // Map each  $lm$  to its  $ll$ 
6    $RGB_{color}^{lp} \leftarrow RGB(BB_{coord}^{lp})$  // Obtain RGB values from each  $ll$ 
7    $\Delta_{color} \leftarrow Match\_Color (BB_{coord}^{lp}, p)$  // Match RGB values to obtain Sectors
8 end for
9 for each  $\Delta_{color}$  do
10   $\Delta_{color} \leftarrow EDS(\Delta_{color})$  // increase Resolution.
11   $\Delta_{grey} \leftarrow GaussianBlur(\Delta_{color})$ 
12   $\Delta_{\{cp_1, cp_2, cp_3\}} \leftarrow Tsi - Tomas(S_{grey})$  // Obtain three sector cornerpoints.
13   $\Delta_{\{cp_c, cp_1, cp_2\}} \leftarrow Distance(\Delta_{\{cp_1, cp_2, cp_3\}})$ 
14   $\langle \leftarrow Angle(\Delta_{\{cp_c, cp_1, cp_2\}})$  // based on lines drawn from corner points.
15   $Data\ Value = \left( \frac{\langle}{360^\circ} \right) \times 100\%$  // Convert  $\langle$  into % value for each sector.
16  Map the obtained values to generate two-tuple consisting of  $Seq_{it}$  and corresponding data value
17 end for

```

Text Recognition (TR): Our TR branch is added after the YOLOv9 model and processes all proposals predicted as text (B_{Coord}^{text}) by the object detection model, where $text = \{t, ll\}$. The text recognition branch employs a Convolutional Recurrent Neural Network (CRNN) [12] based framework. This frame-

work comprises a sequence of CNN layers, which extract and encode features from the detected Pie chart textual objects. These features are then passed through a series of RNN layers to map them into temporal space and subsequently decode them into a sequence of probability distributions representing characters or words at each time step. The CTC loss function is utilized for model optimization during training. Additionally, the CTC decoder is employed for prediction, converting the sequence of probabilities into the final text (Seq_{text}) (line 3).

Sector Data Extraction Method (SDEM): The output of COD is fed to the SEM for extracting angle values from the sectors of each pie detected by the object detection model (B_{coord}^{Pie}). First, we map each legend preview to its respective legend label using the method proposed in [41] (line 5). Second, we extract RGB values from each detected legend preview (B_{coord}^{lp}) (line 6) and then match them to the corresponding color within the pie to extract the corresponding same-color pixel sectors (Δ) (line 7).

To calculate the angle from each sector, we first extract key points for each sector detected from the pie (lines 6–7). This involves initially increasing the resolution using EDS [42] and applying blurring to the patch image to remove false edges (line 10). Then we apply the Tsi-Tomasi corner detection [43] mechanism to detect three key points cp_1, cp_2, cp_3 (line 12). These points are categorized into sector corner points (cp_1, cp_2) and sector center point cp_c . The point that maintains an equal distance from the other two points is classified as the center point, and the angle is computed based on lines drawn from the corner points to this center point. This classification is crucial as it allows us to determine the angle at the center point in relation to the lines connecting the corner points (line 13). Subsequently, the obtained angle is converted into a percentage value for each sector (lines 14–15).

Finally, we generate a two-tuple by mapping legend label and its corresponding sector data value (line 16).

4 Experiments

4.1 Experimental Setup

1. **Datasets:** We evaluated our chart classification model using three datasets: Revision [8], Chagas [9], and FigureQA [10]. In the Revision dataset, there are 2048 images across 10 chart categories like Area and Bar. This dataset is segmented into training, testing, and validation sets, with an 80:20 split for both training and test sets, and the validation set derived from the remaining training data. Similarly, the Chagas dataset is divided into training, validation, and test sets, with 4440 training and 451 test chart images, further split into an 80:20 ratio for training and validation. The Chagas dataset also covers 10 classes such as Venn, Table, and Pareto. The FigureQA dataset serves dual purposes for classification and pie-chart data extraction. Training follows the 80:20 division into training and validation sets. For data extraction, only Pie

chart images from FigureQA were utilized, with 20,000 images for training and 4000 for testing, maintaining the 80:20 split.

2. **Implementation details:** For Chart Classification, the input image was resized to 224×224 . The learning rate was set to 0.0001 with categorical focal loss and used with an Adam optimizer. The model was trained with mini-batches of size 8 for 100 epochs. For the COD process, the input image was resized to 640×640 , and a batch size of 8 was employed, starting with a learning rate of 0.01 and utilizing the SGD optimizer. Training sessions lasted for 50 epochs. As for the TR phase, a batch size of 256 and an initial learning rate of 0.001 were chosen, utilizing the Adam optimizer. The maximum number of epochs was set to 500, with a patience parameter of 50. All models were trained and tested on Nvidia RTX Quadro P4000 GPU.
3. **Evaluation metric:** For chart classification, COD and TR, we employed the standard evaluation metric used by [27]. we employed average accuracy and weighted accuracy to evaluate the chart classification to provide a comprehensive assessment of effectiveness in the classification model. For Pie-chart data extraction, the evaluation metric used is mean error rate & success rate. The error rate for each value of the pie-chart is as follows:

$$\text{error rate} = \frac{|\text{ground truth} - \text{extracted value}|}{\text{ground truth}} \times 100 \quad (9)$$

The success rate is the proportion of pie-charts in which all elements are detected and average error rate is below a particular threshold value. Additionally, the mean error rate also serves as a metric for evaluating the performance of pie-chart data extraction. The mean error rate is calculated as follows:

$$\text{mer} = \frac{1}{m} \sum_{j=1}^m \sum_{k=1}^{|g|} \frac{|g_k - p_k|}{g_k} \quad (10)$$

where g_k is the ground truth value of the k -th element, p_k is a predicted value of the k th element, and m is the number of charts extracted successfully.

4.2 Results

Chart Classification: This section evaluates the performance of the proposed RobFA-Net on Chagas & Revision dataset. Comparison was performed with prior works and state-of-the-art models to evaluate the effectiveness of the proposed architecture. Our proposed model shows better performance in terms of average accuracy on both datasets as shown in Table 1 and Table 3 by effectively capturing both high-level semantic and fine-grained attributes over a wide receptive field due to systematic increase of dilation in back-end module. Significant performance improvements were observed ranging from 5.79% to 1.02% on Chagas dataset and 13.74% to 0.25% on revision dataset. In Chagas dataset, our proposed model surpasses existing methodologies in classifying nearly all types

of charts, with the exceptions of Line and Scatter charts where it demonstrates the second-highest performance. Additionally, it achieves a perfect classification accuracy of 100% on four chart types. In the Revision dataset, our proposed model achieves the best performance across six chart types, including a perfect classification accuracy of 100% on Venn diagrams. For Bar graphs and Radar plots, our model secures the second-highest performance. We also obtained the results on precision, recall and F1 score as shown in Table 2 and Table 4 to obtain the comprehensive assessment of the model’s capabilities and to obtain the nuanced understanding of model behaviour. RobFA-Net out performs previous works across all these evaluation metrics on both datasets. Our proposed model surpasses the second-best method [3] in chagas dataset demonstrating significant enhancements across all evaluated metrics. The precision shows an increase of 0.88%, recall improves by 0.92%, and the F1-score advances by 0.89%. Similarly, the proposed model exhibits improvements across all metrics on the Revision dataset: precision improves by 0.37%, recall by 0.23%, and F1-score by 0.25% against the second best method [20]. Specifically, the RobFA mechanism utilizes global and local attention strategies along with dense connections and obtains global details while also emphasizing important local information in feature maps. FigureQA dataset has been generated synthetically, thus does not have real world chart problems like noise, confusion pair etc. All deep learning models exhibit similar performance on this dataset, having perfect score on all

Table 1. Classwise Accuracy on Chagas dataset. The maximum obtained accuracies are marked in bold.

Category	Choi et al. [7]	Morris et al. [24]	Singh et al. [3]	MDCADNet [20]	J. Thiyam et al. [22]	Touvron et al. [15]	RobFA-Net (Proposed)
Area	97.95	85.71	97.95	97.95	97.95	95.96	100.00
Bar	100	100.00	98.00	98.00	98.00	98.00	100.00
Line	90.19	80.39	92.15	92.15	92.15	82.35	88.23
Map	97.77	91.11	93.33	95.55	97.77	97.77	97.77
Pareto	91.83	85.71	93.87	89.79	91.84	91.84	96.00
Pie	95.65	97.82	100.00	100.00	100.00	97.82	100.00
Radar	94.87	94.87	92.30	89.74	94.87	89.74	97.43
Scatter	93.48	91.30	97.82	97.82	95.65	95.65	96.00
Table	93.10	93.10	96.55	96.55	93.10	93.10	97.00
Venn	100.00	93.61	100.00	100	100.00	100	100
Average	95.56	91.36	96.20	95.75	96.13	94.42	97.15
Weighted Average	95.56	91.13	96.23	95.78	96.23	94.46	97.11

Table 2. Performance metric table on Chagas dataset. The maximum obtained values are marked in bold.

Evaluation Metric	Choi et al. [7]	Morris et al. [24]	Singh et al. [3]	MDCADNet [20]	J. Thiyam et al. [22]	Touvron et al. [15]	RobFA-Net (Proposed)
Precision	0.9563	0.9192	0.9633	0.9586	0.9626	0.9454	0.9718
Recall	0.9557	0.9113	0.9623	0.9578	0.9623	0.9446	0.9712
F1-score	0.9555	0.9119	0.9623	0.9577	0.9622	0.9441	0.9709

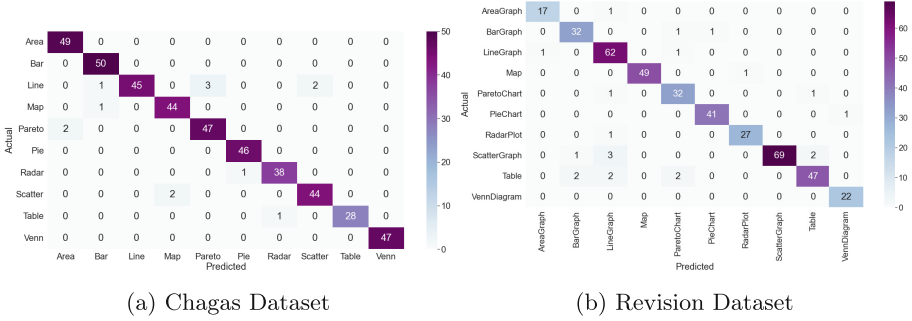


Fig. 5. Confusion matrix of RobFA-Net on a) Chagas Dataset & b) Revision Dataset.

Table 3. Classwise Accuracy on Revision dataset. The maximum obtained accuracies are marked in bold.

Category	Choi et al. [7]	Morris et al. [24]	Singh et al. [3]	MDCADNet [20]	J. Thiyam et al. [22]	Touvron et al. [15]	RobFA-Net (Proposed)
Area Graph	94.44	88.88	94.44	94.44	94.44	94.44	94.44
Bar Graph	91.17	73.52	97.05	94.11	88.23	76.47	94.11
Line Graph	92.18	93.75	89.06	93.75	93.75	82.81	96.88
Map	100	91.11	94.00	82.00	94.00	96.00	98.00
Pareto Chart	91.17	35.29	94.11	94.11	67.64	88.23	94.11
Pie Chart	95.23	95.23	95.23	95.23	97.61	97.62	97.61
Radar Plot	89.28	67.85	100.00	96.42	92.85	85.71	96.42
Scatter Graph	96.00	86.66	93.33	93.33	88.00	92.00	92.00
Table	90.56	88.68	92.45	96.22	94.33	98.11	88.67
Venn Diagram	95.45	90.90	95.45	100.00	95.45	86.36	100
Average	93.55	81.48	94.71	94.97	90.03	89.78	95.22
Weighted Average	93.8	83.57	94.04	94.52	90.10	90.23	94.76

Table 4. Performance metric table on Revision dataset. The maximum obtained values are marked in bold.

Evaluation Metric	Choi et al. [7]	Morris et al. [24]	Singh et al. [3]	MDCADNet [20]	J. Thiyam et al. [22]	Touvron et al. [15]	RobFA-Net (Proposed)
Precision	0.9394	0.84	0.9415	0.9462	0.909	0.9037	0.9497
Recall	0.9381	0.8357	0.9405	0.9454	0.9000	0.9024	0.9476
F1-score	0.9382	0.828	0.9407	0.9454	0.9005	0.9014	0.9478

evaluation metrics. The confusion matrix as shown in Fig. 5 demonstrates our model’s capability to accurately classify challenging chart pairs, such as distinguishing between area and line charts.

COD: We conducted experiments on the latest state-of-the-art object datasets [11, 35, 36], as illustrated in Table 5. We evaluated all these models using a higher intersection over union (IoU) threshold of 0.9. This decision was made because

Table 5. Classwise results of COD at a higher threshold of 0.9 on Figure QA dataset.

Models	Label	Recall	Precision	F1-Score
Yolov9	Pie	1.00	1.00	1.00
	Legend Label	0.962	1.00	0.981
	Legend Preview	0.999	1.00	0.999
	Title	0.992	1.00	0.996
Yolov5	Pie	1.00	1.00	1.00
	Legend Label	0.94	1.00	0.971
	Legend Preview	1.00	1.00	0.999
	Title	0.99	1.00	0.994
Yolov3	Pie	1.00	1.00	1.00
	Legend Label	0.95	1.00	0.976
	Legend Preview	1.00	1.00	0.998
	Title	0.98	1.00	0.991

precise data extraction necessitates data with a higher IoU. Among these models, YOLOv9 outperforms the others in all evaluation metrics. Although the precision remains consistent across all the models, the recall value decreases in the other methods, resulting in a lower F1 score.

Data Extraction: Due to limited availability of datasets containing ground truth for pie chart data, there has been relatively little research in this area. FigureQA stands out as a dataset that provides ground truth for pie charts, making it suitable for evaluating our proposed method. We compare our approach with those of Choi et al. [7] and Paramde et al. [6], both of which rely on pixel count to determine the percentage of each sector in a pie chart. However, these methods struggle with low-resolution pie charts and small sectors, leading to inaccurate classifications. The results presented in Table 6 demonstrate that our proposed corner detection-based method outperforms previous works

Table 6. Success Rate (\uparrow) & mean error rate (mer) (\downarrow) for multiple threshold values of error rates. Maximum success rate & minimum mer is shown in bold.

Threshold value:	<1%	<2%	<3%	<4%	<5%	
Success Rate	0.4585	0.7432	0.8825	0.956	0.986	Choi et al. [7]
	0.06825	0.15025	0.233	0.3045	0.372	Param de [6]
	0.5145	0.7695	0.89625	0.958	0.9865	Proposed Method
mer	0.55	0.89	1.13	1.31	1.40	Choi et al. [7]
	0.52	1.04	1.55	2.00	2.455	Param de [6]
	0.51	0.82	1.05	1.20	1.29	Proposed Method

in pie chart data extraction. We assessed the success rate across various error rate thresholds. The table shows that our method consistently performs better across all threshold values. Even with an error rate of less than 1%, our method successfully extracts data from more than half of the pie charts. Additionally, for the pie charts that were successfully extracted, our method achieves the lowest mean error rate (mer) across all threshold values. This indicates the robustness and accuracy of our data extraction method.

5 Conclusion

We present PiExtract, an end-to-end pipeline for tabular data extraction from pie charts. To accurately classify the charts, we introduce RobFA-Net by designing an RobFA mechanism and a dilated back-end module, which learns robust global and local patterns from chart images, thereby enhancing model performance. Our proposed framework ensures comprehensive object detection and text recognition from charts using YOLOv9 and CRNN approaches. Extensive experimentation conducted on three datasets—Revision [8], Chagas [9], and FigureQA [10]—for chart classification and the pie-chart subset of FigureQA for data extraction showcases the superior performance of our proposed pipeline compared to previous works on these datasets. Our future work includes extending our methodology to other chart types, such as line and bar charts, to broaden the applicability of our approach.

Acknowledgements. This research is supported by the DST under CSRI grant (DST/CSRI/2018/234).

References

1. Davila, K., Setlur, S., Doermann, D., Kota, B.U., Govindaraju, V.: Chart mining: a survey of methods for automated chart analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **43** (2020)
2. Davila, K., Xu, F., Ahmed, S., Mendoza, D.A., Setlur, S., Govindaraju, V.: ICPR 2022: challenge on harvesting raw tables from infographics (chart-infographics). In: 2022 26th International Conference on Pattern Recognition (ICPR). IEEE (2022)
3. Singh, M., Goyal, P.: ChartSight: an automated scheme for assisting visually impaired in understanding scientific charts. In: VISIGRAPP (5: VISAPP) (2021)
4. Singh, M., Kanroo, M.S., Kawoosa, H.S., Goyal, P.: Towards accessible chart visualizations for the non-visually impaired: research, applications and gaps. *Comput. Sci. Rev.* **48** (2023)
5. Lal, J., Mitkari, A., Bhosale, M., Doermann, D.: LineFormer: rethinking line chart data extraction as instance segmentation. *arXiv preprint [arXiv:2305.01837](https://arxiv.org/abs/2305.01837)* (2023)
6. De, P.: Automatic data extraction from 2D and 3D pie chart images. In: 2018 IEEE 8th International Advance Computing Conference (IACC). IEEE (2018)
7. Choi, J., Jung, S., Park, D.G., Choo, J., Elmqvist, N.: Visualizing for the non-visual: enabling the visually impaired to use visualization. In: *Computer Graphics Forum* (2019)

8. Savva, M., Kong, N., Chhajta, A., Fei-Fei, L., Agrawala, M., Heer, J.: Revision: automated classification, analysis and redesign of chart images. In: 24th Annual ACM Symposium on User Interface Software and Technology (2011)
9. Chagas, P., et al.: Architecture proposal for data extraction of chart images using convolutional neural network. In: 2017 21st International Conference Information Visualisation (IV). IEEE (2017)
10. Kahou, S.E., Michalski, V., Atkinson, A., Kádár, Á., Trischler, A., Bengio, Y.: FigureQA: an annotated figure dataset for visual reasoning. arXiv preprint [arXiv:1710.07300](https://arxiv.org/abs/1710.07300) (2017)
11. Wang, C.-Y., Yeh, I.-H., Liao, H.-Y.M.: YOLOv9: learning what you want to learn using programmable gradient information. arXiv preprint [arXiv:2402.13616](https://arxiv.org/abs/2402.13616) (2024)
12. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11) (2016)
13. Bajić, F., Job, J., Nenadić, K.: Chart classification using simplified VGG model. In: 2019 International Conference on Systems, Signals and Image Processing (IWS-SIP). IEEE (2019)
14. Balaji, A., Ramanathan, T., Sonathi, V.: Chart-text: a fully automated chart image descriptor. arXiv preprint [arXiv:1812.10636](https://arxiv.org/abs/1812.10636) (2018)
15. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning (2021)
16. Kavasidis, I., et al.: A saliency-based convolutional neural network for table and chart detection in digitized documents. In: Ricci, E., Rota Bulò, S., Snoek, C., Lanz, O., Messelodi, S., Sebe, N. (eds.) ICIAP 2019, Part II. LNCS, vol. 11752, pp. 292–302. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30645-8_27
17. Poco, J., Heer, J.: Reverse-engineering visualizations: recovering visual encodings from chart images. In: Computer Graphics Forum (2017)
18. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
19. Wu, S., et al.: Improving machine understanding of human intent in charts. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12823, pp. 676–691. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86334-0_44
20. Singh, M., Goyal, P.: MDCADNet: multi dilated & context aggregated dense network for non-textual components classification in digital documents. Expert Syst. Appl. **196**, 116588 (2022)
21. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)
22. Thiyam, J., Singh, S.R., Bora, P.K.: Chart classification: a survey and benchmarking of different state-of-the-art methods. Int. J. Doc. Anal. Recogn. (IJ DAR) **27**(1) (2024)
23. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
24. Morris, D., Müller-Budack, E., Ewerth, R.: SlideImages: a dataset for educational image classification. In: Jose, J.M., et al. (eds.) ECIR 2020. LNCS, vol. 12036, pp. 289–296. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45442-5_36
25. Zhou, X., et al.: East: an efficient and accurate scene text detector. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)

26. Davila, K., Tensmeyer, C., Shekhar, S., Singh, H., Setlur, S., Govindaraju, V.: ICPR 2020 - competition on harvesting raw tables from infographics. In: Del Bimbo, A., et al. (eds.) ICPR 2021. LNCS, vol. 12668, pp. 361–380. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68793-9_27
27. Jung, D., et al.: ChartSense: interactive data extraction from chart images. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 6706–6717 (2017)
28. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
29. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
30. Xia, J., Zhou, Y., Tan, L.: DBGA-net: dual branch global-local attention network for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* (2023)
31. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
32. Sun, L., Cai, Z., Liang, K., Wang, Y., Zeng, W., Yan, X.: An intelligent system for high-density small target pest identification and infestation level determination based on an improved YOLOv5 model. *Expert Syst. Appl.* **239** (2024)
33. Wang, Z., Li, Y., Liu, Y., Meng, F.: Improved object detection via large kernel attention. *Expert Syst. Appl.* **240** (2024)
34. Wang, Y., et al.: Lightweight vehicle detection based on improved YOLOv5s. *Sensors* (2024)
35. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. *arXiv preprint arXiv:1804.02767* (2018)
36. Ultralytics, YOLOv5: Object Detection Software (2023). <https://github.com/ultralytics/yolov5>
37. Wang, C.-Y., Bochkovskiy, A., Liao, M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)
38. Wang, C.-Y., Liao, H.-Y.M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., Yeh, I.-H.: CSP-Net: a new backbone that can enhance learning capability of CNN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2020)
39. Wang, C.-Y., Liao, M., Yeh, I.-H.: Designing network design strategies through gradient path analysis. *arXiv preprint arXiv:2211.04800* (2022)
40. Zheng, Z., Wang, R., Liu, W., Ye, R., Hu, Q., Zuo, W.: Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* (2021)
41. Hadia, K., Suhaib, M., Goyal, P.: LYLAA: a lightweight YOLO based legend and axis analysis method for CHART-infographics. In: Proceedings of the ACM Symposium on Document Engineering (2023)
42. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2017)
43. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision. IEEE (1998)



Machine Learning Solutions for Predicting Bankruptcy in Indian Firms

Chaithra^(✉), Priyanshu Sharma, and Biju R. Mohan

National Institute of Technology, Karnataka, Surathkal, India
{chaithra.217it001,priyanshusharma.222it029,biju}@nitk.edu.in

Abstract. The growing demand to identify potential bankrupt companies has prompted more research into bankruptcy prediction, assisting stakeholders in determining the worthiness of an investment. The Indian stock market offers investment opportunities, but it also involves risk. As a result, it is critical to invest in fundamentally sound companies for long-term investment. To address this need, we created a machine learning-based model for identifying a healthy and distressed firm in the Indian scenario. We created a dataset consisting of 118 bankrupt and 310 healthy firms. The dataset contains three labels: bankrupt, healthy, and financial distress. The addition of the financial distress category improves our ability to recognize and identify firms that are more likely to declare bankruptcy. Recognizing the shortcomings of limited data in the Indian scenario in previous research, our study aimed to include more data instances for training. The dataset included widely recognized financial ratios and macroeconomic data that recognize the interconnectedness of broader economic trends with the company's financial health. Advanced machine learning algorithms, namely Support Vector Machine (SVM), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LGBM), Categorical Boosting (CatBoost), Gradient Boost (GB), and K-Nearest Neighbors (KNN) were applied. The XGBoost and LGBM demonstrated the highest level of classification accuracy and also performed well on real-world data, demonstrating their potential use in supporting investors with decision-making processes.

Keywords: Bankruptcy Prediction · Machine Learning · SMOTE · XGBoost · LGBM

1 Introduction

A company's financial health is a major concern for all stakeholders, as it is a key indicator of potential growth and attractiveness for investment. Financial distress occurs when a company cannot meet its financial obligations. Financial distress does not necessarily lead to bankruptcy. Companies can solve financial issues by restructuring debt, decreasing costs, and obtaining extra funds.

Bankruptcy is often seen as a last resort for organizations experiencing significant financial troubles. According to [23], bankruptcy is a dynamic process and the last stage of financial distress. Bankruptcy is the legal process that may involve financial reorganization, asset liquidation, or other measures as determined by the applicable bankruptcy laws. When the value of debt exceeds the value of assets, bankruptcy is simply a transfer of ownership from equity holders to debt holders, according to [17]. During liquidation, creditors, suppliers, and promoters take precedence over common stockholders, which affects investment returns. Understanding a company’s health before investing is critical for investors to preserve their capital. In this context, using models for predicting bankruptcy or financial health becomes crucial, playing a significant role in helping stakeholders make well-informed decisions. The Indian economy comprises various industries that follow accounting rules and standards. There are existing models in the field of bankruptcy prediction trained on different datasets, but their direct applicability to the Indian context is not possible. Hence, customized methods must be created to provide more accurate forecasts.

In recent years, several researchers have applied machine learning-based and deep learning-based approaches as they have produced promising results. Data is a basic prerequisite for machine learning (ML) and deep learning (DL) based research. The DL and ML models perform better when there is enough training data. While research has been conducted to predict bankruptcy for Indian companies, the datasets used in these studies are not publicly available, unlike datasets for American bankruptcy, Polish bankruptcy, and others. This scarcity resulted in a lack of benchmark datasets and methods tailored explicitly for predicting bankruptcy in the Indian context. Due to this limitation, researchers have had to work with smaller datasets when developing their models.

The main contributions of this paper are

- **Dataset Construction:** Developed a dataset for Indian bankruptcy prediction comprising 310 non-bankrupt and 118 bankrupt companies listed on NSE/BSE from 2010 to 2023. This dataset includes financial data spanning up to 10 years for each company, comprising 62 financial ratios and four macroeconomic variables. The company that is declared bankrupt at year t and the two preceding years ($t-1$), ($t-2$) is labeled bankrupt, while the same company in other years is labeled as financially distressed. A company undergoing restructuring is also categorized as financially distressed. Companies listed in the Nifty 500 are considered healthy instances.
- **Dataset Features:** The dataset includes a broad range of features essential for bankruptcy prediction, covering liquidity, profitability, leverage, and efficiency ratios, along with macroeconomic data for a comprehensive view of firm health. To align with publicly available bankruptcy datasets, such as the Polish Dataset with 64 features and the Taiwanese Dataset with 95 features, features specific to the Indian context were included. Unlike these datasets, which do not typically feature macroeconomic variables, our dataset integrates these important elements.

- **Feature Selection and ML Models:** Applied feature selection methods such as Correlation, KBest, Random Forest, and Recursive Feature Elimination to identify important features and advanced ML algorithms such as Support Vector Machine (SVM), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LGBM), Categorical Boosting (CatBoost), Gradient Boost (GB), K-Nearest Neighbors (KNN) and Artificial Neural Network (ANN) to classify bankrupt, healthy, and financially distressed.

The remainder of the paper is organized as follows: Section 2 reviews the literature on the Indian bankruptcy prediction methods and datasets. Section 3 describes the data collection and labeling mechanism. It also describes the classification methods, important features, and evaluation metrics. Section 4 illustrates the results and analysis. Finally, this paper mentions the limitations and scope for future research.

2 Related Works

Exploring financial health through predicting financial distress or bankruptcy is a significant area within accounting and finance that has garnered attention since the 1990s. The foundational steps in bankruptcy prediction were taken by Beaver (1966), employing univariate analysis. This was followed by Altman (1968), who advanced the methodology by incorporating multivariate discriminant techniques, utilizing financial ratios in the context of the United States. More recently, research has delved into machine learning and deep learning-based models in this domain, necessitating ample data to analyze patterns for identifying companies at risk of bankruptcy.

2.1 Bankruptcy Prediction Methods and Dataset in the Indian Scenario

Bapat et al. [3] analyzed 72 bankrupt and non-bankrupt companies from 1991 to 2013, using 35 financial ratios. The non-bankrupt company was selected from the same industry and matched for asset size. Multiple Discriminant Analysis (MDA), Logistic Regression (LR), and Neural Networks (NN) were used. Singh et al. [21] developed a bankruptcy prediction model for Indian manufacturing companies involving 208 companies. Distressed firms were identified using Board of Industrial and Financial Reconstruction (BIFR) references, while non-distressed companies were matched randomly. The study used 25 financial ratios and MDA, Logit, and Probit Models.

Shrivastav et al. [20] study considered private and public sector banks in India between January 2000 and December 2017. Their study focused on banks with data available for the four years preceding the failure. The sample consisted of 59 banks, 42 surviving banks, and 17 failed banks. Twenty-five financial ratios were used, and the prediction was made using SVM with Linear Kernel (SVMLK)

Table 1. Summary of the methods and dataset used in the Indian Scenario

Reference	Methodology	Sample Ratio	Features	Accuracy
[3]	MDA, LR, NN	72 bankrupt and non-bankrupt	35	MDA: 70.45%, LR: 75%, NN: 77.27%
[21]	MDA, Logit, Probit	104 bankrupt and non-bankrupt	25	MDA: 67.69%, Logit: 48.46%, Probit: 71.54%
[20]	SVMLK, SVMRK	17 failed and 42 survived banks	25	SVMLK: 92.86%, SVMRK: 71.43%
[19]	Binary logistic regression model (M1) with both financial and non-financial, binary logistic model with financial (M2)	82 bankrupts and non-bankrupt companies	12	M1-AUC: 0.8758, M2-AUC: 0.8594
[2]	LR, Lasso Regression, DT, RF, XGBoost, and SVM	262 bankrupt and 262 non-bankrupt	18	LR: 85.8%, Lasso: 82.8%, DT: 89.6%, RF: 92.8%, XGBoost: 90.5%, SVM: 82.9%
[5]	Data Envelopment Analysis (DEA) model and NN	260 listed iron and steel companies	30	DEA: 98.85%, NN: 99.62%
[12]	LR, RF, AdaBoost, ANN	17 failed and 42 survived banks	26	LR: 68.65%, RF: 58.26%, AdaBoost: 98.8%, ANN: 99%

and SVM with Radial Basis Kernel (SVMRK) Function. The non-parametric feature selection method called “Relief Algorithm” was used to select features.

The study by Shetty et al. [19] aimed to predict corporate financial distress in the Indian industrial sector using non-financial indicators such as independent directors on the board and promoters’ ownership stake. The sample data included 82 companies that filed for bankruptcy under the Insolvency and Bankruptcy Code (IBC) and 82 financially sound companies. The data was gathered from Ace Analyzer and analyzed using a Binary Logistic Regression Model (M1) with financial and non-financial data and a Binary Logistic Model (M2) with only financial data.

Arora et al. [2] used a dataset of BSE companies, including 262 bankrupt and 262 financially sound firms, from 2016 to 2019. The size decile was determined by average income and assets over three years. The dataset included 18 independent features from liquidity, profitability, efficiency, and solvency categories. The models used in the study included LR, Lasso Regression (Lasso), Decision Tree (DT), RF, XGBoost, and SVM. In their study on Indian steel businesses, Ghosh

et al. [5] used the CMIE Prowess database to gather 1040 observations from 260 publicly traded companies over four years based on national industry categorization codes. The study used data envelopment analysis and neural networks to analyze the data, and these models' classification performance was superior to that of the Altman Z-score model. Their study comprised 29 financial variables and one non-financial variable, age.

Oberoi et al. [12] used a dataset of 59 Indian banks from 2000–2018 to analyze their survival and failure classes. The dataset included 618 instances, covering 26 financial and non-financial features, and used ML models like LR, RF, AdaBoost, and ANN.

Knaojia et al. [7] used a sample of 68 listed bankrupt companies from May 2016 to the end of 2017–2018. Using a matched-pair sample method, they paired bankrupt and non-bankrupt enterprises. The final sample comprised 68 pairs of listed firms. Data on corporate governance, ownership, financial, and firm-specific variables were collected from annual reports and the CMIE Prowess database for the five years leading up to bankruptcy. Their study used the LR model and the Cox proportional hazard model. Table 1 summarizes the methods and dataset used in the Indian Scenario.

3 Methodology

The Fig. 1 provides an overview of the proposed bankruptcy prediction system. The creation of the dataset is the initial step, followed by data preparation, which comprises feature selection. The dataset is divided into train and test, with 80% of the data being in the train set.

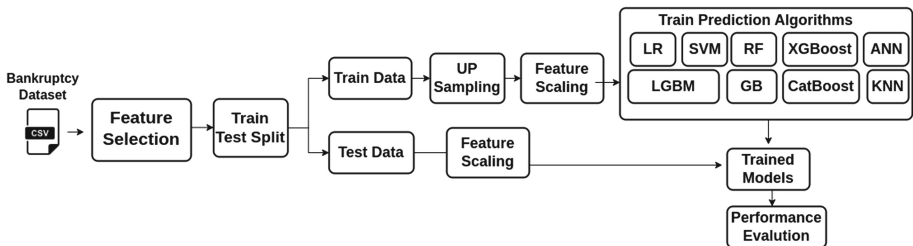


Fig. 1. Methodology for Bankruptcy Prediction.

The dataset is imbalanced, with the majority class being the healthy and the minority classes, i.e., bankrupt and financial distress. The minority classes are upsampled using the Synthetic Minority Oversampling Technique (SMOTE). After training machine learning algorithms on the up-sampled examples, the best-performing model is selected to help investors make judgments.

3.1 Dataset Construction

The Fig. 2 provides an overview of dataset construction. The data collection procedure began with obtaining the names of bankrupt and non-bankrupt companies. We gathered bankrupt company names from the website of the Insolvency and Bankruptcy Board of India (IBBI), which is the governing body for insolvency and bankruptcy procedures. The non-bankrupt companies are taken from the Nifty 500 Index as of September 29, 2023, and this data was taken from the NSE Website. After identifying the list of bankrupt and non-bankrupt companies, the companies' financial statements are scraped. Financial statements include a balance sheet detailing assets, equity, and liabilities, an income statement covering revenues, costs, and profit/loss, and a cash flow statement outlining operational cash inflows and outflows. Multiple sources, including the NSE¹ and BSE², moneycontrol website³, and company annual reports, contributed to the dataset. Macroeconomic data was sourced from World Bank Open Data⁴ and Open Government Data (OGD) Platform India⁵. For each company, we extracted values from financial statements spanning ten years. The code used to scrape the data and to construct the dataset is shared in Github.⁶

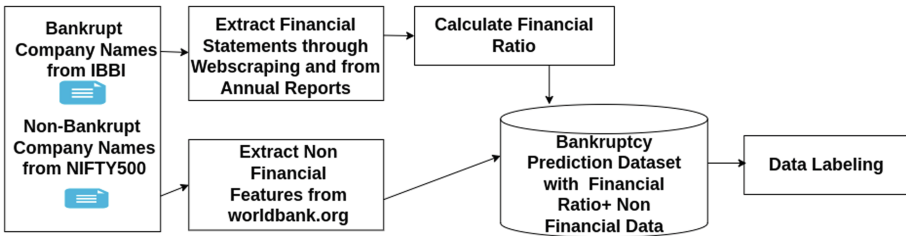


Fig. 2. Dataset Construction Process

Financial Ratios and Macroeconomic Variables: The features of the dataset are the financial ratios and macroeconomic variables. The financial ratios are calculated from the financial statements. The ratios considered in this study are profitability, liquidity, insolvency, efficiency, and activity ratios. The financial ratios widely considered in many research studies are utilized in this study. In addition to standard financial ratios, our dataset includes other commonly cited ratios from the literature. The details of the features are given in Table 2. The

¹ www.nse.com.

² www.bseindia.com.

³ www.moneycontrol.com.

⁴ data.worldbank.org.

⁵ data.gov.in.

⁶ <https://github.com/priyanshu710/Financial-Dataset-using-Web-Scraping>.

Table 2. Financial Formulas and References

Code	Formula	References
X 1	Current Assets/Current Liabilities	[9–11, 13, 16, 18, 22], Polish Dataset, Taiwanese Dataset
X 2	(Current Assets - Inventories)/Current Liabilities	[13, 16, 22], Taiwanese Dataset
X 3	Cash and Cash Equivalents/Current Liabilities	[11, 13]
X 4	Total Liabilities/Total Equity	[11, 13]
X 5	Current Liabilities/Total Liabilities	[13]
X 6	Equity Share Capital/Fixed Assets	[13], Polish Dataset
X 7	Net Sales/Average Total Assets	[13, 16]
X 8	Net Sales/Average Current Assets	[13]
X 9	Gross Profit/Net Sales	[11, 13, 16, 22]
X 10	Operating Profit/Net Sales	[13], Polish Dataset
X 11	Net Profit/Net Sales	[9, 10, 13], Polish Dataset
X 12	Net Profit/Total Assets	[10, 11, 13, 22], Polish Dataset
X 13	Total Debt/Total Assets	[9–11, 13, 22]
X 14	Working Capital/Total Assets	[4, 9, 10, 13, 16, 22], Polish Dataset
X 15	Sales/Total Assets	[4, 9, 10]
X 16	(Total Assets - Total Assets Previous Year)/Total Assets Previous Year	[4]
X 17	Net Profit/Net Sales	[16]
X 18	Cash & Short Term Investment/Total Assets	[9, 10]
X 19	Cash & Short Term Investment/(Equity Share Capital + Total Liability)	[9, 10]
X 20	Cash/Total Assets	[9, 10]
X 21	Cash/Current Liabilities	[9, 10]
X 22	(Inventory - Inventory Previous Year)/Inventory Previous Year	[9]
X 23	Inventory/Sales	[9]
X 24	(Current Liabilities - Cash)/Total Asset	[9, 10]
X 25	Current Liabilities/Sales	[9, 10]
X 26	Total Liabilities/Total Assets	[9, 10], Polish Dataset
X 27	Total Liabilities/(Equity Share Capital + Total Liabilities)	[9, 10]
X 28	Net Income/(Equity Share Capital + Total Liabilities)	[9]
X 29	Operating Income/Total Assets	[9, 10]
X 30	Operating Income/Sales	[9, 10]
X 31	Quick Assets/Current Liabilities	[9, 10]
X 32	Dividends/Net Income	[15]
X 33	EBIT/Overall Capital Employed	[15]
X 34	Net Cash Flow/Revenue	[11]
X 35	Cash Flow from Operations/Total Debt	[11, 13]
X 36	EBT/Current Liabilities	[13]
X 37	EBT/Total Equity	[13]
X 38	Equity/Total Liabilities	[13]
X 39	(Gross Profit + Depreciation)/Sales	Polish Dataset
X 40	Quick Assets/Total Assets	Polish Dataset
X 41	Gross Profit/Total Assets	Polish Dataset

(continued)

Table 2. (*continued*)

Code	Formula	References
X 1	Current Assets/Current Liabilities	[9–11, 13, 16, 18, 22], Polish Dataset, Taiwanese Dataset
X 42	Operating Expenses/Total Liabilities	Polish Dataset
X 43	(Current Assets - Inventory)/Short term Liabilities	Polish Dataset
X 44	Current Assets/Total Liabilities	Polish Dataset
X 45	Short Term Liabilities/Total Assets	Polish Dataset
X 46	(Current Assets - Inventory - Short Term Liabilities)/(Sales - Gross Profit - Depreciation)	Polish Dataset
X 47	(Net Profit + Depreciation)/Total Liabilities	Polish Dataset
X 48	Working Capital/Fixed Assets	Polish Dataset
X 49	(Total Liabilities - Cash)/Sales	Polish Dataset
X 50	Long Term Liability/Equity Capital	Polish Dataset
X 51	Current Assets/Total Assets	Taiwanese Dataset
X 52	Current Liabilities/Assets	Taiwanese Dataset
X 53	Inventory/Working Capital	Taiwanese Dataset
X 54	Inventory/Current Liability	Taiwanese Dataset
X 55	Current Liabilities/Total Liability	Taiwanese Dataset
X 56	Working Capital/Equity Share Capital	Taiwanese Dataset
X 57	Current Liabilities/Equity Share Capital	Taiwanese Dataset
X 58	Long Term Liability/Current Assets	Taiwanese Dataset
X 59	Total Income/Total Expense	Taiwanese Dataset
X 60	Total Expense/Assets	Taiwanese Dataset
X 61	Net Sales/Quick Assets	Taiwanese Dataset
X 62	Sales/Working Capital	Taiwanese Dataset
X 63	Inflation Rate	[6, 8]
X 64	Unemployment Rate	[6, 8]
X 65	Real Interest Rate	–
X 66	GDP	[6, 8]

Polish dataset: <https://archive.ics.uci.edu/dataset/365/polish+companies+bankruptcy+data>

Taiwanese dataset: <https://archive.ics.uci.edu/dataset/572/taiwanese+bankruptcy+prediction>

dataset includes a maximum of 10 years of data for every non-bankrupt company. We attempted to gather a decade’s worth of data for most bankrupt firms, while ten years’ information was unavailable for some. In these situations, the available information was used. Overall, there are 3576 instances in the dataset.

Data Labeling. In our study, firms were labeled bankrupt when a company was declared bankrupt by a court or when a case was admitted for the Corporate Insolvency Resolution Process. The year a company declared bankruptcy is denoted as the benchmark year t . This means that $(t-1)$, $(t-2)$ represent 1, 2 years before the bankruptcy occurred. We labeled bankruptcy year t and two years prior bankruptcy as bankrupt for that particular company [14], and for the remaining years, we labeled the company as financially distressed. While undergoing a restructuring procedure, the company may be able to regain its financial stability, but its creditors may be at risk. As a result, we classified enterprises undergoing restructuring as being in financial crisis. Thus, we labeled companies in the restructuring process as financially distressed. The year of bankruptcy and the year at which the company went for resolution information was taken from IBBI⁷. The matching companies from the same sector and the same size as bankrupt companies from the Nifty 500 index were considered healthy and labeled non-bankrupt in our dataset. The sectors considered for the non-bankrupt and bankrupt companies in our dataset are shown in Figs. 3 and 4.

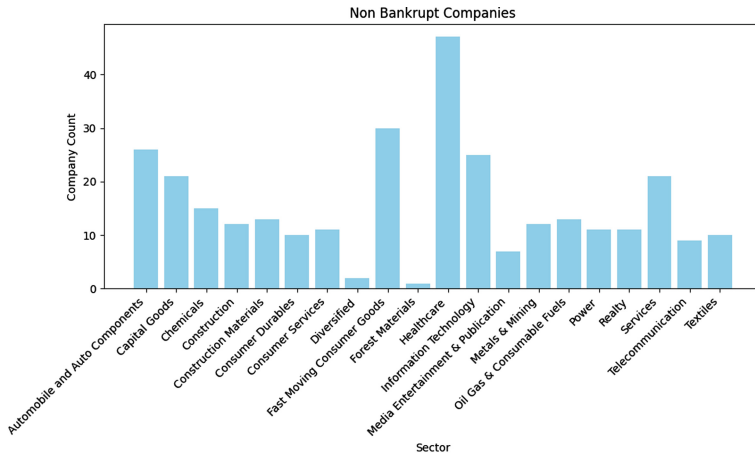


Fig. 3. Sectors considered for selecting Non-Bankrupt Companies

3.2 SMOTE Oversampling Approach

The SMOTE is an oversampling approach designed to balance datasets. The Fig. 5 shows the class distribution of our initial dataset. The SMOTE technique takes a subset of the data from the minority class. Synthetic examples are generated from the feature space. New samples are produced through interpolation between many positive examples that are close to one another. In order to

⁷ <https://ibbi.gov.in/en/claims/cd-summary>.

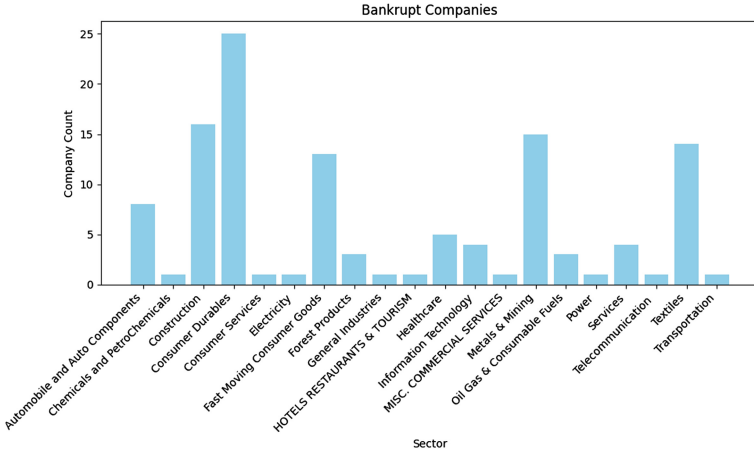


Fig. 4. Sectors considered for selecting Bankrupt Companies

train the classification model, these artificial examples are added to the original dataset. All the minority samples are upsampled to match the count of the majority class.

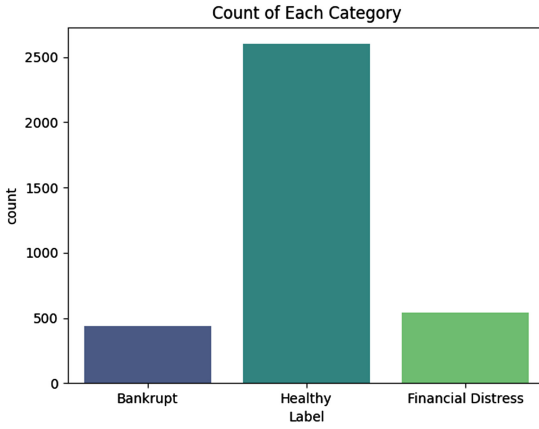


Fig. 5. Class Distribution

3.3 ML Algorithms Implementation Details

The algorithms widely used for bankruptcy prediction tasks are implemented in our dataset. The optimal hyperparameter values for the models are chosen using GridSearchCV, and details are given in Table 3.

Table 3. ML Algorithms and Hyperparameters

Algorithms	Hyperparameters
LR	Regularization (C) = 1
SVM	Kernel = rbf, Gamma = 10, Regularization = 10
RF	No. of Estimators = 500, Maximum Depth = 7
GB	No. of Estimators = 500, Maximum Depth = 7
XGBoost	No. of Estimators = 500, Maximum Depth = 7, Learning Rate = 0.1
LGBM	No. of Estimators = 500, Maximum Depth = 7, Learning Rate = 0.1
CatBoost	Iterations = 500, Depth = 7, Learning Rate = 0.1
KNN	Neighbors = 3, Weights = distance
ANN	No. of Neurons in Hidden Layer 1 = 64, No. of Neurons in Hidden Layer 2 = 32, Activation Function = ReLU, Optimizer = Adam

To evaluate the ML algorithms, the metrics accuracy, precision, recall, and F1-score are used, and their calculations are detailed in Eqs. 1-4.

$$Accuracy = \frac{No. of Correct Predictions}{Total No. of Predictions} \quad (1)$$

$$Precision = \frac{True Positives}{True Positives + False Positives} \quad (2)$$

$$Recall = \frac{True Positives}{True Positives + False Negatives} \quad (3)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

4 Results and Discussion

Many previous studies on bankruptcy prediction focused on binary classification, distinguishing between bankrupt and non-bankrupt companies. In the literature, the labeling process involved assigning the bankrupt label to companies that had been officially declared bankrupt, with the label sometimes being extended to cover all years of available data for that company. However, this approach fails to capture the dynamic nature of a company's financial health, which can shift from a stable position to bankruptcy over time. Our labeling strategy includes designating the year a company is declared bankrupt, as well as the two preceding years, as bankrupt. The remaining years are classified as financially distressed. This approach helps to identify companies that may be heading for bankruptcy in the near future.

4.1 Feature Selection

To identify the important features for bankruptcy prediction, we implemented the following feature selection methods.

- **Correlation:** Measures the statistical correlation between each feature and the target variable, aiming for a subset with low feature-feature correlation and high feature-target correlation to enhance predictive power. Features with a correlation of 0.9 or higher with other features are removed.
- **RF:** Uses an ensemble of DTs to quantify each feature’s contribution to prediction performance.
- **KBest:** Select the top-K features using `mutual_info_classif` scoring function, which measures the dependency between each feature and the target variable.
- **Recursive Feature Elimination (RFE) with LR:** It works by repeatedly building a model and eliminating the least important feature based on the model’s coefficients. This process continues until the desired number of features is reached.

Table 4. Feature Selection Methods and No. of Selected Features

Methods	No of Features
Correlation	44
KBest	55
RFE	55
RF	31

4.2 Feature Analysis

The feature selection method and the number of features selected are given in Table 4. The important features that are identified by taking the intersection of features obtained from all four methods are X4, X6, X7, X9, X10, X18, X28, X31, X35, X36, X37, X38, X41, X50, X56, X57, X59, X63, X65 and X66. These features are categorized into four types: 1) Leverage Ratios, 2) Liquidity Ratios, 3) Profitability Ratios, and 4) Efficiency Ratios.

1. Leverage Ratios

- Total Liabilities to Equity Ratio (X4): Indicates how much debt a company uses to finance its assets relative to shareholder equity.
- Long-Term Liability to Equity Capital (X50): Measures the proportion of long-term debt to equity, indicating potential long-term financial burdens.
- Current Liabilities to Equity Share Capital (X57): Reflects the extent to which current obligations are financed by equity, highlighting short-term financial risks.

2. Liquidity Ratios

- Cash & Short-Term Investment to Total Assets (X18): Shows the proportion of liquid assets, indicating the company's ability to cover short-term liabilities quickly.
- Quick Assets to Current Liabilities (X31): Assesses the ability to meet short-term obligations with the most liquid assets, excluding inventory.
- Cash Flow from Operations to Total Debt (X35): Measures the ability to cover total debt with operating cash flow, reflecting liquidity and debt servicing capacity.

3. Profitability Ratios

- Gross Profit to Net Sales (X9): Indicates how efficiently a company produces goods/services relative to sales.
- Operating Profit to Net Sales (X10): Shows core operational efficiency and profitability.
- Net Income to (Equity Share Capital + Total Liabilities) (X28): Measures overall profitability relative to the total financing (equity and debt).
- Dividends to Net Income (X32): Reflects the portion of income paid out as dividends.

4. Efficiency Ratios

- Net Sales to Average Total Assets (X7): Measures how effectively a company uses its assets to generate sales.
- Gross Profit to Total Assets (X41): Assesses the efficiency in generating gross profit from total assets.

The macroeconomic variables Inflation Rate (X63) or Real Interest Rate (X65) can increase borrowing costs and reduce profitability, while GDP (X66) growth rates can influence overall business conditions. High leverage ratios increase financial risk and the burden of debt repayment, potentially leading to liquidity crises and bankruptcy. Low liquidity ratios indicate a company's struggle to meet short-term obligations, a key bankruptcy risk. Declining profitability ratios can result in insufficient funds to cover expenses and debt, heightening bankruptcy risk. Poor efficiency ratios suggest inadequate returns from assets, negatively impacting overall financial health and increasing the likelihood of financial distress.

4.3 ML Models and Its Classification Results

The ML models were implemented and tested with different feature sets, and the majority of the models performed better when all the features were considered, such as RF, XGBoost, and LGBBoost, GB with an accuracy of 91%. The performance measures for all the implemented models are presented in Table 5. XGBoost and LGBBoost with RFE feature selection and LGBBoost with correlation-based feature selection methods achieved an accuracy of 91%. The classification report and confusion matrix for these methods are detailed in Table 6. Class 0 indicates bankruptcy, class 1 indicates financial distress, and class 2 indicates healthy, i.e., non-bankrupt class. The XGBoost model with all

Table 5. Performance of ML Models for Bankruptcy Prediction (The best results are highlighted in boldface)

Model Name	Feature Selection	Accuracy	Precision	Recall	F1-Score
LR	Correlation (0.9)	83	0.84	0.83	0.83
	Kbest (55)	82	0.84	0.82	0.83
	RFE (55)	83	0.85	0.83	0.83
	Random Forest (Median)	83	0.86	0.84	0.83
	All Features	82	0.84	0.82	0.82
SVM	Correlation (0.9)	85	0.85	0.85	0.85
	Kbest (55)	83	0.83	0.83	0.83
	RFE (55)	83	0.83	0.83	0.83
	Random Forest (Median)	85	0.85	0.86	0.85
	All Features	85	0.85	0.85	0.85
RF	Correlation (0.9)	90	0.9	0.9	0.9
	Kbest (55)	91	0.91	0.91	0.91
	RFE (55)	90	0.91	0.91	91
	Random Forest (Median)	90	0.91	0.90	0.90
	All Features	91	0.91	0.91	0.91
XGBoost	Correlation (0.9)	90	0.9	0.9	0.9
	Kbest (55)	91	0.91	0.91	0.91
	RFE (55)	91	0.91	0.91	0.91
	Random Forest (Median)	89	0.9	0.89	0.9
	All Features	91	0.91	0.91	0.91
LGBM	Correlation (0.9)	91	0.91	0.91	0.91
	Kbest (55)	91	0.91	0.91	0.91
	RFE (55)	91	0.91	0.91	0.91
	Random Forest (Median)	90	0.9	0.9	0.9
	All Features	91	0.91	0.91	0.91
CatBoost	Correlation (0.9)	90	0.9	0.9	0.9
	Kbest (55)	90	0.9	0.9	0.9
	RFE (55)	90	0.9	0.9	0.9
	Random Forest (Median)	89	0.9	0.89	0.89
	All Features	90	0.90	0.90	0.90
Gradient Boost	Correlation (0.9)	90	0.9	0.9	0.9
	Kbest (55)	90	0.9	0.9	0.9
	RFE (55)	91	0.91	0.91	0.91
	Random Forest (Median)	90	0.9	0.9	0.9
	All Features	91	0.91	0.91	0.91
KNN	Correlation (0.9)	85	0.86	0.85	0.85
	Kbest (55)	85	0.86	0.85	0.85
	RFE (55)	86	0.87	0.86	0.86
	Random Forest (Median)	79	0.82	0.79	0.80
	All Features	85	0.86	0.85	0.85
ANN	Correlation (0.9)	85	0.86	0.85	0.85
	Kbest (55)	83	0.85	0.84	0.83
	RFE (55)	84	0.85	0.84	0.84
	Random Forest (Median)	77	0.83	0.77	0.78
	All Features	84	0.85	0.84	0.84

features has a recall score of 0.72 for identifying a bankrupt company, 0.81 for identifying a financially distressed and 0.96 for identifying a healthy company. The feature importance determined by XGBoost and LGBM models using the `plot_importance()` method with gain metric for the top 20 features is shown in Fig. 6.

To assess the effectiveness of our implemented model, we used financial data from a sample of stock companies. We specifically looked at data from Coal India, Infosys, Havells, JSW, Tata Steel, Apollo Hospital and GVK for the year 2023. According to a study conducted by Abdullah et al. [1], JSW Steel Ltd and Tata Steel Ltd are in the Grey Zone of the Z-score, which indicates that they are experiencing financial difficulties. In 2022, GVK filed with IBBI for the corporate insolvency resolution process. We tested the companies' data collected for the years 2022-2023 with the best-performing models on our dataset. Table 7 summarizes the predictions generated by models. When every feature was kept, the predictions produced by XGBoost matched the actual situation.

Table 6. Classification Report and Confusion Matrix

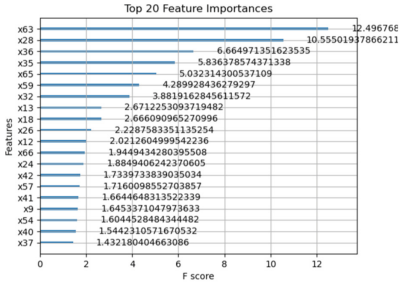
Method	Class	Classification Report				Confusion Matrix		
		P	R	F1-Score	Acc.	0	1	2
XGBoost (RFE)	0	0.76	0.74	0.75	91	74	19	16
	1	0.78	0.78	0.78		18	97	20
	2	0.96	0.96	0.96		39	42	569
LGBM (RFE)	0	0.77	0.72	0.75	91	74	19	16
	1	0.80	0.81	0.81		18	97	20
	2	0.96	0.97	0.96		39	42	569
LGBM (Correlation)	0	0.79	0.72	0.76	91	77	16	16
	1	0.77	0.79	0.78		19	94	22
	2	0.95	0.96	0.96		46	37	567
XGBoost (All Features)	0	0.77	0.72	0.75	91	79	16	14
	1	0.78	0.81	0.80		13	109	13
	2	0.96	0.96	0.96		10	14	626

4.4 Performance Comparison

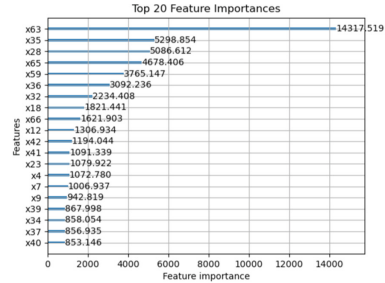
Our study implemented state-of-the-art techniques in the field of bankruptcy prediction, such as SVM, RF, XGBoost, and LGBM, Gradient Boosting, CatBoost, KNN. Our analysis revealed that the XGBoost and LGBM models performed better with our dataset. It is important to note that comparing these results to previous literature may be difficult due to differences in the datasets used. Each dataset has distinct characteristics, making direct comparisons difficult.

Table 7. Model Predictions

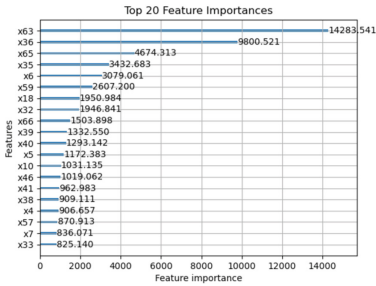
Company	XGBoost (RFE)	LGBM (RFE)	LGBM (Correlation)	XGBoost (All Features)
Coal India	Healthy	Healthy	Bankrupt	Bankrupt
Infosys	Healthy	Healthy	Healthy	Healthy
Havells	Financial Distress	Bankrupt	Bankrupt	Financial Distress
JSW	Financial Distress	Bankrupt	Bankrupt	Financial Distress
Tata Steel	Bankrupt	Bankrupt	Bankrupt	Financial Distress
GVK	Healthy	Healthy	Bankrupt	Bankrupt



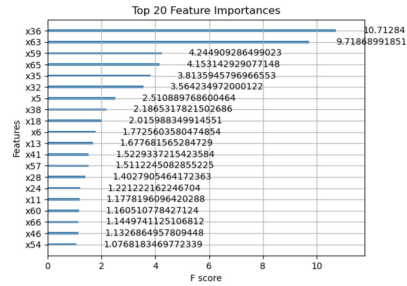
(a) XGBoost with RFE



(b) LGBM with RFE



(c) LGBM with Correlation



(d) XGBoost with all Features

Fig. 6. Feature Importance

5 Conclusion

The Indian stock market offers investment opportunities but also involves risks, especially in identifying companies facing financial distress, which is important for a long-term investment. In this study, the bankruptcy prediction model for the Indian scenario was implemented. The study used a dataset comprising companies from various sectors, with companies labeled as bankrupt, non-bankrupt, and financially distressed. Incorporating financial distress helped to identify the companies likely to become bankrupt soon. The use of financial ratios and macroeconomic variables, widely used in the literature, helped identify patterns

for recognizing bankrupt companies. In this study, widely used ML algorithms such as LR, SVM, RF, XGBoost, LGBM, CatBoost, GradientBoost, and KNN algorithms were implemented on our dataset.

Including an extensive dataset for training helped the machine learning algorithm learn the pattern. XGBoost, LGBM models achieved classification accuracy of 91%. The models were also tested with new instances of real-world data, and the XGBoost model, when all the features were retained, was able to match the actual situation. Future research could include, in addition to financial ratios, the age of the company, the number of employees, and the sentiment hidden in the annual report. Adding these features may help improve the model's classification accuracy.

References

1. Abdullah, M., et al.: Dynamics of speed of leverage adjustment and financial distress in the Indian steel industry. *J. Open Innov. Technol. Market Complex.* **9**(4), 100152 (2023)
2. Arora, P., Saurabh, S.: Predicting distress: a post insolvency and bankruptcy code 2016 analysis. *J. Econ. Finance* **46**(3), 604–622 (2022)
3. Bapat, V., Nagale, A.: Comparison of bankruptcy prediction models: evidence from India. *Acc. Finance Res.* **3**(4), 91–98 (2014)
4. Barboza, F., Kimura, H., Altman, E.: Machine learning models and bankruptcy prediction. *Expert Syst. Appl.* **83**, 405–417 (2017)
5. Ghosh, A., Kapil, S.: Is Altman's model efficient in predicting bankruptcy? - A comparison among the Altman z-score, dea, and ann models. *J. Inf. Optim. Sci.* **43**(6), 1191–1207 (2022)
6. Kanapickienė, R., Kanapickas, T., Nečiūnas, A.: Bankruptcy prediction for micro and small enterprises using financial, non-financial, business sector and macroeconomic variables: the case of the lithuanian construction sector. *Risks* **11**(5), 97 (2023)
7. Kanojia, S., Gupta, S.: Bankruptcy in Indian context: perspectives from corporate governance. *J. Manag. Gov.* **27**(2), 505–545 (2023)
8. Keswani, S., Wadhwa, B.: Withdrawn: association among the selected macroeconomic factors and Indian stock returns (2021)
9. Mai, F., Tian, S., Lee, C., Ma, L.: Deep learning models for bankruptcy prediction using textual disclosures. *Eur. J. Oper. Res.* **274**(2), 743–758 (2019)
10. Mancisidor, R.A., Aas, K.: Using multimodal learning and deep generative models for corporate bankruptcy prediction. arXiv preprint [arXiv:2211.08405](https://arxiv.org/abs/2211.08405) (2022)
11. Montesinos, A.: Profit prediction based on financial statements using deep neural network. In: 2022 IEEE World AI IoT Congress (AIIoT), pp. 533–537. IEEE (2022)
12. Oberoi, S.S., Banerjee, S.: Bankruptcy prediction of Indian banks using advanced analytics. *Econ. Stud. J.* **4**, 22–41 (2023)
13. Özparlak, G., Dilidüzgün, M.Ö.: Corporate bankruptcy prediction using machine learning methods: the case of the USA. *Uluslararası Yönetim İktisat ve İşletme Dergisi* **18**(4), 1007–1031 (2022)
14. Pisula, T.: An ensemble classifier-based scoring model for predicting bankruptcy of polish companies in the podkarpackie voivodeship. *J. Risk Financ. Manag.* **13**(2), 37 (2020)

15. Rasolomanana, O.M.: Bankruptcy prediction model using machine learning. Ph.D. thesis (2022)
16. Saladi, S.D., Yarlagadda, R.: An enhanced bankruptcy prediction model using fuzzy clustering model and random forest algorithm. *Revue d'Intelligence Artificielle* **35**(1) (2021)
17. Senbet, L.W., Wang, T.Y., et al.: Corporate financial distress and bankruptcy: a survey. *Found. Trends® Finance* **5**(4), 243–335 (2012)
18. Shetty, S., Musa, M., Brédart, X.: Bankruptcy prediction using machine learning techniques. *J. Risk Financ. Manag.* **15**(1), 35 (2022)
19. Shetty, S.H., Vincent, T.N.: The role of board independence and ownership structure in improving the efficacy of corporate financial distress prediction model: evidence from India. *J. Risk Financ. Manag.* **14**(7), 333 (2021)
20. Shrivastav, S.K., Ramudu, P.J.: Bankruptcy prediction and stress quantification using support vector machine: evidence from Indian banks. *Risks* **8**(2), 52 (2020)
21. Singh, B.P., Mishra, A.K.: Re-estimation and comparisons of alternative accounting based bankruptcy prediction models for Indian companies. *Financ. Innov.* **2**, 1–28 (2016)
22. Soui, M., Smiti, S., Mkaouer, M.W., Ejbali, R.: Bankruptcy prediction using stacked auto-encoders. *Appl. Artif. Intell.* **34**(1), 80–100 (2020)
23. Volkov, A., Benoit, D.F., Van den Poel, D.: Incorporating sequential information in bankruptcy prediction with predictors based on Markov for discrimination. *Decis. Support Syst.* **98**, 59–68 (2017)



Efficient Object Detection via Fine-Grained Regularization with Global Initialization

Binhan Chen¹, Qiaojun Wu¹, Song Chen¹, and Yi Kang¹

University of Science and Technology of China, Anhui, China
{ab980125,qju}@mail.ustc.edu.cn, {songch,ykang}@ustc.edu.cn

Abstract. The improvement in accuracy of object detection networks has been progressive in facilitating real vision applications. However, the increase in accuracy comes with the cost of increasing memory footprint and computations and it has presented ever increasing challenges for deployment of updated object detection networks. To address these challenges, we propose a structured pruning method called Fine-grained Regularization with Global Initialization (FRGI). Considering the varying impact of filters on the subsequent layers that process the same L1-norm but different means, FRGI introduces a mean-aware sparsity term during the global sparsification to promote near-equal means of the filters, then initializes the global pruned filters by prioritizing them based on the L1-norm. The expressive power on the pruned structures selected in the initialization is transferred with minimum loss in accuracy by applying fine-grained regularization. Moreover, for the residual blocks commonly found in object detection networks, FRGI averages the L1-norm of related filters. We show through extensive validations on the MS-COCO dataset that FRGI leads to more efficient object detection models for all sizes of object detection networks.

Keywords: object detection · fine-grained regularization · global initialization

1 Introduction

The improvement in object detection network accuracy has been instrumental in facilitating the application of vision tasks, such as auto-driving [19], and industrial detection [34]. However, these model performances are heavily reliant on continuous parameters and computation expansion [45], resulting in over-parameterization which has proven to be a common feature of well-performing object detection models [42]. This issue increases the memory footprint, energy consumption, and inference latency of these models [4].

To achieve efficient object detection, more and more efforts have focused on model compression techniques, including knowledge distillation [13,45], neural architecture search [48], and neural network pruning. In this work, we solely focus on structured pruning, which is compatible with other compression methods and generates obvious gains on existing platforms.

Structured pruning directly reduces the memory footprint, energy consumption, and latency of CNN during inference by removing filters or channels [33]. Currently, there are many heuristic methods to achieve structured pruning, such as pruning at initialization [37], redundancy-based [41], and magnitude-based [26]. Among these methods, magnitude-based pruning is still the most popular, which defines a criteria to evaluate the importance of model parameters thereby pruning as many parameters as possible with minimum loss in accuracy. Some works focus on exploring more effective evaluation criteria, such as L1-norm [11,47], γ in BatchNorm [24,49], and activation-related parameters [15,30]. In addition, some works focus on how to better transfer the expressive power of pruning structure, such as driving weights in pruning structures close to 0 by growing regularization [36,38,46]. However, applying these methods to modern object detection models is not easy due to the following downsides: (1)The core of a pruning method is to find criteria to evaluate the importance of model parameters [38]. Most pruning works only focus on the selection of which criteria to evaluate the importance, focusing on solving the “What” problem, but not the “How” problem. For example, how to conduct a fairer evaluation based on one of the criteria and which nodes to conduct the evaluation is the “How” problem. (2) Most pruning methods rely on structure pruning in a coarse-grained way, sometimes even operating on sparse weights at the beginning which often leads to significant performance loss; (3) For the residual blocks in the modern object detection models, there is no targeted method to enhance the pruning performance of these convolution layers.

To deal with the above issue, we propose a method called Fine-grained Regularization with Global Initialization (FRGI). Specifically, we found that for methods that evaluate importance based on the L1-norm, there may be cases in which varying impact of filters on the subsequent layers that process the same L1-norm but different means can cause inaccurate ranking. First, we add a mean-aware sparsity item on the basis of regularization to make the mean of filters nearly equal to resolve the issue of fair ranking. To solve the problem of which nodes to rank and to avoid pruning based on sparse weights, we perform independent ranking. We call the above sparse and ranking processes global initialization. Based on the initialization results, we conduct a regularization process on the original model. In addition to imposing penalties on common filters, we carefully selected kernels that needed to be removed and made their weights approach 0 gradually by L2 regularization, which resulted in less loss in accuracy. We then average the related filters in the residual blocks and use these means to participate in the ranking of the entire model. Main contributions of this work are summarized as follows:

- We propose FRGI to prune object detection networks by the global initialization based on mean-aware sparsity.
- Our proposed FRGI provides a general idea for structured pruning of the object detection models. For models with residual blocks, we unify them to ordinary convolution layers by averaging and minimize accuracy loss through fine-grained growing regularization.
- We conduct experiments using FRGI on both lightweight and deep detection networks, and the results on MS-COCO datasets show that networks generated by FRGI use less memory footprint, require less computations, and keep accuracy unchanged when compared to state-of-the-art references.

2 Related Work

Modern Detectors. Modern object detection networks are mainly divided into two categories, anchor-based and anchor-free [51]. Anchor-free network does not provide a priori knowledge [9, 32], and its accuracy is slightly lower than anchor-based networks generally. Anchor-based networks are divided into one-stage networks and two-stage networks. Two-stage networks need to select candidate regions before detecting [10, 25], which is redundant and slower than one-stage networks [14, 18, 28]. The earliest one-stage network is YOLOv1 [27], which divides an image into grid cells and predicts the bounding box and class probability for each cell. Subsequently, the detector series has continuously absorbed the most advanced detection techniques and achieved SOTA results. To meet the needs of different applications, the latest YOLOv5 [14] and YOLOv6 [18] have designed lightweight models with as few as millions of parameters and deep models with as many as tens of millions of parameters. Therefore, conducting experiments using this series of models is advantageous for directly comparing the effects of pruning.

Neural Network Pruning. Network pruning is a compression technique that solves over-parameterization [33, 38, 46]. Based on the granularity, pruning can be divided into two types: structured pruning and unstructured pruning. Unstructured pruning can result in irregular sparsity [8, 16]. Utilizing this irregular sparsity for acceleration requires special software and hardware support, and the acceleration gain on general-purpose computing platforms is extremely limited [7, 39]. Structured pruning preserves structural regularity [11, 46, 47] by removing filters or other rule structures from the network, which is beneficial for achieving obvious acceleration gains on existing deployment platforms. In this paper, we tackle structured pruning instead of unstructured pruning for effortless acceleration. Most pruning work mainly focuses on more sound pruning criteria to select unimportant weights [38, 46]. Except for the L1-norm, γ in BatchNorm, and activation-related parameters above, the geometric median of filter [12], spectral clustering [50] and similarity [5] are also used as evaluation criteria. Among them, criteria based on weight magnitudes are the most prevailing ones, so we will also use them to solve the “How” pruning. In addition, some pruning works evaluate the importance of weights based on weight magnitudes and focus on

transferring the expressive power of the pruning structure. However, they only selected the pruned filters in a coarse-grained way, while ignoring the kernel [36, 38, 46] in the kept filters. For residual blocks, there are no targeted methods to enhance the pruning performance of convolutional layers in residual blocks.

3 Methodology

In this section, we first describe three key issues in the “How” pruning of object detection networks and propose a global initialization method based on mean-aware sparsity. Then, we show how to minimize the performance loss during the transfer of the pruning structure expressive power by employing fine-grained regularization based on the initialization results. We finalize the method by combining these key steps under one realm of FRGI.

3.1 Global Initialization

Our method is based on regularization to form the desired hardware-friendly sparsity structure, and there are two types of classical approaches exist in previous work using such methods. One is to determine the pruning filters layer by layer based on the target pruning ratio before or after regularization [38, 46], which lacks a global view and yields local optimized results. The other is to prune directly based on the importance ranking results on the basis of a sparse model [11, 24]. Although this approach carries out the importance evaluation from a global perspective, the performance has been irreversibly damaged by the indiscriminate regularization due to its pruning on the basis of the sparse model, even if the fine-tuning after the pruning restores some of the accuracy. In light of these considerations, we propose an independent global initialization process that not only evaluates importance from a global perspective but also isolates it from the subsequent expressive power transfer. Primarily, there are three questions to answer in FRGI regarding the “How” pruning: (1) which pruning criteria to assess the importance of the filters and which regularization to introduce the basic sparsity; (2) how to prune the residual blocks in the object detection network; and (3) how to address the issue that the filters with the same L1-norm but different means may eventually lead to a completely different impact on the output.

(1) Pruning Criterion and Regularization Form. As the most popular pruning method, there are many subdivision criteria in the magnitude-based structured pruning methods. Compared with other criteria, L1-norm is directly related to each weight and possesses advantages in cost and flexibility, so we simply employ L1-norm as the pruning criterion. Although L1 regularization is well-known for inducing sparsity in deep learning, a challenge lies in achieving the desired trade-off between sparsity and accuracy by adjusting the sparsity coefficient[31]. Additionally, the gradient of L1 regularization is not proportional to the weight magnitude, while L2 regularization exhibits this property, making

its sparsification more controllable. Because of this, we opt for L2 regularization to introduce the basic sparsity. Specifically, given the original loss function \mathcal{L} of the object detection network Θ , the total loss function with the L2 regularization term added is formulated as:

$$\varepsilon_l(\Theta; \mathcal{D}) = \mathcal{L}(\Theta; \mathcal{D}) + \frac{1}{2} \sum_{i,l} \delta \|W_i^l\|_2^2 \quad (1)$$

where \mathcal{D} stands for the training dataset, W_i^l represents the i -th filter in the l -th layer of the network, and δ stands for the L2 regularization coefficient corresponding to the weights in that filter. It should be noted that a uniform and constant L2 regularization coefficient is applied to each filter during the global initialization.

(2) Residual Block Pruning. Figure 1 provides a figurative illustration of residual block pruning. It can be observed that due to the presence of the add operator within the residual block, the indexes of pruned filters in the two connected Conv layers must align. For simplicity, the unconstrained convolutional layer within the residual block is referred to as the “free layer,” and the layer constrained by the add operator is referred to as the “related layer.” For the pruning of the related layer, many pruning methods for classification networks directly choose to ignore the related layer [38], while another naive method takes the intersection of the Conv layers in the related layers for pruning after importance ranking based on the L1-norm. The experiment results show that for detection networks with multiple nested residual blocks, the intersection of pruning filters within them is often very small or even empty. Therefore, neither of these two residual block processing methods is feasible. Moreover, for the large amounts of residual blocks present in image super-resolution (SR) networks, some works have proposed to randomly select a set of unimportant filters in the related layer based on the pruning ratio before pruning [46]. Although this method is effective for SR networks, it leads to unacceptable accuracy degradation when applied to object detection networks. Based on these facts, we propose a simple yet effective method for residual block pruning. First, the L1-norm of each filter in the related layer is calculated, and subsequently, the mean of L1-norm is calculated in that group of related layers, forming the formula as:

$$L_{i,1mean} = \frac{1}{n} \sum_{l=1}^n L_{i,1norm}^l \quad (2)$$

where n represents the number of Conv layers in this group of related layers, which is determined by the number of nested residual blocks. $L_{i,1norm}^l$ denotes the L1-norm of the i -th filter in the l -th Conv layer in the related layer, and $L_{i,1mean}$ is the L1-norm mean of the same indexed filters in the related layer. By using this mean value, along with the L1-norms of all the free layers in the detection model, a unified importance ranking is established, and the initialization of global pruned filters is completed.

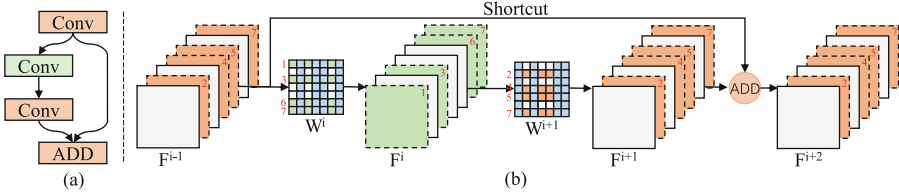


Fig. 1. Illustration of residual block pruning. (a) The common residual block structure in the object detection network, the Conv layers directly connected to the add operator input are all related (in orange), and the other Conv layers are all free (in green). (b) Unrolled representation of a pruned residual block, the input feature map F is represented by a 3D cube, each 2D rectangle represents a channel; the weights W are expanded into a 2D matrix, each row of which represents a filter, and each column represents the kernels of the same index in all filters of that layer. The shaded cells in W represent the pruned structures, where each column of the pruned kernels is determined based on the indexes of the pruned filters of the previous layer. (Color figure online)

(3) Mean-Aware Sparsity. Let’s observe a phenomenon through a simple example. Suppose the input is $x_1 = [1, 2, 3, 1]$ and the two different filters are $w_1 = [-4, -2, -1, 1], w_2 = [3, 2, 1, -1]$. If the filters are chosen based on L1-norm only, $L_{1norm, w_1} = 8$ is chosen instead of $L_{1norm, w_2} = 7$. However, after passing through the ReLU function, the outputs are $y_1 = 0$ and $y_2 = 9$, respectively. Clearly, even though w_1 has a larger L1-norm, its expressive power may be weaker. In summary, this issue often exists in pruning methods that introduce sparsity based on regularization. When the L1-norm of filters is the same but the means differ, it can lead to different outputs and potentially have a completely different impact on the subsequent layers. A short derivation, suppose an input X and a filter W , the output is given by $Y = W * X$, where $*$ denotes the convolution operation. For filters within the same layer, their inputs are the same and consist of non-negative values activated by ReLU. To simplify the process with an assumption that the elements in X so are the same $x_i = c$, c is a non-negative constant and i denotes the element index. Given that the convolution operation is essentially a linear process, we can derive the output as:

$$y_i = w_i * x_i = \sum_{i=1}^n w_i c = n w_{mean} c \quad (3)$$

where $\sum_{i=1}^n w_i c$ represents the multiply-accumulate operation of the weights and the input elements, and w_{mean} denotes the mean of each filter. When the filters L1-norm is the same, the outputs are different in the case of different means unless the $c = 0$ occurs. Consequently, when the elements in the input X are non-negative values randomly distributed, there exists a possibility of completely different outputs.

To address this issue, we introduce a mean-aware sparsity term based on L2 regularization. The updated total loss function is formulated as:

$$\varepsilon_l(\Theta; \mathcal{D}) = \mathcal{L}(\Theta; \mathcal{D}) + \frac{1}{2} \left(\sum_{i,l} \delta \|W_i^l\|_2^2 + 2\vartheta \left| \frac{\sum_j^m w_{i,j}^l}{\mathcal{C}(W_i^l)} \right| \right) \quad (4)$$

where ϑ represents the sparsity coefficient of the mean-aware sparse term and $\sum_j^m w_{i,j}^l$ stands for the summation of all weights in the filter W_i^l . The $\mathcal{C}()$ function is used to count the number of weights in the filter W_i^l . The mean-aware sparse term is introduced in the regularization process, so that the gradient of each weight is also related to the mean of its filter, and finally, the mean of each filter tends to approach zero on the basis of L2 regularization. The effect of introducing the mean-aware sparsity is illustrated in Fig. 2.

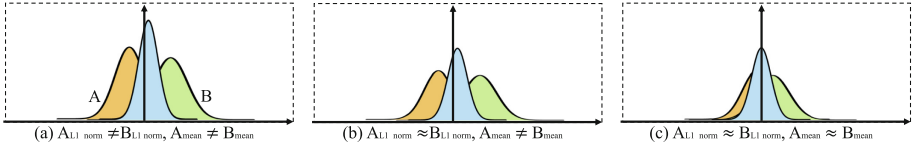


Fig. 2. Illustration of the mean-aware sparsity effect. (a) The distribution of the original filter weights; (b) the distribution of filter weights after L2 regularization, L1-norm reduction; (c) the distribution of the filter weights after L2 regularization with mean-aware sparsity, L1-norm reduced and the means are approximately equal.

3.2 Fine-Grained Regularization

Based on the global initialization, we obtain the pruned filters and the kept filters at the goal pruning ratio. The pruning is directly based on this result in related work, but the models are all lossy under indiscriminate sparsity. In FRGI, we introduce growth regularization for the pruned filters in the original model. Unlike the coarse-grained growth regularization of the pruned filters [36, 38, 46], which is shown in Fig. 1, in the 2D weight matrix, regularization is applied only to the green and orange filters (horizontal), ignoring the blue kernels that also need to be pruned (vertical) as determined by the index of the pruned filters in the previous layer. Such coarse-grained treatment will lead to significant accuracy loss in object detection networks that require precise tuning. To this end, we will apply regularization to both filters and kernels that need to be pruned, driving all the weights of these structures close to zero, completely transferring the expressive power to the kept structures, and minimizing the accuracy loss. The formula for fine-grained regularization is:

$$\varepsilon_l(\Theta; \mathcal{D}) = \mathcal{L}(\Theta; \mathcal{D}) + \frac{1}{2} \sum_{i,j} \delta_i^j \left\| \mathcal{S}_i^j \right\|_2^2 \quad (5)$$

where \mathcal{S}_i^j denotes the i -th filter of the j -th layer, or the i -th kernel of the j -th filter, depending on the structure to be pruned flexibly. δ_i^j denotes the L2 regularization coefficient corresponding to the pruning structure, if the filter or kernel is located in the set of pruning structures determined in the global initialization, then $\delta_i^j > 0$; otherwise, $\delta_i^j = 0$. To safely and completely transfer the expressive power of the pruning structure, we gradually increase the regularization coefficient, formulated as follows:

$$\delta_i^j = \delta_i^j + \Delta \quad (6)$$

The initial value of δ_i^j is set to 0, and Δ represents the update step size, which is set to $\Delta = 10^{-5}$ in our experiments with an update frequency of every 10 iterations. Unlike the δ set in the global initialization on the scale of $10^{-4} \sim 10^{-3}$, the upper limit of δ_i^j is set to 0.02 in the growth regularization to drive the weights in the pruning structure close to 0. The gradual increase of the regularization coefficient starting from 0 is intended to allow the detection model to gradually adapt to the sparsity pattern. Once the upper limit is reached, the regularization continues at this upper limit for a certain period of time.

3.3 Learn Efficient Object Detection Models via FRGI

With all the aforementioned steps, we formulate the final FRGI. The pruning process takes a trained object detection model as input and modifies the loss objective function to the one of interest in the FRGI using Eq. 4 and Eq. 5 at different stages. The indexes of the pruned structures are obtained in the global initialization, and the complete transfer of the expressive power is achieved in the fine-grained regularization. During the pruning, instead of zeroing the weights as in unstructured pruning, these pruning structures are directly removed from the network, resulting in a compact object detection network. In view of the fact that only 50 epochs are required for global initialization and 65 epochs for fine-grained regularization (coefficient growth and stabilization) in FRGI, a fine-tuning of 150 epochs is performed to slightly improve the accuracy after the pruning is completed. Although the entire pruning involves multiple steps, the total number of epochs needed is still less than training the original model once.

4 Experiments

Firstly, we provide an introduction to the experimental settings to ensure result reproducibility. Then, we show through extensive validation on the MS-COCO dataset that FRGI leads to more efficient object detection models, irrespective of whether the architecture is lightweight or deep. Last but not least, through ablation studies and comparison with other pruning methods, we have verified the effectiveness of some improvements in FRGI.

4.1 Experimental Setup

Datasets and Architectures: We choose the MS-COCO dataset [21], which contains 80 object categories of over 200K images and is widely used to benchmark SOTA object detectors due to its rich annotations data and challenging scenarios. FRGI is applied to models of varying parameter scales, including the lightweight YOLOv5-S model and the deep object detection model YOLOv5-L [14]. The former contains only a few million parameters, while the latter contains over 46 million parameters.

Training Settings: All activation functions are set to ReLU, and other training settings are officially consistent. The $\delta \in [10^{-4}, 10^{-3}]$ and the $\vartheta = 0.1\delta$. Other settings have been introduced in Sects. 3.2 and 3.3. All experiments are conducted on Nvidia RTX3090 (for lightweight model) and A100 GPU.

4.2 Comparisons with Lightweight Object Detection Networks

By applying FRGI to the lightweight YOLOv5-S (using different pruning ratios), we are able to obtain multiple lighter models, named YOLO S1, S2, and S3. These models have outperformed various other lightweight networks listed in Table 1 in terms of both parameter quantity and accuracy. Unlike most current approaches that rely on carefully designed architecture for creating lightweight models, the YOLO-S series only requires pruning from the original large model. For instance, compared to the latest YOLOv6-N, YOLO-S1 achieves equivalent accuracy with fewer parameters. Furthermore, while having the same parameter

Table 1. Comparison between YOLOv5-S’s pruning models and lightweight object detection models

Model	Backbone	Input Size	Pram./M	GFLOPs	mAP _{50:95}	mAP ₅₀
YOLOv3-tiny [28]	Tiny Darknet	320	8.85	3.3	14.0	29.0
YOLOv5-S [10]	v5 small	640	7.2	16.5	36.5	55.7
YOLOv4-tiny [1]	Tiny Darknet	320	6.06	4.11	-	40.2
TT-YOLOv5-S [23]	-	640	4.9	18.9	34.2	54.6
SSDLite [29]	MobileNetv1	300	4.31	2.3	22.2	-
YOLOv6-N [18]	EfficientRep	640	4.3	11.1	35.9	51.2
MLNet [20]	Tri-bone	640	2.1	5.9	28.7	46.8
YOLOv5-N [10]	v5 nano	640	1.86	4.5	26.5	-
ABFLMC _{YOLO} [22]	-	640	1.48	4.4	22.8	40.0
YOLOX-Tiny [9]	Darknet53	416	6.2	5.8	32.8	50.3
YOLO-S1	v5 small	640	3.76	13.1	35.9	54.9
YOLO-S2	v5 small	640	1.86	9.3	31.3	50.0
YOLO-S3	v5 small	640	1.11	7.2	26.5	44.2

quantity as YOLOv5-N, YOLO-S2 obtains an 18.1% improvement in accuracy. YOLO-S3 can achieve the same level of accuracy as YOLOv5-N while saving 40.3% of its parameters. Unfortunately, the YOLO-S series models still have shortcomings in computation. So it is necessary to achieve a trade-off among memory footprint, computation, and accuracy.

We also visualize the detection effect of the pruning model, as shown in Fig. 3. The features extracted from different channels are also consistent with expectations, and there are no valid features in the pruning channels.

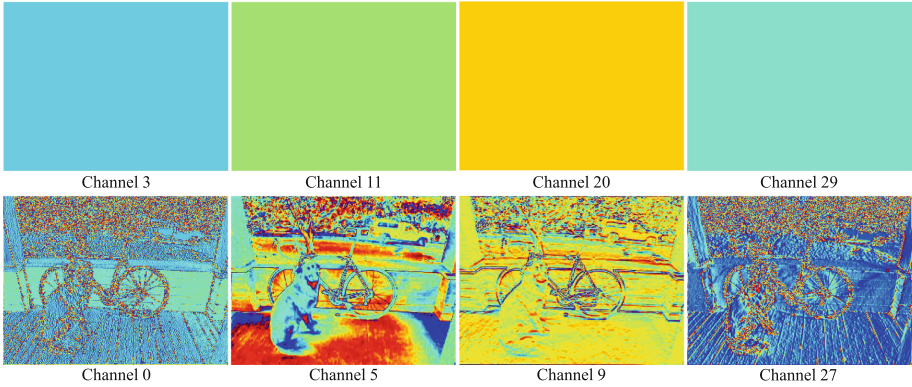


Fig. 3. Visualization of pruning models. The four columns are the output feature maps corresponding to the pruned filters and the kept filters of the first layer in the YOLO-S3 model. Rich features are extracted from the kept channel, while there are no valid features in the pruned channel.

4.3 Comparisons with Deep Object Detection Networks

By applying FRGI to the YOLOv5-L (using different pruning ratios), we are able to obtain slimmer models, named YOLO L1, L2, and L3. In comparison with various deep object detection models listed in Table 2, the YOLO-L series models offer benefits in terms of memory footprint, computation, and accuracy. Specifically, YOLO-L2 obtains the same parameters as YOLOv5-S while achieving 12.8% improvement in accuracy and YOLO-L3 is able to achieve higher accuracy than YOLOv5-S while saving 56.8% of its parameters.

4.4 Comparison with Other Pruning Methods

We further compare the models obtained by using FRGI with those obtained by other structured pruning methods. Considering that each work is pruned based on different original models, it is fairer to compare the decrease in relevant indicators. As shown in Table 3, these comparative results indicate that FRGI has the potential to achieve efficient object detection networks.

Table 2. Comparison between YOLOv5-L’s pruning models and deep object detection models

Model	Backbone	Input Size	Pram./M	GFLOPs	mAP _{50:95}	mAP ₅₀
DETR-DC5-R101 [2]	ResNet101	640	60	253	44.9	64.7
Faster RCNN-R101-FPN [2]	ResNet101	640	60	246	42.0	62.5
YOLOv5-L [14]	v5 large	640	46.5	109.6	46.7	65.4
SSD [44]	ResNet50	640	45.8	42.3	32.3	51.7
YOLOv5-S [14]	v5 small	640	7.2	16.5	36.5	55.7
ResNet50-FPN [31]	ResNet50	640	34	97	37	-
YOLOv6-T [18]	EfficientRep	640	15.0	36.7	40.3	56.6
Gold YOLO-S [35]	-	640	21.5	46.0	45.4	62.5
DFFT-M [3]	-	640	-	67.0	45.7	64.8
YOLO-L1	v5 large	640	16.0	47.8	45.3	64.0
YOLO-L2	v5 large	640	7.16	34.3	41.2	60.1
YOLO-L3	v5 large	640	3.11	26.6	37.6	56.5

Table 3. Comparison between Fine-grained Regularization with Global Initialization(FRGI) and other pruning methods

Model	Backbone	Pram./M	GFLOPs	mAP _{50:95}	mAP ₅₀
YOLOv3 [43]	Darknet53	236	65.86	-	55.2
CAP-YOLO (60%) [43]	Darknet53	86.4	25.32	-	48.7
yolov5m [17]	v5 m	21.4	51.3	43.6	62.7
NS-YOLOv5m [17]	v5 m	15.0	27.9	40.2	58.5
YOLOv5-S [14]	v5 small	7.2	16.5	36.5	55.7
YOLO-S1	v5 small	3.76	13.1	35.9	54.9
Model	Backbone	Pram./M	GFLOPs	mAP _{50:95}	mAP ₅₀
YOLOv5l [6]	v5 large	47	115	48.1	-
Pruned-YOLOv5 [6]	v5 large	3	30	38.2	-
YOLOv5l [40]	v5 large	46.7	115.4	47.0	66.0
YOLOv5l-pruned [40]	v5 large	16.1	49.1	45.5	64.5
YOLOv5-L [14]	v5 large	46.5	109.6	46.7	65.4
YOLO-L1	v5 large	16.0	47.8	45.3	64.0
YOLO-L3	v5 large	3.11	26.6	37.6	56.5

4.5 Ablation Studies

Validation of Residual Block Pruning. We now perform an ablation study to verify the performance of residual block pruning. As shown in Table 4. The models with residual block pruning achieve equivalent or higher accuracy with a

Table 4. Ablation study on residual block pruning

#	Model	Pruning Ratio	Pram./M	GFLOPs	mAP _{50:95}
1	without	55.7%	3.19	11.6	33.4
2	with	66.1%	2.44	10.7	33.3
3	without	73.9%	1.88	8.3	27.7
4	with	74.2%	1.86	9.3	31.3

Table 5. Ablation study on fine-grained regularization

#	Model	Pruning Ratio	Pram./M	GFLOPs	mAP _{50:95}
1	without	47.8%	3.76	13.1	34.6
2	with	47.8%	3.76	13.1	35.9
3	without	83.9%	1.16	7.36	23.8
4	with	83.9%	1.16	7.36	27.0

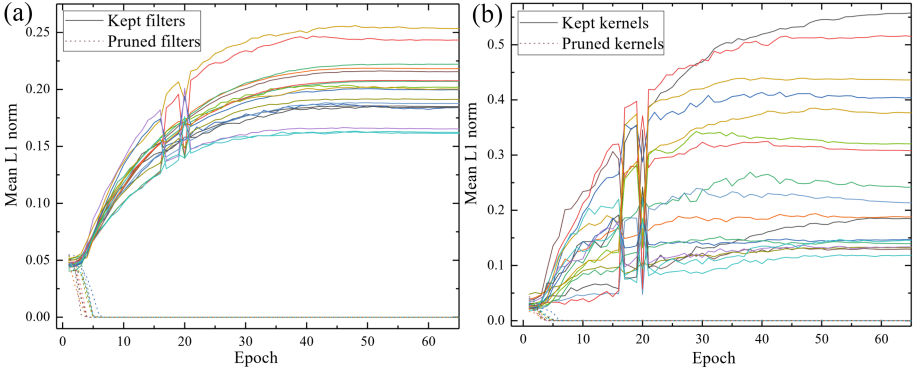


Fig. 4. Visualization the transfer process of expressive power of the #4 model in Table 5. (a) Comparison of L1-norm between pruned filters and kept filters in the first layer of the model; (b) Comparison of L1-norm between pruned kernels and kept kernels in the second layer of the model. As expected, the weight in the pruning structure gradually decreases to nearly 0, while the weights in the kept structure arise spontaneously.

higher pruning ratio. Meanwhile, residual block pruning is beneficial to achieve higher accuracy with a higher pruning ratio.

Validation of Fine-Grained Regularization. Table 5 shows the comparison of pruning experiments with or without fine-grained regularization. Experiments on fine-grained regularization are based on the baseline with the same pruning rate. It can be seen that fine-grained regularization can help to achieve higher accuracy, especially in a higher pruning ratio.

Visualization the Transfer Process of Expressive Power. In Fig. 4, we can visualize the transfer process of expressive power by drawing L1-norm of the pruning structure and the kept structure during fine-grained regularization. It can be seen that as the regularization coefficient continues to increase, the L1-norm of pruning filters and kernels gradually decreases to nearly 0. Interestingly, without applying special treatment to the weights in the kept structure, its L1-norm arises spontaneously. The network’s self-recovery is similar to the compensation effect in the human brain [6].

5 Conclusion

In this paper, we point out the limitations of existing structured pruning in the importance ranking based on L1-norm and coarse-grained ways of transferring the expressive power of pruning structure. In particular, previous works mostly focused on the “What” problem and neglected the “How” problem of pruning. From this perspective, we propose a global initialization method that independently solves the “How” problem through residual block pruning and mean-aware sparsity. Furthermore, we use fine-grained growth regularization to minimize the accuracy degradation caused by the expressive power transfer of pruning structures. We show through extensive validation on the MS-COCO dataset that FRGI leads to more efficient object detection models, irrespective of whether the architecture is lightweight or deep. As the structured pruning ratio cannot be directly converted into the reduction ratio of computation, we will focus on reducing FLOPs to achieve a trade-off among memory footprint, computation, and accuracy in the future.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv preprint [arXiv:2004.10934](https://arxiv.org/abs/2004.10934) (2020)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
3. Chen, P., et al.: Efficient decoder-free object detection with transformers. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13670, pp. 70–86. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20080-9_5
4. Dai, X., et al.: Chamnet: towards efficient network design through platform-aware model adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11398–11407 (2019)
5. Ding, X., Ding, G., Guo, Y., Han, J.: Centripetal SGD for pruning very deep convolutional networks with complicated structure. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4943–4953 (2019)
6. Duffau, H., et al.: Functional recovery after surgical resection of low grade gliomas in eloquent brain: hypothesis of brain compensation. *J. Neurol. Neurosurg. Psychiatr.* **74**(7), 901–907 (2003)



7. Elsen, E., Dukhan, M., Gale, T., Simonyan, K.: Fast sparse convnets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14629–14638 (2020)
8. Frantar, E., Alistarh, D.: SPDY: accurate pruning with speedup guarantees. In: International Conference on Machine Learning, pp. 6726–6743. PMLR (2022)
9. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOx: exceeding YOLO series in 2021. arXiv preprint [arXiv:2107.08430](https://arxiv.org/abs/2107.08430) (2021)
10. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(9), 1904–1916 (2015)
11. He, Y., Kang, G., Dong, X., Fu, Y., Yang, Y.: Soft filter pruning for accelerating deep convolutional neural networks (2018)
12. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4340–4349 (2019)
13. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531) (2015)
14. Jocher, G., et al.: Ultralytics/YOLOv5: v6. 0-yolov5n'nano'models, roboflow integration, tensorflow export, opencv DNN support. Zenodo (2021)
15. Kang, M., Han, B.: Operation-aware soft channel pruning using differentiable masks. In: International Conference on Machine Learning, pp. 5122–5131. PMLR (2020)
16. Lee, N., Ajanthan, T., Torr, P.H.: Snip: single-shot network pruning based on connection sensitivity. arXiv preprint [arXiv:1810.02340](https://arxiv.org/abs/1810.02340) (2018)
17. Li, B., Wu, B., Su, J., Wang, G.: EagleEye: fast sub-net evaluation for efficient neural network pruning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 639–654. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_38
18. Li, C., et al.: YOLOv6: a single-stage object detection framework for industrial applications. arXiv preprint [arXiv:2209.02976](https://arxiv.org/abs/2209.02976) (2022)
19. Li, G., Ji, Z., Qu, X.: Stepwise domain adaptation (SDA) for object detection in autonomous vehicles using an adaptive centernet. *IEEE Trans. Intell. Transp. Syst.* **23**(10), 17729–17743 (2022). <https://doi.org/10.1109/TITS.2022.3164407>
20. Li, Y., Wu, Q., Chen, S., Kang, Y.: Multi-scale lightweight neural network for real-time object detection. In: Khanna, S., Cao, J., Bai, Q., Xu, G. (eds.) PRICAI 2022, Part III. Lecture Notes in Computer Science, vol. 13631, pp. 199–211. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20868-3_15
21. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
22. Liu, M., Luo, S., Han, K., DeMara, R.F., Bai, Y.: Autonomous binarized focal loss enhanced model compression design using tensor train decomposition. *Micromachines* **13**(10), 1738 (2022)
23. Liu, M., Luo, S., Han, K., Yuan, B., DeMara, R.F., Bai, Y.: An efficient real-time object detection framework on resource-constricted hardware devices via software and hardware co-design. In: 2021 IEEE 32nd International Conference on Application-Specific Systems, Architectures and Processors (ASAP), pp. 77–84. IEEE (2021)

24. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2736–2744 (2017)
25. Lu, X., Li, B., Yue, Y., Li, Q., Yan, J.: Grid R-CNN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7363–7372 (2019)
26. Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient inference (2017)
27. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
28. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767) (2018)
29. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
30. Tan, C.M.J., Motani, M.: Dropnet: reducing neural network complexity via iterative pruning. In: International Conference on Machine Learning, pp. 9356–9366. PMLR (2020)
31. Tan, M., Pang, R., Le, Q.V.: Efficientdet: scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781–10790 (2020)
32. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
33. Tsauo, G., Tourtzky, D.S., Lu, T.K., Burgess, N.: Advances in Neural Information Processing Systems?. Morgan Kaufmann Publishers, Burlington (1998)
34. Usamentiaga, R., Lema, D.G., Pedrayes, O.D., Garcia, D.F.: Automated surface defect detection in metals: a comparative review of object detection and semantic segmentation using deep learning. *IEEE Trans. Ind. Appl.* **58**(3), 4203–4213 (2022)
35. Wang, C., et al.: Gold-YOLO: efficient object detector via gather-and-distribute mechanism. In: Advances in Neural Information Processing Systems, vol. 36 (2024)
36. Wang, H., Hu, X., Zhang, Q., Wang, Y., Yu, L., Hu, H.: Structured pruning for efficient convolutional neural networks via incremental regularization. *IEEE J. Sel. Top. Sig. Process.* **14**(4), 775–788 (2019)
37. Wang, H., Qin, C., Zhang, Y., Fu, Y.: Emerging paradigms of neural network pruning. arXiv preprint [arXiv:2103.06460](https://arxiv.org/abs/2103.06460) (2021)
38. Wang, H., Qin, C., Zhang, Y., Fu, Y.: Neural pruning via growing regularization (2021)
39. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
40. Ye, H., Zhang, B., Chen, T., Fan, J., Wang, B.: Performance-aware approximation of global channel pruning for multitask CNNs. *IEEE Trans. Pattern Anal. Mach. Intell.* (2023)
41. Yvinec, E., Dapogny, A., Cord, M., Bailly, K.: Red: looking for redundancies for data-free structured compression of deep neural networks. In: Advances in Neural Information Processing Systems, vol. 34, pp. 20863–20873 (2021)
42. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* **64**(3), 107–115 (2021)

43. Zhang, J., Wang, P., Zhao, Z., Su, F.: Pruned-YOLO: learning efficient object detector using model pruning. In: Farkaš, I., Masulli, P., Otte, S., Wermter, S. (eds.) ICANN 2021. LNCS, vol. 12894, pp. 34–45. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86380-7_4
44. Zhang, J., Zhao, Z., Su, F.: Efficient-receptive field block with group spatial attention mechanism for object detection. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 3248–3255. IEEE (2021)
45. Zhang, M., Wang, L., Campos, D., Huang, W., Guo, C., Yang, B.: Weighted mutual learning with diversity-driven model compression. In: Advances in Neural Information Processing Systems, vol. 35, pp. 11520–11533 (2022)
46. Zhang, Y., Wang, H., Qin, C., Fu, Y.: Learning efficient image super-resolution networks via structure-regularized pruning. In: International Conference on Learning Representations (2021)
47. Zhang, Y., Wang, H., Qin, C., Fu, Y.: Learning efficient image super-resolution networks via structure-regularized pruning. In: International Conference on Learning Representations (2022). <https://openreview.net/forum?id=AjGC97Aofee>
48. Zheng, X., et al.: Neural architecture search with representation mutual information. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11912–11921 (2022)
49. Zhuang, T., Zhang, Z., Huang, Y., Zeng, X., Shuang, K., Li, X.: Neuron-level structured pruning using polarization regularizer. In: Advances in Neural Information Processing Systems, vol. 33, pp. 9865–9877 (2020)
50. Zhuo, H., Qian, X., Fu, Y., Yang, H., Xue, X.: SCSP: spectral clustering filter pruning with soft self-adaption manners. arXiv preprint [arXiv:1806.05320](https://arxiv.org/abs/1806.05320) (2018)
51. Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: a survey. Proc. IEEE (2023)



On Trace of PGD-Like Adversarial Attacks

Mo Zhou^(✉)  and Vishal M. Patel 

Johns Hopkins University, Baltimore, MD 21218, USA
{mzhou32, vpate136}@jhu.edu

Abstract. Adversarial attacks pose security concerns to deep learning applications, but their characteristics are under-explored. Yet largely imperceptible, a strong trace could have been left by PGD-like attacks in an adversarial example. Recall that PGD-like attacks trigger the “local linearity” of a network, which implies different extents of linearity for benign or adversarial examples. Inspired by this, we construct an *Adversarial Response Characteristics* (ARC) feature to reflect the model’s gradient consistency around the input to indicate the extent of linearity. Under certain conditions, it qualitatively shows a gradually varying pattern from benign example to adversarial example, as the latter leads to *Sequel Attack Effect* (SAE). To quantitatively evaluate the effectiveness of ARC, we conduct experiments on CIFAR-10 and ImageNet in a challenging setting. The results suggest that SAE, reflected through the ARC feature, is an effective and unique trace of PGD-like attacks. Our method is designed to generalize with a scarce amount of data, which remains feasible even when access to the full training dataset is impossible. Code: <https://github.com/cdluminate/advtrace>.

Keywords: Adversarial Example Characteristics · Adversarial Response Characteristics · Sequel Attack Effect

1 Introduction

Recent studies reveal the vulnerabilities of deep neural networks [23, 29], where undesired outputs are triggered by an imperceptible perturbation. The attacks pose safety and security concerns for various applications. The PGD-like attacks, including BIM [23], PGD [29], MIM [12], and APGD [9], are strong and widely used in the literature, but under-explored for their characteristics.

Yet, we speculate that a strong attack leaves a strong trace in its result, as in the feature maps [48]. In this paper, we consider an *extremely tough setting* – to identify the trace of PGD-like attacks, given an *already-trained* deep neural network and merely a *tiny set* (*e.g.*, 50) of training data, *without* any change in

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78122-3_6.

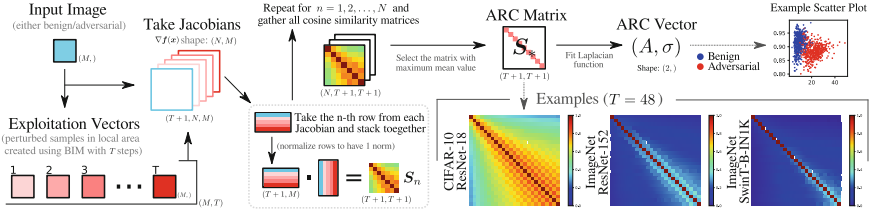


Fig. 1. Diagram for the ARC matrix and the ARC vector. They reflect the model’s gradient consistency within a local linear area around the input to indicate the extent of linearity. Shallow network like ResNet-18 shows higher linearity to benign examples, while deeper networks like ResNet-152 and SwinT-B-IN1K show lower linearity. The ARC matrix consistently shows the trend of increasing gradient consistency along the diagonal line across different networks. Note, we use the same colorbar for all figures afterward in order to keep figures tidy.

architecture or weights, *nor* any auxiliary deep networks. Such a setting requires no dependency on external models or data, and can reveal deeper characteristics of adversarial examples. Expectedly, it is still feasible in more difficult scenarios, even when data is impossible to access (*e.g.*, Federated Learning [30]) and third-party forensics analysis. It also helps us understand adversarial examples.

For instance, in the Federated Learning [30] scenario, a data-demanding attack detector requires access to the training data, which infringes on user privacy regardless of performance. In contrast, data-undemanding detectors can survive in such scenarios, as collecting merely 50 samples from volunteers is still practical. But this type of method is mostly uncharted.

Recall that FGSM [15], the foundation of PGD-like attacks, attributes the vulnerability to “local linearity” being easily triggered by adversarial perturbations. Thus, we conjecture that a network behaves in a greater extent of linearity to adversarial examples than benign (*i.e.*, unperturbed) ones. With the first-order Taylor expansion of a network, “local linearity” implies high gradient proximity in the respective local area. Thus, we can select a series of data points with stable patterns near the input as exploitation vectors using the BIM [23] attack, and then compute the model’s Jacobian matrices with respect to them. Next, the *Adversarial Response Characteristics* (ARC) matrix is constructed from these Jacobian matrices reflecting the gradient direction consistency across exploitation vectors. Unlike benign examples, the results of PGD-like attacks trigger *Sequel Attack Effect* (SAE), leaving higher values in the ARC matrix, reflecting higher gradient consistency around the input, as shown in Fig 1.

The ARC matrix can be simplified into a 2-D ARC vector by fitting a Laplacian function due to their resemblance. This simplifies the interpretation of subsequent procedures. Apart from the qualitative analysis, the ARC vector can be quantitatively evaluated by attack detection and attack type recognition using SVM-based classifiers. The ARC vector can be used for *informed* attack detection (the perturbation magnitude ϵ is known) with an SVM-based binary classifier, or *uninformed* attack detection (the perturbation magnitude ϵ is unknown) with an SVM-based ordinal regression model. The ARC vector can also be used for *attack type recognition* in similar settings with the same set of SVMs.

Experimental results suggest that the SAE reflected through ARC is the unique trace of PGD-like attacks. Meanwhile, through SAE we can also infer attack details, including the loss function and the ground-truth label once the attack is detected.

We evaluate our method on CIFAR-10 [22] with ResNet-18 [17], and ImageNet [10] with ResNet-152 [17] / SwinT-B-IN1K [26]. Visualizations and quantitative experimental results for attack detection and attack-type recognition manifest the effectiveness of our method in identifying SAE. SAE is the unique trace of PGD-like attacks, which also possesses considerable generalization capability among PGD-like attacks even if available training data is very scarce.

Contributions. We present the ARC features to identify and characterize the unique trace, *i.e.*, SAE of PGD-like attacks from adversarial examples. Through the lens of the ARC feature (reflecting the network’s gradient behavior), we also obtain insights into why networks are vulnerable, and why adversarial training works well as a defense. Although our method is specific to PGD-like attacks due to strong assumptions, it is **(1)** intuitive (human-interpretable due to simplicity and not creating a deep model); **(2)** light-weighted (requires no auxiliary deep model); **(3)** non-intrusive (requires no change to the network architecture or weights); **(4)** data-undemanding (generalizes with only a few samples).

2 Related Works

Adversarial Attack and Defense. Neural networks are found vulnerable [15, 43]. Based on this, attacks with different threat models are designed, including white-box attacks, transferability attacks, and black-box attacks [11]. [19] attribute the existence of adversarial examples to non-robust features. To counter the attacks, adversarial training [29, 38, 47] is the most promising defense, but it leads to an expensive training process and suffers from a notable generalization gap. Other types of defenses may suffer from various types of adaptive attacks [4, 44].

Local Linearity is revealed by [15], which leads to a series of defenses and analyses. A “locally linear” model can be used as a theoretical foundation for attacks and defenses [16]. [38] regularize the model to behave linearly in the vicinity of data. [3] show that the network being non-linear locally results in FGSM training failure. [5] show that local linearity arises at initialization. Based on Lipschitz theoretical concepts, [20] presents a layer sustainability analysis framework, and a layer-wise regularized adversarial training method. Our method characterizes PGD-like adversarial examples using local linearity.

Adversarial Example Detection [1, 6] predicts whether a given image is adversarial or not. This can be achieved through adversarial training [50], sub-network [32] or extra loss [35], but it will be costly for ImageNet. Generative methods check reconstruction error [31] or probability density [41], but are data-demanding for accurate distributions. Auxiliary deep models [25, 33] require a large amount of data. Feature statistics methods [21, 24, 27, 28, 39] leverage (high-dimensional) features, but most of them are data-demanding for accuracy.

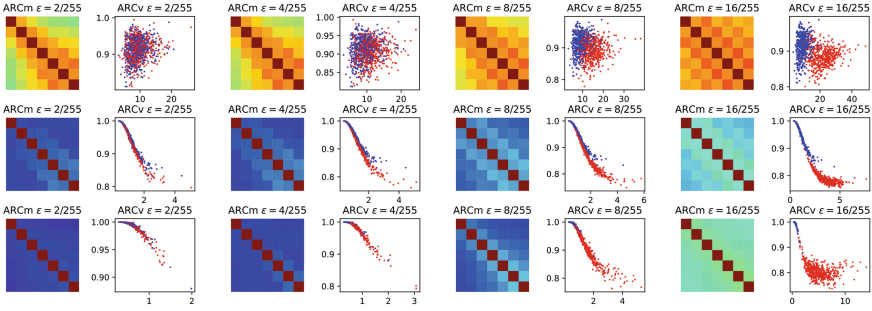


Fig. 2. The ARC features (*i.e.* ARC matrix/vector) of adversarial examples created by the BIM attack. 1st row: ResNet-18 on CIFAR-10; 2nd row: ResNet-152 on ImageNet; 3rd row: SwinT-B-IN1K on ImageNet. Blue/red dots in the scatter plots correspond to the benign/adversarial examples, respectively. The cluster centers of the ARC vector correlate with ϵ . Note, we use the same colorbar as shown in Fig. 1 and omit it here to keep the figure tidy.

Many related works lack ImageNet evaluation and sensitivity analysis with varying attack parameters, even if the difficulty changes accordingly.

3 Adversarial Response Characteristics

A network $f(\cdot)$ maps $\mathbf{x} \in \mathbb{R}^M$ into a pre-softmax output $\mathbf{y} \in \mathbb{R}^N$, where the maximum element after softmax corresponds to the class prediction $\hat{c}(\mathbf{x})$, which should match with the ground truth $c(\mathbf{x})$. A typical attack aims to find an imperceptible adversarial perturbation $\mathbf{r} \in \mathbb{R}^M$ that induces misclassification [23], *i.e.*, $\arg \max_n f_n(\mathbf{x} + \mathbf{r}) \neq c(\mathbf{x})$ where $\|\mathbf{r}\|_p \leq \epsilon$, $\mathbf{x} + \mathbf{r} \in [0, 1]^M$, and $f_n(\cdot)$ is the n -th element of vector function $f(\cdot)$.

According to [15], a neural network is vulnerable as the “locally linear” property is triggered by attack. Thus, we assume that the network $f(\cdot)$ behaves relatively non-linear against benign examples, while relatively linear against adversarial examples. Then, $f(\cdot)$ can be approximated by the first-order Taylor expansion around an either benign or adversarial sample $\tilde{\mathbf{x}}$ (denote $\tilde{\mathbf{x}} \triangleq \mathbf{x} + \mathbf{r}$):

$$f_n(\tilde{\mathbf{x}} + \boldsymbol{\delta}) \approx f_n(\tilde{\mathbf{x}}) + \boldsymbol{\delta}^T \nabla f_n(\tilde{\mathbf{x}}), \forall n \in \{1, 2, \dots, N\}, \tag{1}$$

where $\boldsymbol{\delta}$ is a small vector exploiting the local area around the point $\tilde{\mathbf{x}}$, and gradient vector $\nabla f_n(\cdot)$ is the n -th row in Jacobian $\nabla f(\cdot)$ of size $N \times M$. We name the twice-perturbed $\tilde{\mathbf{x}} + \boldsymbol{\delta}$ as “exploitation vector”. This equation means in order to reflect linear behavior, the first-order gradient $\nabla f_n(\cdot)$ should remain in high consistency (similarity) in the local area regardless of $\boldsymbol{\delta}$. In contrast, when $\tilde{\mathbf{x}}$ is not adversarial ($\mathbf{r} = \mathbf{0}$), neither Taylor approximation nor the gradient consistency is expected to hold. Next, the gradient consistency will be quantized to reveal the difference between benign and adversarial inputs.

Adversarial Response Characteristics (ARC). Using random noise as δ does not lead to a stable pattern of change in a series of exploitation vectors $\{\tilde{\mathbf{x}} + \delta_t\}_{t=0,1,\dots,T}$. Instead, we use the Basic Iterative Method (BIM) [23] to make $\mathbf{f}(\cdot)$ more linear starting from $\tilde{\mathbf{x}}$, which means to “continue” the attack if $\tilde{\mathbf{x}}$ is already adversarial, or “restart” otherwise. However, the ground-truth label for an arbitrary $\tilde{\mathbf{x}}$ is *unknown*. Since PGD-like attacks tend to make the ground-truth least-likely based on our observation, we treat the least-likely prediction $\check{c}(\mathbf{x})$ as the label.¹ Then, for $t = 0, 1, \dots, T$, the BIM iteratively maximizes the cross entropy $L_{\text{CE}}(\tilde{\mathbf{x}} + \delta, \check{c}(\mathbf{x}))$:

$$\delta_{t+1} \leftarrow \text{Clip}_{\Omega} \left(\delta_t + \alpha \text{sign}[\nabla L_{\text{CE}}(\tilde{\mathbf{x}} + \delta_t, \check{c}(\mathbf{x}))] \right), \quad (2)$$

where $\text{Clip}_{\Omega}(\cdot)$ clips the perturbation to the L_p bound centered at $\tilde{\mathbf{x}}$, and $\delta_0 = \mathbf{0}$. If the input $\tilde{\mathbf{x}}$ is benign, then the network behavior is expected to change from “very non-linear” to “somewhat-linear” during the process; if the input $\tilde{\mathbf{x}}$ is already adversarially perturbed, then the process will “continue” the attack, making the model even more “linear” – we call this *Sequel Attack Effect (SAE)*.

To quantize the extent of “linearity”, we measure the model’s gradient consistency across exploitation vectors with cosine similarity. For each $f_n(\cdot)$, we construct a matrix \mathbf{S}_n of shape $(T+1, T+1)$, where for $\forall i, j = 0, 1, \dots, T$:

$$s_n^{(i,j)} = \cos [\nabla f_n(\tilde{\mathbf{x}} + \delta_i), \nabla f_n(\tilde{\mathbf{x}} + \delta_j)]. \quad (3)$$

As the model $\mathbf{f}(\cdot)$ becomes more “linear” to the input (higher gradient consistency), the off-diagonal values in \mathbf{S}_n are expected to gradually increase from the top-left to the bottom-right corner. Note that the attack may not necessarily make all $f_n(\cdot)$ behave linear, so we select the most representative cosine matrix with the highest mean as the *ARC matrix*: $\mathbf{S}_* \triangleq \mathbf{S}_{n^*}$, where $n^* = \arg \max_n \sum_{i,j} s_n^{(i,j)}$.

The example ARC matrixes can be found in Fig. 1. We note the values in the ARC matrix are high along the diagonal line, and drastically decrease when far away from the diagonal. Due to ARC matrix resembling Laplacian function with the matrix diagonal being the center, we simplify it into a 2-dimensional *ARC vector* (A, σ) by fitting $\mathcal{L}(i, j; A, \sigma) = A \exp(-|i - j|/\sigma)$ with Levenberg-Marquardt algorithm [46], where i, j are matrix row and column indexes, while A and σ are function parameters. For brevity, we abbreviate ARC matrix as “ARCm”, and ARC vector as “ARCv”. The overall process for ARCm/ARCv calculation is shown in Fig. 1.

Visualizing SAE. We compute ARCm based on some benign examples using $T=48$, as shown in Fig. 1. The trend of being gradually “linear” (higher cosine similarity) along the diagonal is found across architectures. Thus, SAE is similar to “continuing” an attack from halfway on the diagonal in such a large ARCm.

¹ Please note the subtle difference between $\check{c}(\mathbf{x})$ and $\hat{c}(\mathbf{x})$ – The $\hat{c}(\mathbf{x})$ is the model prediction corresponding to the *largest* logit value, while $\check{c}(\mathbf{x})$ is the “least-likely” prediction corresponding to the *smallest* logit value.

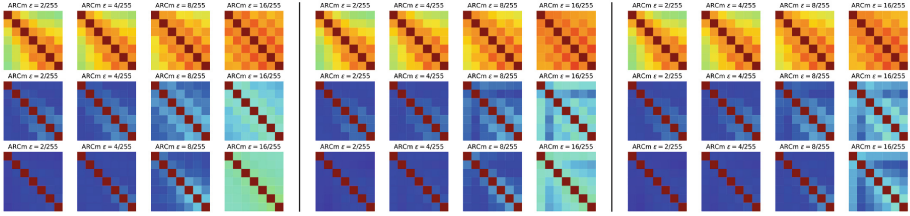


Fig. 3. ARCm with adversarial examples created by PGD (left), MIM (middle), and APGD (right) attacks. The three rows correspond to ResNet-18, ResNet-152, and SwinT-B-IN1K, respectively. These attacks manifest similar SAE in ARCm. Note, we use the same colorbar as shown in Fig. 1 and omit it here.

As illustrated in Fig. 2, already adversarially perturbed input (using BIM) leads to larger cosine similarity at the very first exploitation vectors as perturbation magnitude ϵ increases from 0 to 16/255. Meanwhile, the cluster separation for ARCv is more and more clear. Thus, a clear and gradually changing pattern can be seen in ARCm and ARCv. This pattern is even valid and clear for the state-of-the-art ImageNet models. In brief, SAE is reflected by higher gradient consistency in ARCm, or greater σ and smaller A in ARCv. Similar results for other attacks in Fig. 3 indicate the possibility of generalization among them with only training samples from the BIM attack.

Uniqueness of SAE. Whether SAE can be consistently triggered depends on the following conditions simultaneously being *true*: **(I)** whether the input is adversarially perturbed by an *iterative* projected gradient update method for many steps; **(II)** whether the attack leverages the *first-order gradient* of the model; **(III)** whether the L_p boundary types are the same for the two stages, *i.e.*, attack and exploitation vectors; **(IV)** whether the loss functions for the two stages are the same; **(V)** whether the labels used (if any) for the two stages are relevant. Namely, only when the attack and exploitation vectors “match”, can SAE be uniquely triggered as the exploitation vectors “continue” an attack, or they will “restart” an attack. Thus, in Fig. 1, Fig. 2 and Fig. 3, all the conditions are true as they involve PGD-like attacks. Due to the strong assumptions, the SAE being insensitive to non-PGD-like attacks (*e.g.*, [7]) is a *limitation*. However, the unique SAE meanwhile shows a possibility of inferring the attack details leveraging the above conditions. SAE is the trace of PGD-like attacks. Ablations for these five conditions are presented in Sect. 5.

Adaptive Attack exists against defenses [44] and detection [6]. To avoid SAE, an adaptive attack must reach a point where the corresponding ARCm has a mean value as small as that for benign examples. Intuitively, an adaptive attack has to simultaneously solve $\min_{\mathbf{r}} \|\mathbf{S}_*(\mathbf{x} + \mathbf{r})\|_F$ (Frobenius norm) alongside its objective. It, however, requires the gradient of Jacobians, namely at least $T + 1$ Hessian matrices, *i.e.*, $\nabla^2 f_n(\cdot)$ of size $M \times M$ for gradient descent. This is computationally prohibitive as in the typical ImageNet setting (*i.e.*, $M=3 \times 224 \times 224$), a Hessian in `float32` precision needs 84.4GiB memory. At this point, the cost of such adaptive attack that hides SAE is much higher than computing ARC.

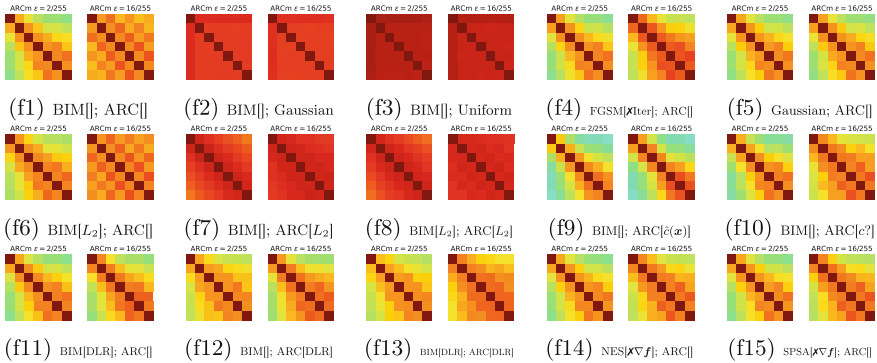


Fig. 4. Ablation of SAE uniqueness by adjusting exploitation vectors for ARC. Each subfigure of ARCM pair has two annotations: (1) attack and its settings, where empty brackets mean default setting unless overridden: [L_p is L_∞ ; Loss is L_{CE} ; \checkmark (is) iterative; \checkmark (can access) gradient $\nabla f(\cdot)$]; (2) exploitation vector settings, e.g. “ARC[]” with the default setting [L_p is L_∞ ; Loss is L_{CE} ; Label is $\checkmark(\cdot)$]. The “c?” means random guess. This figure is supplementary to Table 2. We reuse the identical colorbar as in Fig. 1 and omit the colorbar here in order to save horizontal space and maintain a tidy layout.

Instead, a viable way to avoid SAE is to use non-PGD-like attacks that break the SAE uniqueness conditions. This paper focuses on characterizing the unique trace of existing PGD-like attacks, instead of a general detection or defense.

4 Quantitative Evaluation of ARC Feature

In order to quantitatively support the effectiveness of ARC/SAE, we adopt it for two potential tasks, namely attack detection and attack type recognition. Attack detection aims to identify the attempt to adversarially perturb an image *even if* it fails to change the prediction (but leaves a trace).² Attack type recognition aims to identify whether an adversarial example is created by PGD-like attacks. Our method relies on the uniqueness of SAE to PGD-like attacks.

Informed Attack Detection determines whether an arbitrary input $\tilde{\mathbf{x}}$ is adversarially perturbed, while the perturbation magnitude ε is *known*. It can be viewed as a binary classification problem, where the input is ARCv of $\tilde{\mathbf{x}}$, and the output 1 indicates “adversarially perturbed”, while 0 indicates “unperturbed”. Thus, for a given $\varepsilon = 2^k/255$ where $k \in \{1, 2, 3, 4\}$, a corresponding SVM [36] classifier $h_k(\tilde{\mathbf{x}}) \in \{0, 1\}$ can be trained using some benign ($\varepsilon=0$) samples and their adversarial counterparts ($\varepsilon=2^k/255$). Even if the training data only involves the BIM attack, we expect generalization for other PGD-like attacks from visualization results despite domain shift.

Uninformed Attack Detection determines whether an arbitrary input $\tilde{\mathbf{x}}$ is adversarially perturbed when the perturbation magnitude ε is *unknown*. It can

² It is undesirable to wait until the attack has succeeded.

be viewed as an ordinal regression [34] problem, where the input is ARCv, and the output is the estimation of k , namely $\hat{k} \in \{0, 1, 2, 3, 4\}$. The corresponding estimate of ε is $\hat{\varepsilon} = \mathbf{1}\{\hat{k} > 0\}2^{\hat{k}}/255$, where $\mathbf{1}\{\cdot\}$ is the indicator function. Specifically, this is implemented as a series of binary classifiers (SVM), where the k -th ($k \neq 0$) classifier predicts whether the level of perturbation is greater or equal to k , *i.e.*, whether $\hat{k} \geq k$. Note, based on our visualization, the ARCv cluster of adversarial examples is moving away from that of benign examples as ε (or k) increases. This means the ARCv of an adversarial example with $\hat{k} \geq k$ will also cross the decision boundary of the k -th SVM $h_k(\cdot)$. Namely the SVM $h_k(\cdot)$ can also tell whether $\hat{k} \geq k$, and thus can be reused. Finally, the ordinal regression model is the sum of predictions over the SVMs: $\hat{k} = \sum_{k \in \{1, 2, 3, 4\}} h_k(\tilde{\mathbf{x}})$. A perturbation is detected as long as $\hat{k} > 0$. Estimating k (or ε) for $\tilde{\mathbf{x}}$ is similar to matching its ARCm position inside a larger ARCm calculated starting from a benign example. The estimate does not have to be precise, as the detection is already successful once any SVM correctly raises an alert.

Although a detector in practice knows nothing about a potential attack including the attack type, evaluation of uninformed attack detection with *known* attack type is enough. Regarding the performance of uninformed attack detection given a specific attack type as a conditional performance, the expected performance in the wild can be calculated as the sum of conditional performance weighted by the prior probabilities that the corresponding attack happens.

Inferring Attack Details. Due to the SAE uniqueness in Sect. 2, once attack is detected, we can predict that the attack: (I) performs projected gradient update iteratively; (II) uses the first-order gradient of $\mathbf{f}(\cdot)$; (III) uses the same type of L_p bound as exploitation vectors (L_∞ by default); (IV) uses the same loss as exploitation vectors ($L_{CE}(\cdot, \cdot)$ by default); (V) uses a ground-truth label that is relevant to the least-likely class $\check{c}(\tilde{\mathbf{x}})$ used for exploitation vectors (in many cases $\check{c}(\tilde{\mathbf{x}})$ is exactly the ground-truth). In other words, model prediction can be corrected into the least-likely class $\check{c}(\tilde{\mathbf{x}})$ upon detection. The disadvantage of ARC being insensitive to non-PGD-like attacks is meanwhile the advantage of being able to infer attack details of PGD-like attacks.

Attack Type Recognition determines whether an adversarial input is created by PGD-like attacks in the uninformed setting for forensics purposes. The corresponding binary classifier can be built upon the previously discussed detectors, because SAE only responds to PGD-like attacks.

5 Experiments

In this section, we quantitatively verify the effectiveness of the ARC features in two applications under an *extremely tough setting*. The MNIST evaluation is omitted, as the corresponding conclusions may not hold [6] on CIFAR-10, let alone ImageNet. We evaluate ResNet-18 [17] on CIFAR-10 [22]; ResNet-152 [17] and SwinT-B-IN1K [26] on ImageNet [10] with their official pre-trained weights (this reflects the advantage of our method for being non-intrusive). Our code is publically available at <https://github.com/cdluminate/advtrace>.

ARC Feature Parameter. For the BIM attack for exploitation vectors, we set step number $T = 6$, and step size $\alpha = 2/255$ under the L_∞ bound with $\varepsilon = 8/255$. Note, the mean value of ARCM will tend to 1 with a larger T , making ARCV less separatable. We choose $T = 6$ to clearly visualize the value changes within ARCM, but this does not necessarily lead to the best performance.

Training. We train SVMs $h_k(\cdot)$ with RBF kernel. We randomly select **50** training samples from CIFAR-10, and perturb them using *only* BIM with magnitude $\varepsilon = 2/255, 4/255, 8/255, 16/255$, respectively. Then each of the four $h_k(\cdot)$ is trained with ARCV of the benign ($\varepsilon = 0$) samples and perturbed ($\varepsilon = 2^k/255$) samples. Likewise, for ImageNet we randomly select **50** training samples and train SVM in a similar setting separately for ResNet-152 and SwinT-B-IN1K. The weight for benign examples can be adjusted for training to control the False Positive Rate (FPR).

Testing. For CIFAR-10, all 10000 testing data and their perturbed versions with different ε are used to test our SVM. For ImageNet, we randomly choose 1024 testing samples due to costly Jacobian computation. A wide range of adversarial attacks are involved, including (1) PGD-like attacks: BIM [23], PGD [29], MIM [12], APGD [9], AutoAttack (AA) [9]; (2) Non-PGD-like attacks: (2.1) other white-box attacks: FGSM [15], C&W [7] (we use $\varepsilon \in \{0.5, 1.0, 2.0, 3.0\}$ in L_2 case), FAB [8], FMN [37]; (2.2) transferability attacks: DI-FGSM [49], TI-FGSM [13] (using ResNet-50 as proxy); (2.3) score-based black-box methods: NES [18], SPSA [45], Square [2]. AutoAttack is regarded as PGD-like because APGD is its most significant contributor for success rate.

Metrics. The SVMs are evaluated with Detection Rate (DR, *a.k.a.*, True Positive Rate) and False Positive Rate (FPR). For inferring the ground-truth label, we report the original accuracy for perturbed examples (denoted as “Acc”) and that after correction (denoted as “Acc*”). Mean Average Error (MAE) is also reported for ordinal regression. Accuracy is reported for attack type recognition.

5.1 ARC for Attack Detection

For each network, the corresponding SVMs are trained and evaluated as shown in Table 1. Columns with a concrete ε value are informed attack detection, while the “ $\varepsilon=?$ ” column is uninformed attack detection. As can be expected from visualization results, the ARCV clusters are gradually becoming separatable with ε increasing, hence the increase of DR. Notably, the large perturbations (*i.e.*, $\varepsilon = 16/255$) are hard to defend [38], but can be consistently detected across architectures. The ARC feature is especially effective for Swin-Transformer, because this model transitions faster from being non-linear to being linear than other architectures. Such characteristics are beneficial for SAE.

Upon detection of an attack, our method can correct the prediction into the least-likely class as a post-processing step. Its success rate depends on whether the attack is efficient to make the ground-truth class least-likely, and whether the network is easy for the attack to make a class least-likely. From Table 1, both ResNet-18 and SwinTransformer have such a property and lead to high

Table 1. Informed and Uninformed (the “ $\epsilon=?$ ” column) Attack Detection. All numbers are percentages with the “%” sign omitted, except for MAE. Numbers greater than 50% are in bold font.

Dataset Model	Attack	$\epsilon = 2/255$				$\epsilon = 4/255$				$\epsilon = 8/255$				$\epsilon = 16/255$				$\epsilon = ?$				
		DR	FPR	Acc	Acc*	DR	FPR	Acc	Acc*	DR	FPR	Acc	Acc*	DR	FPR	Acc	Acc*	MAE	DR	FPR	Acc	Acc*
CIFAR-10 ResNet-18	BIM	0.0	0.0	33.5	33.5	0.0	0.0	6.4	6.4	32.3	1.5	0.4	17.8	79.2	1.1	0.0	62.4	1.55	30.9	1.5	10.1	30.7
	PGD	0.0	0.0	33.7	33.7	0.0	0.0	6.4	6.4	33.0	1.5	0.4	18.6	81.2	1.1	0.0	64.8	1.54	31.5	1.5	10.1	31.5
	MIM	0.0	0.0	30.4	30.4	0.0	0.0	6.5	6.5	37.5	1.5	0.4	22.3	84.5	1.1	0.0	67.4	1.50	33.6	1.5	9.3	32.4
	APGD	0.0	0.0	29.3	29.3	0.0	0.0	5.1	5.1	36.9	1.5	0.2	20.7	78.8	1.1	0.0	55.8	1.53	31.5	1.5	8.7	28.0
	AA	0.0	0.0	27.4	27.4	0.0	0.0	2.1	2.1	37.3	1.5	0.0	20.6	78.4	1.1	0.0	55.6	1.53	31.6	1.5	7.4	26.8
	?	0.0	0.0	30.9	30.9	0.0	0.0	5.3	5.3	35.4	1.5	0.3	20.0	80.4	1.1	0.0	61.2	1.53	31.8	1.5	9.1	29.9
ImageNet ResNet-152	BIM	0.0	0.0	0.0	0.0	4.7	1.4	0.0	0.0	20.5	1.4	0.0	0.0	91.6	1.4	0.0	0.4	1.36	30.6	1.6	0.0	0.1
	PGD	0.0	0.0	0.0	0.0	4.7	1.4	0.0	0.0	18.8	1.4	0.0	0.0	85.9	1.4	0.0	0.0	1.44	28.9	1.6	0.0	0.0
	MIM	0.0	0.0	0.0	0.0	2.3	1.0	0.0	0.0	4.7	1.4	0.0	0.0	81.2	1.4	0.0	0.0	1.52	23.8	1.6	0.0	0.2
	APGD	0.0	0.0	0.0	0.0	2.0	1.4	0.0	0.0	11.3	1.4	0.0	0.0	61.7	1.4	0.0	0.4	1.59	19.7	1.6	0.0	0.1
	AA	0.0	0.0	0.0	0.0	2.5	1.4	0.0	0.0	10.7	1.4	0.0	0.0	61.5	1.4	0.0	0.0	1.59	19.9	1.6	0.0	0.0
	?	0.0	0.0	0.0	0.0	3.2	1.4	0.0	0.0	13.2	1.4	0.0	0.0	76.3	1.4	0.0	0.2	1.50	24.6	1.6	0.0	0.1
ImageNet SwinT-B-IN1K	BIM	4.1	1.6	6.1	6.2	13.7	2.0	0.0	8.4	77.3	2.0	0.0	74.0	97.9	0.2	0.0	97.9	0.96	49.1	2.0	1.5	47.3
	PGD	3.9	1.6	2.3	3.1	16.4	2.0	0.0	10.9	72.7	2.0	0.0	68.8	98.4	0.2	0.0	98.4	1.01	48.6	2.0	0.6	45.9
	MIM	1.6	1.6	0.0	1.6	10.2	2.0	0.0	10.2	63.3	2.0	0.0	63.3	93.8	0.2	0.0	93.8	1.09	43.8	2.0	0.0	43.8
	APGD	1.4	1.6	0.0	1.0	5.3	2.0	0.0	4.5	32.6	2.0	0.0	25.2	65.0	0.2	0.0	51.0	1.37	29.4	2.0	0.0	23.2
	AA	1.8	1.6	0.0	1.0	5.7	2.0	0.0	4.3	31.6	2.0	0.0	25.0	68.4	0.2	0.0	54.1	1.37	29.5	2.0	0.0	23.2
	?	2.6	1.6	1.7	2.6	10.2	2.0	0.0	7.7	55.5	2.0	0.0	51.2	84.7	0.2	0.0	79.0	1.16	40.1	2.0	0.4	36.7

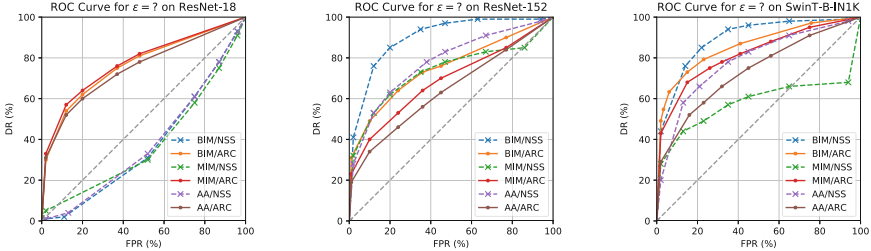


Fig. 5. ROC of SVMs in Table 1 & Table 3.

classification accuracy after correction. For ResNet-152, the least-likely label is merely relevant (not identical) to the ground truth due to network property during attack, hence leading to effective detection but not correction (this will be explained in the next subsection). In contrast, the correction method performs best on Swin-Transformer, as it can restore classification accuracy from 0.4% to 36.7% even if both the concrete type of PGD-like attack and ϵ are unknown (“Attack=?” row and “ $\epsilon=?$ ” column in Table 1), assuming flat prior. By adjusting the weights of benign examples, the decision boundary can be moved to influence FPR, as shown in Fig. 5. In particular, the proposed method performs very well for Swin Transformer, especially when FPR is required to be low.

5.2 Sequel Attack Effect as Unique Trace

The SAE is unique to PGD-like attacks, as it requires the conditions in Sect. 2 to hold for consistent effectiveness. To clarify this, we change the attack settings (quantitatively in Table 2), or the exploitation vector for ARCM (qualitatively on CIFAR10 in Fig. 4), and then review these conditions:

Table 2. Ablation of SAE uniqueness by varying attacks. The row (t1) is regarded as a baseline, and the notation “..” means “same as the baseline”. SAE will show consistent effectiveness when the conditions in Sect. 2 are satisfied.

#	Attack				ARC			ResNet-18 w/ $\epsilon = ?$				ResNet-152 w/ $\epsilon = ?$				SwinT-B-IN1K w/ $\epsilon = ?$								
	Name	L_p	Loss	Iter.	$\nabla f(\cdot)$	L_p	Loss	Label	MAE	DR	FPR	Acc	Acc*	MAE	DR	FPR	Acc	Acc*	MAE	DR	FPR	Acc	Acc*	
t1	BIM	∞	CE		Yes	Yes	∞	CE	$\hat{c}(\mathbf{x})$	1.55	30.9	1.5	10.1	30.7	1.36	30.6	1.6	0.0	0.1	0.96	49.1	2.0	1.5	47.3
t2	BIM	2	1.27	49.9	1.5	2.6	39.0	1.98	3.5	1.6	0.2	0.2	2.02	1.0	2.0	1.4	1.8
t3	BIM	..	DLR	1.98	2.1	1.5	10.5	10.6	1.63	18.9	1.6	0.0	0.6	1.44	27.5	2.0	1.8	6.6
t4	FGSM	No	1.96	3.4	1.5	30.3	29.5	1.63	18.6	1.6	8.4	6.8	1.44	27.1	2.0	44.9	32.4
t5	C&W	2	C&W	1.99	1.2	1.5	0.0	0.0	2.02	2.3	1.6	0.0	0.0	2.03	1.6	2.0	0.0	0.0
t6	FAB	..	FAB	1.99	1.0	1.5	10.6	10.5	2.00	2.5	1.6	9.2	9.2	2.03	0.8	2.0	9.4	9.4
t7	FMN	..	FMN	1.99	1.4	1.5	8.8	8.6	2.02	2.1	1.6	0.0	0.0	2.03	0.8	2.0	0.0	0.0
t8	DI-FGSM	..	DI-FGSM	..	No	1.98	2.2	1.5	42.9	42.0	1.98	3.5	1.6	27.9	27.5	1.87	8.2	2.0	67.2	62.1
t9	TL-FGSM	..	TL-FGSM	..	No	1.98	1.9	1.5	59.4	58.3	2.00	2.9	1.6	40.0	39.1	2.02	1.6	2.0	72.3	70.9
t10	NES	No	1.94	4.7	1.5	38.6	39.4	1.98	3.1	1.6	28.3	27.3	2.02	1.6	2.0	50.6	49.4
t11	SPSA	No	1.97	3.0	1.5	39.2	39.1	2.00	3.1	1.6	29.9	28.9	2.00	2.7	2.0	52.7	50.6
t12	Square	..	Square	..	No	1.99	1.6	1.5	85.7	84.3	2.02	2.1	1.6	68.6	67.4	1.84	10.2	2.0	77.9	70.1
t13	Gaussian	..	N/A	No	No	1.99	1.7	1.5	87.0	85.6	2.00	2.7	1.6	75.2	73.2	2.00	3.1	2.0	82.4	79.7
t14	Uniform	..	N/A	No	No	1.99	1.8	1.5	86.6	85.0	1.97	4.1	1.6	73.6	70.9	1.84	10.2	2.0	81.8	73.2

I. Iterative attack (Iter.). The single-step version of PGD, *i.e.*, FGSM (t4, f4) does not effectively exploit the search space within the L_p bound, and hence will not easily trigger linearity and SAE. Swin Transformer slightly reacts against FGSM due to its own characteristics of being easy to turn linear. Thus, SAE requires the attack to be iterative;

II. Gradient access ($\nabla f(\cdot)$). Transferability attacks (t8, t9) use proxy model gradients, and hence could not trigger SAE. NES (t10, f14) and SPSA (t11, f15) can be seen as PGD using gradients estimated from only network logits, but can still not trigger SAE as it cannot efficiently trigger linearity. Neither does Square attack (t12). Thus, SAE requires that the attacks use the target model gradient;

III. Same L_p bound. When the attack is BIM in L_2 bound (t2, f6), SAE is not triggered for ImageNet models, because the change of L_p influences the perturbation search process. However, SAE is still triggered for CIFAR-10 possibly due to relatively low-dimensional search space. This means CIFAR-10 property does not necessarily generalize to ImageNet. When ARC has been changed accordingly (f7, f8), the feature clusters are still separatable. Thus, SAE requires the same type of L_p bound for consistent effect;

IV. Same loss. If L_{CE} is switched to, *e.g.*, DLR [9] (t3, f11), the SAE is significantly reduced. However, if exploitation vectors are also created using DLR (f12, f13), SAE will be triggered again. Thus, SAE requires a consistent loss;

V. Relevant label. The most-likely label $\hat{c}(\tilde{\mathbf{x}})$ for exploitation vectors leads to the least significant SAE (f9). Besides, even a random label ($c?$) leads to moderate SAE (f10), while the least-likely label $\check{c}(\tilde{\mathbf{x}})$ (which is ground-truth label in many cases) leads to distinct SAE (f1). The most significant SAE correspond to $\check{c}(\tilde{\mathbf{x}}) = c(\mathbf{x})$. To maximize cross-entropy, the local linearity of a large portion of output functions $f_n(\cdot)$ has been triggered. Thus, SAE requires a relevant label (if any) for exploitation vectors.

When the exploitation vectors are created using random noise (f2, f3), SAE is not triggered. Neither does random noise as an attack trigger SAE (t13, t14, f5). Other non-PGD-like attacks (t5, t6, t7) do not trigger SAE either. A special case

Table 3. Comparison with existing methods that are compatible with our setting. Since only have a tiny amount of data is allowed in the problem setting as discussed in Sect. 1, only NSS [21] is compatible and able to properly generalize. Other existing methods require a much larger amount of data to generalize.

Method	Metric	BIM				PGD				MIM				APGD				AA								
		2/255	4/255	8/255	16/255	?	2/255	4/255	8/255	16/255	?	2/255	4/255	8/255	16/255	?	2/255	4/255	8/255	16/255	?					
ImageNet ResNet-152																										
NSS [21]	DR	2.9	19.1	39.6	47.2	41.6	2.9	19.9	39.6	46.5	41.1	4.2	31.2	41.4	9.1	32.9	1.1	12.6	28.3	35.7	29.1	1.0	11.9	29.8	33.3	28.7
	FPR	0.4	1.4	1.2	1.4	2.0	0.4	1.4	1.2	1.4	2.0	0.4	1.4	1.2	1.4	2.0	0.6	1.4	1.2	1.4	2.0	0.4	1.4	1.2	1.4	2.0
ARC	DR	0.0	4.7	20.5	91.6	30.6	0.0	4.7	18.8	85.9	28.9	0.0	2.3	4.7	81.2	23.8	0.0	2.0	11.3	61.7	19.7	0.0	2.5	10.7	61.5	19.9
	FPR	0.0	1.4	1.4	1.4	1.6	0.0	1.4	1.4	1.4	1.6	0.0	1.4	1.4	1.4	1.6	0.0	1.4	1.4	1.4	1.6	0.0	1.4	1.4	1.4	1.6
ImageNet SwinT-B-IN1K																										
NSS [21]	DR	4.5	16.2	42.4	47.5	44.2	4.9	15.8	41.8	47.1	44.1	12.3	28.7	29.3	4.5	28.9	1.6	11.0	31.3	35.5	31.1	1.4	10.4	31.8	35.1	30.8
	FPR	0.6	1.0	1.2	1.6	2.3	0.6	1.0	1.2	1.6	2.3	0.6	1.0	1.2	1.5	2.3	0.6	1.0	1.2	1.6	2.3	0.6	1.0	1.2	1.6	2.3
ARC	DR	4.1	13.7	77.3	97.9	49.1	3.9	16.4	72.7	98.4	48.6	1.6	10.2	63.3	93.8	43.8	1.4	5.3	32.6	65.0	29.4	1.8	5.7	31.6	68.4	29.5
	FPR	1.6	2.0	2.0	0.2	2.0	1.6	2.0	2.0	0.2	2.0	1.6	2.0	2.0	0.2	2.0	1.6	2.0	2.0	0.2	2.0	1.6	2.0	2.0	0.2	2.0

is targeted PGD-like attack, where the creation of exploitation vectors needs to use negative cross-entropy loss on the most-likely label to reach a similar level of effectiveness. We focus on the untargeted attack to avoid complications.

The non-PGD attacks, or PGD variants do not meet all conditions do not consistently trigger SAE across architectures, because they provide a less “matching” starting point for exploitation vectors, and hence make the BIM for exploitation vectors “restart” an attack, where the network behaves non-linear again. Only when all the conditions are satisfied will SAE be consistently triggered across different architectures, especially for ImageNet models. As for label correction, PGD-like attacks can effectively leak the ground-truth labels in adversarial examples, as long as the attack can reduce the logit value to the lowest.

In summary, SAE is the unique trace of PGD-like attacks. Although insensitive to non-PGD-like attacks, SAE is a specific signature [42] of PGD-like ones.

5.3 Comparison with Previous Detection Methods

As discussed in Sect. 2, due to our extremely tough problem setting – (1) lightweight (no additional deep model); (2) non-intrusive (does not change the model architecture or weights); (3) data-undemanding (can generalize with a tiny amount of data), the most relevant methods among related works that do not lack ImageNet evaluation are [21, 24, 27, 28, 39]. But [24, 27, 28, 39] still require a considerable amount of data to build accurate (relatively) high-dimensional statistics. The remaining NSS [21] method craft Natural Scene Statistics features, which are fed into SVM for binary classification. Namely, only NSS [21] and ARC can properly generalize with a very small amount of training samples (such as 50 images) among existing works for adversarial example detection.

In particular, the NSS [21] feature is a 18-dimensional vector, extracted from an image with a set of manually designed rules, without any trained machine learning model. Hence, it satisfies the requirements discussed previously and is

compatible to our problem setting. After extracting the features from the images, we can train standad SVMs based on these features, following Sect. 4.

We also adopt the trained SVMs in our ordinal regression framework, with a reduced training set size to 100 (the combination of 50 benign samples and 50 BIM adversarial examples based on the same set of 50 images) for each SVM for a fair comparison. All SVMs are tuned to control FPR to a very low range ($\leq 2\%$). The results and ROC curves for the uninformed attack detection task (*i.e.*, with “ $\varepsilon=?$ ”) can be found Table 3 and Fig. 5.

It is noted that (1) SVM with the 18-D NSS feature may fail to generalize due to insufficient sampling (hence the below-diagonal ROC); (2) NSS performs better for small ε , but performance saturates with larger ε , because NSS does not incorporate any cue from network gradient behavior; (3) small ε is difficult for ARC, but its performance soars with larger ε towards 100%, which is consistent and expected from our visualization; (4) SVM with ARCv can generalize against all PGD-like attacks, while NSS failed for MIM; (5) SVM with NSS may generalize against some non-PGD-like attacks [21], but not ARC due to SAE uniqueness; (6) SVM with the 2-D NSS feature (“Method 2” in [21]) fails to generalize.

Thus, ARC achieves competitive performance consistently across these settings, because it is low-dimensional, and incorporates effective gradient cues. Apart from these, ARC also provides a new perspective to understanding attack and defense from model’s gradient behavior, as discussed in Sect. 6. The motivation for the tough problem setting is also elaborated in the Appendix.

5.4 ARC for Attack Type Recognition

As discussed in Sect. 4, attack type recognition is a binary classification task to identify whether a given adversarial example is created by PGD-like attacks or not. By gathering the 14 sets (5 sets from Table 1 and 9 sets from Table 2 t4-t12) of adversarial examples involved in Table 1 and Table 2, a test dataset for attack type recognition can be constructed. As each set has the same number of samples, the binary classification accuracy can be calculated as the average of the DR for PGD-like attacks and $(1 - DR)$ for non-PGD-like attacks in the uninformed (*i.e.*, “ $\varepsilon=?$ ”) setting. The results are 74.2%, 70.2%, 74.7% for ResNet-18, ResNet-152, SwinT-B-IN1K, respectively.

This means ARC is effective in distinguishing PGD-like adversarial examples and non-PGD-like ones. It is effective because our proposed ARC is specific to PGD-like attacks and does not respond to other types of attacks.

6 Discussions and Justifications

Ordinal Regression. Intuitively, the uninformed attack detection can be formulated as a standard regression to estimate a continuous k value. However, this introduces an undesired additional threshold hyper-parameter for deciding

whether an input with *e.g.*, 0.5 estimation is adversarial. Ordinal regression produces discrete k values and avoids such ambiguity and unnecessary parameter.

Training Set Size. Each SVM has only 100 training data (*i.e.*, 50 benign + 50 adversarial). The 2-D ARCV distribution being so simple that can be described by a small amount of data points (see Fig. 2), allows an SVM to generalize with less than 100 data points. The performance gain becomes marginal with 200 training samples or more, as feature distribution is already well represented.

Combination with Adversarial Training. From our experiment and recent works [14, 29, 38],

its noted that (1) small perturbations are hard to detect, but easy to defend; while (2) large perturbations are hard to defend, but easy to detect. However, combining defense and our detection is not effective on ImageNet. As shown in Fig. 6, we compute ARCM based on regular ResNet-50 (from PyTorch) and adversarially trained ResNet-50 on ImageNet (from [14]).

Unlike the regular ResNet-50, adversarially trained one has a much higher mean value in ARCM, resulting in almost non-separable ARCV. This means adversarial training makes the model very linear around the data [40]. Namely, the network is trained to generalize while being already very linear to the input, and thus it will be hard to make the model behave even more linear to manipulate the output by the attack to achieve a specific goal.

Limitations. This paper focuses on characterizing a specific type of attacks, instead of a general detection or defense method. The following are the limitations of the ARC feature in potential applications: (1) The SAE is unique but specific to only PGD-like attacks; (2) Jacobian computation is very slow for ImageNet models because it requires 1000 iterations of backward pass. Thus, we are unable to evaluate on all ImageNet data with 2 Nvidia Titan Xp GPUs.

7 Conclusions

We design the Adversarial Response Characteristics (ARC) feature with the intuition that a model behaves “linearly” against adversarial examples, in which PGD-like attacks will leave a unique trace. The ARC feature can characterize PGD-like attacks, when the “unique trace”, namely the Sequel Attack Effect (SAE) is observed. Extensive qualitative visualizations and quantitative experiments demonstrate the effectiveness of the proposed ARC/SAE. In particular, our method is effective across both CNN-based and Transformer-based architectures, multiple perturbation levels, and different datasets.

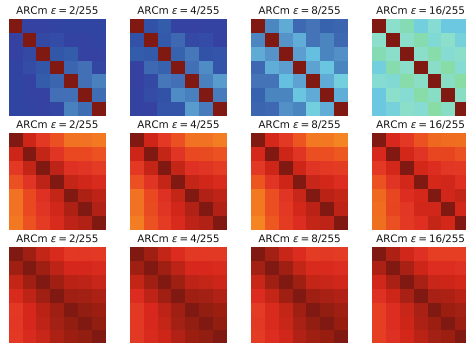


Fig. 6. ARCM from regular (1st row), and robust ResNet-50 (2nd row w/ $\epsilon=4/255$, 3rd row w/ $\epsilon=8/255$). Note, we use the same colorbar as shown in Fig. 1.

References







1. Aldahdooh, A., et al.: Adversarial example detection for DNN models: a review and experimental comparison. *Artif. Intell. Rev.* **55**(6), 4403–4462 (2022)
2. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search (2020)
3. Andriushchenko, M., Flammarion, N.: Understanding and improving fast adversarial training. In: *NeurIPS*, vol. 33, pp. 16048–16059 (2020)
4. Athalye, A., et al.: Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In: *ICML*, pp. 274–283. PMLR (2018)
5. Bartlett, P., Bubeck, S., Cherapanamjeri, Y.: Adversarial examples in multi-layer random ReLU networks. In: *NeurIPS*, vol. 34, pp. 9241–9252 (2021)
6. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: bypassing ten detection methods. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14 (2017)
7. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: *IEEE S&P*, pp. 39–57. IEEE (2017)
8. Croce, F., Hein, M.: Minimally distorted adversarial examples with a fast adaptive boundary attack. In: *ICML*, pp. 2196–2205. PMLR (2020)
9. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: *ICML* (2020)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: *CVPR*, pp. 248–255. IEEE (2009)
11. Dong, Y., et al.: Benchmarking adversarial robustness on image classification. In: *CVPR* (June 2020)
12. Dong, Y., et al.: Boosting adversarial attacks with momentum. In: *CVPR* (June 2018)
13. Dong, Y., Pang, T., Su, H., Zhu, J.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: *CVPR*, pp. 4312–4321 (2019)
14. Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., Tsipras, D.: Robustness (python library) (2019)
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. *ICLR* (2015)
16. Gopalakrishnan, S., Marzi, Z., Madhoo, U., Pedarsani, R.: Combating adversarial attacks using sparse representations. *ICLRw* (2018)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint [arXiv:1512.03385](https://arxiv.org/abs/1512.03385) (2015)
18. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: *ICML*, pp. 2137–2146. PMLR (2018)
19. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. In: *NeurIPS*, vol. 32 (2019)
20. Khaloeei, M., Mehdi Homayounpour, M., Amirmazlaghani, M.: Layer-wise regularized adversarial training using layers sustainability analysis framework. *Neurocomput.* **540**(C), 126182 (2023)
21. Kherchouche, A., et al.: Detection of adversarial examples in deep neural networks with natural scene statistics. In: *IJCNN*, pp. 1–7 (2020)
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
23. Kurakin, A., Goodfellow, I., Bengio, S., et al.: Adversarial examples in the physical world. *ICLRw* (2016)

24. Li, X., Li, F.: Adversarial examples detection in deep networks with convolutional filter statistics. In: ICCV, pp. 5775–5783 (2017)
25. Liao, F., et al.: Defense against adversarial attacks using high-level representation guided denoiser. CoRR arxiv preprint [arxiv:abs/1712.02976](https://arxiv.org/abs/1712.02976) (2017)
26. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: ICCV (2021)
27. Lu, J., Issaranon, T., Forsyth, D.A.: SafetyNet: detecting and rejecting adversarial examples robustly. CoRR arxiv preprint [arxiv:abs/1704.00103](https://arxiv.org/abs/1704.00103) (2017)
28. Ma, S., Liu, Y.: NIC: detecting adversarial samples with neural network invariant checking. In: NDSS (2019)
29. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. ICLR (2018)
30. McMahan, B., et al.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS, vol. 54, pp. 1273–1282 (2017)
31. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: ACM SIGSAC, pp. 135–147. CCS '17 (2017)
32. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. arXiv preprint [arXiv:1702.04267](https://arxiv.org/abs/1702.04267) (2017)
33. Nayak, G.K., Rawal, R., Chakraborty, A.: Dad: data-free adversarial defense at test time. In: WACV, pp. 3562–3571 (2022)
34. Niu, Z., Zhou, M., Wang, L., Gao, X., Hua, G.: Ordinal regression with multiple output CNN for age estimation. In: CVPR (June 2016)
35. Pang, T., Du, C., Dong, Y., Zhu, J.: Towards robust detection of adversarial examples. In: NeurIPS, pp. 4584–4594. NIPS'18 (2018)
36. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. JMLR **12**, 2825–2830 (2011)
37. Pintor, M., et al.: Fast minimum-norm adversarial attacks through adaptive norm constraints. In: NeurIPS, vol. 34, pp. 20052–20062 (2021)
38. Qin, C., et al.: Adversarial robustness through local linearization. In: NeurIPS (2019)
39. Roth, K., Kilcher, Y., Hofmann, T.: The odds are odd: a statistical test for detecting adversarial examples. In: ICML, vol. 97, pp. 5498–5507 (2019)
40. Roth, K., Kilcher, Y., Hofmann, T.: Adversarial training is a form of data-dependent operator norm regularization. In: NeurIPS, vol. 33, pp. 14973–14985 (2020)
41. Song, Y., et al.: PixelDefend: leveraging generative models to understand and defend against adversarial examples. In: ICLR (2018)
42. Souri, H., Khorramshahi, P., Lau, C.P., Goldblum, M., Chellappa, R.: Identification of attack-specific signatures in adversarial examples. CoRR arxiv preprint [arxiv:abs/2110.06802](https://arxiv.org/abs/2110.06802) (2021)
43. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint [arXiv:1312.6199](https://arxiv.org/abs/1312.6199) (2013)
44. Tramer, F., Carlini, N., Brendel, W., Madry, A.: On adaptive attacks to adversarial example defenses. NeurIPS **33**, 1633–1645 (2020)
45. Uesato, J., O’donoghue, B., Kohli, P., Oord, A.: Adversarial risk and the dangers of evaluating against weak attacks. In: ICML, pp. 5025–5034. PMLR (2018)
46. Virtanen, P., et al.: SciPy 1.0: fundamental algorithms for scientific computing in python. Nat. Methods **17**, 261–272 (2020)
47. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. In: NeurIPS (2020)

48. Xie, C., Wu, Y., van der Maaten, L., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: CVPR (June 2019)
49. Xie, C., et al.: Improving transferability of adversarial examples with input diversity. In: CVPR, IEEE (2019)
50. Yin, X., Kolouri, S., Rohde, G.K.: Gat: generative adversarial training for adversarial example detection and robust classification. In: ICLR (2020)



CAB-KWS: Contrastive Augmentation: An Unsupervised Learning Approach for Keyword Spotting in Speech Technology

Weinan Dai¹(✉) , Yifeng Jiang² , Yuanjing Liu³ , Jinkun Chen⁴ ,
Xin Sun⁵ , and Jinglei Tao³ 

¹ Trine University, Phoenix, USA
wnd17460@gmail.com

² Boston University, Boston, USA
yjiang8@bu.edu

³ Georgia Institute of Technology, Atlanta, GA, USA
{yliu3689, jinglei.tao}@gatech.edu

⁴ Faculty of Computer Sciences, Dalhousie University, Halifax, Canada
jinkun.chen@dal.ca

⁵ Texas A&M University, College Station, TX, USA

Abstract. This paper addresses the persistent challenge in Keyword Spotting (KWS), a fundamental component in speech technology, regarding the acquisition of substantial labeled data for training. Given the difficulty in obtaining large quantities of positive samples and the laborious process of collecting new target samples when the keyword changes, we introduce a novel approach combining unsupervised contrastive learning and a unique augmentation-based technique. Our method allows the neural network to train on unlabeled data sets, potentially improving performance in downstream tasks with limited labeled data sets. We also propose that similar high-level feature representations should be employed for speech utterances with the same keyword despite variations in speed or volume. To achieve this, we present a speech augmentation-based unsupervised learning method that utilizes the similarity between the bottleneck layer feature and the audio reconstructing information for auxiliary training. Furthermore, we propose a compressed convolutional architecture to address potential redundancy and non-informative information in KWS tasks, enabling the model to simultaneously learn local features and focus on long-term information. This method achieves strong performance on the Google Speech Commands V2 Dataset. Inspired by recent advancements in sign spotting and spoken term detection, our method underlines the potential of our contrastive learning approach in KWS and the advantages of Query-by-Example Spoken Term Detection strategies. The presented CAB-KWS provide new perspectives in the field of KWS, demonstrating effective ways to reduce data collection efforts and increase the system's robustness.

W. Dai and Y. Jiang—Contributed equally to this work.

Keywords: key word spotting · contrastive learning · unsupervised learning

1 Introduction

Keyword Spotting (KWS) is a fundamental application in the field of speech technology, playing a pivotal role in real-world scenarios, particularly in the context of interactive agents such as virtual assistants and voice-controlled devices. KWS is designed to detect a small set of pre-defined keywords within an audio stream. This capability is crucial for two primary reasons. First, it enables the initiation of interactions through specific commands like “hey Siri” or “OK, Google,” effectively serving as an explicit cue for the system to start processing subsequent speech. Second, KWS can identify sensitive words within a conversation, thereby playing a vital role in protecting the privacy of the speaker. Given these applications, it is crucial to develop accurate and reliable KWS systems for effective real-world speech processing [9, 11, 18].

Despite the considerable advancements in KWS, a significant challenge that persists is the acquisition of sufficient labeled data for training. This is especially true for positive samples, which are often harder to obtain in large quantities. This issue is further exacerbated when the keyword changes, as it necessitates the collection of new target samples, a process that can be both time-consuming and resource-intensive. To address these challenges, we propose a novel approach that leverages the power of unsupervised contrastive learning and a unique augmentation-based method. Additionally, another potential problem is redundant information, speeches are noisy and complex, where only some key phrases are highly related to the keywords. However, convolutional methods treat all the word windows equally, ignoring that different words have different importance and should be weighted differently within word windows. Besides, the sliding windows used in the convolutional methods produce a lot of redundant information. Thus, it is important to reduce the non-informative and redundant information and distinguish the contributions of different convolutional features.

Our method enables the neural network to be trained on unlabeled datasets, reducing the reliance on extensive labeled data. This technique can greatly enhance the performance of downstream tasks, even in scenarios where labeled datasets are scarce. Additionally, we propose that speech utterances containing the same keyword, regardless of variations in speed or volume, should exhibit similar high-level feature representations in KWS tasks. To achieve this, we present a speech augmentation-based unsupervised learning approach. This method leverages the similarity of bottleneck layer features, along with audio reconstruction information, for auxiliary training to improve system robustness.

In addition to these innovations, we propose a compressed convolutional architecture for the KWS task. This architecture, designed to tackle the issue of redundant information, has demonstrated strong performance on the Google Speech Commands V2 Dataset. By doing so, it enables the model to learn local features and focus on long-term information simultaneously, thereby enhancing its performance on the KWS task.

Our approach is inspired by recent advancements in the field of sign spotting and spoken term detection. For instance, Varol et al. [21] demonstrated the effectiveness of Noise Contrastive Estimation and Multiple Instance Learning in sign spotting, which could provide insights into the use of contrastive learning in KWS. Similarly, the works of Tejedor et al. [19, 20] on Query-by-Example Spoken Term Detection (QbE STD) highlight the potential of QbE STD strategies in outperforming text-based STD in unseen data domains, reinforcing the potential advantages of our proposed method.

Our major contributions in this work are as follows:

- We introduce a compact convolutional architecture for the KWS task that achieves strong results on the Google Speech Commands V2 Dataset.
- We develop an unsupervised loss and a contrastive loss to evaluate the similarity between original and augmented speech, as well as the proximity within each minibatch.
- We introduce a speech augmentation-based unsupervised learning approach, utilizing the similarity between the bottleneck layer feature, as well as the audio reconstructing information for auxiliary training.

The remainder of this paper is structured as follows. Section 2 provides an overview of related work in the areas of data augmentation, unsupervised learning, and other methodologies of KWS tasks. Section 3 offers a background on contrastive learning. Section 4 details the proposed model architecture and our augmentation-based unsupervised contrastive learning loss. Section 5 discusses the configuration, research questions, and experimental setups. Section 6 presents the experimental results and compares them with other pre-training methods. We also discuss the relationship between pre-training steps and the performance of downstream KWS tasks. Finally, Sect. 7 concludes the paper with a summary of our findings and potential avenues for future work.

2 Related Work

Data augmentation is widely acknowledged as an effective technique for enriching the training datasets in speech applications, such as Automatic Speech Recognition (ASR) and Keyword Spotting (KWS). Various methods have been explored, such as vocal tract length perturbation [5], speed-perturbation [8], and the introduction of noisy audio signals [4]. More recently, spectral-domain augmentation techniques, such as SpecAugment [15] and WavAugment [7], have been developed to further improve the robustness of speech recognition systems. In this work, we extend these efforts by applying speed and volume perturbation in our speech augmentation method.

While supervised learning has been the primary approach in the KWS area, it often requires large amounts of labeled data, which can be challenging to obtain, especially for less frequently used languages. This has sparked growing interest in weakly supervised and unsupervised approaches. For example, Noisy Student Training, a semi-supervised learning technique, has been employed in ASR [16]

and subsequently adapted for robust keyword spotting [17]. Additionally, unsupervised methods for KWS have been investigated [3, 10, 25], yielding promising outcomes. Building on these efforts, we propose an unsupervised learning framework for the keyword spotting task in this paper.

The Google Speech Commands V2 Dataset is a widely used benchmark for novel ideas in KWS. Numerous works have performed experiments on this dataset, introducing various architectures and methods. For instance, a convolutional recurrent network with attention was introduced by [2], and a deep residual network, MatchboxNet, was proposed by [12]. More recently, an edge computing-focused model called EdgeCRNN [24] was introduced, along with a method that integrates triplet loss-based embeddings with a modified K-Nearest Neighbor (KNN) for classification [22]. In this work, we also evaluate our speech augmentation-based unsupervised learning method on this dataset and compare it with other unsupervised approaches, including CPC [13], APC [1], and MPC [6].

3 Preliminary Study of Contrastive Learning

In the context of a classification task involving K classes, we consider a dataset $\{\mathbf{x}_i, y_i\}_{i=1}^N$ with N training samples. Each $\mathbf{x}_i \in \mathbb{R}^L$ represents an input sentence of L words, and each $y_i \in 1, 2, \dots, K$ is the corresponding label. We denote the set of training sample indexes by $\mathcal{I} = 1, 2, \dots, N$ and the set of label indexes by $\mathcal{K} = 1, 2, \dots, K$.

We explore the realm of self-supervised contrastive learning, a technique that has demonstrated its effectiveness in numerous studies. Given N training samples $\{\mathbf{x}_i\}_{i=1}^N$ with a number of augmented samples, the standard contrastive loss is defined as follows:

$$\mathcal{L}_{\text{self}} = \frac{1}{N} \sum_{i \in \mathcal{I}} - \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_{j(i)}/\tau)}{\sum_{a \in \mathcal{A}_i} \exp(\mathbf{z}_i \cdot \mathbf{z}_a/\tau)} \quad (1)$$

Here, \mathbf{z}_i is the normalized representation of \mathbf{x}_i , $\mathcal{A}_i := \mathcal{I} \setminus i$ is the set of indexes of the contrastive samples, the \cdot symbol denotes the dot product, and $\tau \in \mathbb{R}^+$ is the temperature factor.

However, self-supervised contrastive learning does not utilize supervised signals. A previous study [Khosla et al., 2020] incorporated supervision into contrastive learning in a straightforward manner. It simply treated samples from the same class as positive samples and samples from different classes as negative samples. The following contrastive loss is defined for supervised tasks:

$$\mathcal{L}_{\text{sup}} = \frac{1}{N} \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{P}_i|} \sum_{p \in \mathcal{P}_i} - \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p/\tau)}{\sum_{a \in \mathcal{A}_i} \exp(\mathbf{z}_i \cdot \mathbf{z}_a/\tau)} \quad (2)$$

Despite its effectiveness, this approach still requires learning a linear classifier using the cross-entropy loss apart from the contrastive term. This is because the contrastive loss can only learn generic representations for the input examples. Thus, we argue that the supervised contrastive learning developed so far appears to be a naive adaptation of unsupervised contrastive learning to the classification.

4 Proposed Method

The keyword spotting task can be framed as a sequence classification problem, where the keyword spotting network maps an input audio sequence $X = \{x_0, x_1, \dots, x_T\}$ to a set of keyword classes $Y \in y_{1:S}$. Here, T represents the number of frames, and S denotes the number of classes. Our proposed keyword spotting model, depicted in Fig. 1(A), consists of five key components: (1) Compressed Convolutional Layer, (2) Transformer Block, (3) Feature Selection Layer, (4) Bottleneck Layer, and (5) Projection Layer.

4.1 Compressed Convolutional Layer

The Compressed Convolutional Layer replaces the CNN block in the original design. This layer learns dense and informative frame representations from the input sequence X . Specifically, it utilizes convolutional neural networks (CNNs), an attention-based soft-pooling approach, and residual convolution blocks for feature extraction and compression.

Frame Convolution. Just as in the original CNN block, the convolution operation is applied to each frame. Given the input sequence X and the i -th filter, the convolution for the j -th frame is expressed as

$$\mathbf{x}_j^i = \text{conv}(\{\mathbf{x}_j, \mathbf{x}_{j+1}, \dots, \mathbf{x}_{j+k_i-1}\}; \mathbf{W}_x^i), \quad (3)$$

where \mathbf{W}_x^i is the learned parameter of the i -th filter.

Attention-Based Soft-Pooling. To eliminate redundant information in the speech dataset, we propose an attention-based soft-pooling operation on the frame representations learned by the previous equation. Specifically, given a frame x_j , its neighboring frames $\{x_{j+1}, \dots, x_{j+g-1}\}$, and the corresponding filter f_i , we first learn the local-based attention scores $\alpha_j^i = \mathbf{W}_\alpha^i \mathbf{x}_j^i + b$ with softmax function, and then conduct the soft-pooling operation to obtain the compressed representation as in the following equation:

$$\mathbf{o}_p^i = \sum_{q=j}^{j+g-1} \beta_q^i \mathbf{x}_q^i \quad (4)$$

Residual Convolution Block. We now have a denoised matrix $\{\mathbf{o}_1^i, \mathbf{o}_2^i, \dots, \mathbf{o}_p^i\}$ that represents the input sequence X . To avoid vanishing gradients and facilitate model training, we introduce residual blocks on top of the compressed features. In particular, we replace the batch norm layer with the group norm layer. Let a denotes the number of residual blocks, we have

$$\mathbf{r}_p^i = \text{ResidualBlock}(\{\mathbf{o}_p^i, \dots, \mathbf{o}_{p+a-1}^i\}), \quad (5)$$

where ResidualBlock is the operation of the residual convolution block.

4.2 ResLayer Block

Transformer Block. The output from the Compressed Convolutional Layer, $R = \{r_1^i, r_2^i, \dots, r_p^i\}$, is then fed into the Transformer Block. This block captures long-term dependencies in the sequence via the self-attention mechanism: $E_{tran} = \text{Self-Attention} \times M (R)$, where M is the number of self-attention layers.

Feature Selecting Layer. Following the Transformer Block, the Feature Selecting Layer is implemented to extract keyword information from the sequence E_{tran} .

$$E_{feat} = \text{Concat} (E_{tran} [T - r, T]), \quad (6)$$

Here, the last r frames of E_{tran} are gathered, and all the collected frames are concatenated together into one feature vector E_{feat} .

Bottleneck and Project Layers. After the Feature Selecting Layer, a Bottleneck Layer and a Projection Layer are added. These layers map the hidden states to the predicted classification classes \tilde{Y} .

$$E_{bn} = \text{FC}_{bn} (E_{feat}), \quad (7)$$

$$\tilde{Y} = \text{FC}_{proj} (E_{bn}), \quad (8)$$

Finally, the cross-entropy (CE) loss for supervised learning and model fine-tuning is computed based on the predicted classes \tilde{Y} and ground truth classes Y . $\mathcal{L}_{ce} = \text{CE}(Y, \tilde{Y})$.

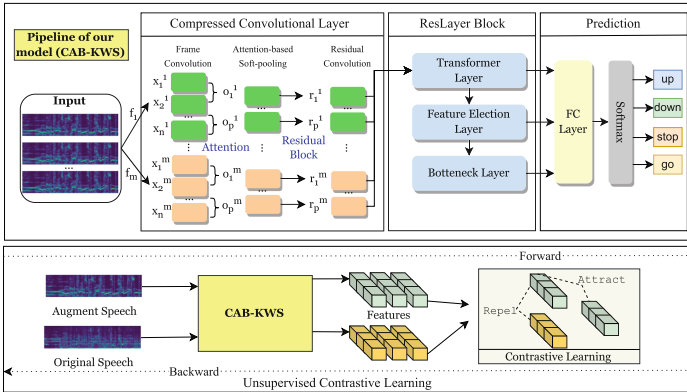


Fig. 1. A: The architecture of our CAB-KWS for the keyword spotting task consists of a compressed layer, ResLayer Block, and Decision Block. B: The proposed method integrates speech augmentation with unsupervised and contrastive learning for audio processing.

4.3 Augmentation Method

Data augmentation is a widely utilized technique to enhance model performance and robustness, particularly in speech-related tasks. In this study, we delve into speed and volume-based augmentation in the context of unsupervised learning for keyword detection. A specific audio sequence, represented as $X = A(t)$, is defined by its amplitude A and time index t .

Regarding speed augmentation, a speed ratio symbolized by λ_{speed} is established to modify the speed of X . The following formula describes this process: $X^{\text{aug}} = A(\lambda_{\text{speed}} t)$. For volume augmentation, similarly, we set an intensity ratio, λ_{volume} , to alter the volume of X , as presented in the following equation: $X^{\text{aug}} = \lambda_{\text{volume}} A(t)$. By using various ratios λ_{speed} and λ_{volume} , we can generate multiple pairs of speech sequences, (X, X^{aug}) , to facilitate the training of the audio representation network via unsupervised learning. The fundamental assumption is that speech utterances, regardless of speed or volume variations, should exhibit similar high-level feature representations for keyword-spotting tasks.

4.4 Contrastive Learning Loss

We aim to align the softmax transform of the dot product between the feature representation \mathbf{z}_i and the classifier $\boldsymbol{\theta}_i$ of the input example X_i with its corresponding label. Let $\boldsymbol{\theta}_i^*$ denote the column of $\boldsymbol{\theta}_i$ that corresponds to the ground-truth label of \mathbf{x}_i . We aim to maximize the dot product $\boldsymbol{\theta}_i^{*T} \mathbf{z}_i$. To achieve this, we learn a better representation of $\boldsymbol{\theta}_i$ and \mathbf{z}_i using supervised signals.

The Dual Contrastive Loss exploits the relation between different training samples to maximize $\boldsymbol{\theta}_i^{*T} \mathbf{z}_j$ if \mathbf{x}_j has the same label as \mathbf{x}_i , while minimizing $\boldsymbol{\theta}_i^{*T} \mathbf{z}_j$ if \mathbf{x}_j carries a different label from \mathbf{x}_i .

To define the contrastive loss, given an anchor \mathbf{z}_i originating from the input example \mathbf{x}_i , we take $\{\boldsymbol{\theta}_j^*\}_{j \in \mathcal{P}_i}$ as positive samples and $\{\boldsymbol{\theta}_j^*\}_{j \in \mathcal{A}_i \setminus \mathcal{P}_i}$ as negative samples. The contrastive loss is defined as follows:

$$\mathcal{L}_z = \frac{1}{N} \sum_i i \in \mathcal{I} \frac{1}{|\mathcal{P}_i|} \sum_{p \in \mathcal{P}_i} p - \log \frac{\exp(\boldsymbol{\theta}_p \cdot \mathbf{z}_i / \tau)}{\sum_{a \in \mathcal{A}_i} \exp(\boldsymbol{\theta}_a \cdot \mathbf{z}_i / \tau)} \quad (9)$$

Here, $\tau \in \mathbb{R}^+$ is the temperature factor, $\mathcal{A}_i := \mathcal{I} \setminus i$ is the set of indexes of the contrastive samples, $\mathcal{P}_i := \{p \in \mathcal{A}_i : y_p = y_i\}$ is the set of indexes of positive samples, and $|\mathcal{P}_i|$ is the cardinality of \mathcal{P}_i . Similarly, given an anchor $\boldsymbol{\theta}_i^*$, we take $\{\mathbf{z}_j\}_{j \in \mathcal{P}_i}$ as positive samples and $\{\mathbf{z}_j\}_{j \in \mathcal{A}_i \setminus \mathcal{P}_i}$ as negative samples. The contrastive loss is defined as follows:

$$\mathcal{L}_\theta = \frac{1}{N} \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{P}_i|} \sum_{p \in \mathcal{P}_i} p - \log \frac{\exp(\boldsymbol{\theta}_i \cdot \mathbf{z}_p / \tau)}{\sum_{a \in \mathcal{A}_i} \exp(\boldsymbol{\theta}_i \cdot \mathbf{z}_a / \tau)} \quad (10)$$

Finally, Dual Contrastive Loss is the combination of the above two contrastive loss terms:

$$\mathcal{L}_{\text{Dual}} = \mathcal{L}_z + \mathcal{L}_\theta \quad (11)$$

As illustrated in Fig. 1(B), the structure of the proposed unsupervised learning method rooted in augmentation, involves two primary steps akin to other unsupervised strategies: (1) unsupervised data undergoes initial pre-training and (2) supervised KWS data is then fine-tuned. The pre-training phase sees the extraction of a bottleneck feature by training the unlabelled speech, which is subsequently used for KWS prediction in the fine-tuning stage.

In pre-training, the paired speech data (X, X^{aug}) is fed into CNN-Attention models with identical parameters. Since X^{aug} is derived from X , the unsupervised method we've developed assumes that both X and X^{aug} will yield analogous high-level bottleneck features. This implies the speech content remains identical regardless of the speaker's speed or volume. The network's optimization, therefore, must highlight the similarity between X and X^{aug} . The Mean Square Error (MSE) \mathcal{L}_{sim} is utilized to determine the distance between X and X^{aug} 's output.

$$\mathcal{L}_{sim} = \frac{1}{U_{bn}} \sum_{u=0}^{U_{bn}} |E_{bn}(u) - E_{bn}^{aug}(u)|^2 \quad (12)$$

In this context, U_{bn} represents the dimensions of the bottleneck feature vector, while E_{bn} and E_{bn}^{aug} correspond to the bottleneck layer outputs for the original speech X and the augmented speech X^{aug} , respectively.

The network also includes an auxiliary training branch designed to predict the average feature of the speech segment input, helping the network learn the intrinsic characteristics of speech utterances. To achieve this, the average vector of the input Fbank vector X is first calculated along the time axis t . A reconstruction layer connected to the bottleneck layer is then used to reconstruct this average Fbank vector \tilde{X} . The MSE loss \mathcal{L}_x is applied to measure the similarity between the original and reconstructed audio vectors along the feature dimension U_x .

$$\mathcal{X} = \frac{1}{T} \sum_T (X) \quad \tilde{X} = \text{FCreconstruct}(E_{bn}) \quad \mathcal{L}_x = \frac{1}{U_x} \sum_{u=0}^{U_x} |\mathcal{X}(u) - \tilde{X}(u)|^2 \quad (13)$$

In this context, U_x denotes the dimension of the Fbank feature vector, and \mathcal{X} represents the mean vector of X . The loss \mathcal{L}_x^{aug} between the augmented average audio \mathcal{X}^{aug} and the reconstructed feature \tilde{X}^{aug} can be similarly defined as:

$$\mathcal{L}_x^{aug} = \frac{1}{U_x} \sum_{u=0}^{U_x} |\mathcal{X}^{aug}(u) - \tilde{X}^{aug}(u)|^2 \quad (14)$$

Hence, the final unsupervised learning (UL) loss function \mathcal{L}_{ul} comprises of the three aforementioned losses \mathcal{L}_{sim} , \mathcal{L}_x , and \mathcal{L}_x^{aug}

$$\mathcal{L}_{ul} = \lambda_1 \mathcal{L}_{sim} + \lambda_2 \mathcal{L}_x + \lambda_3 \mathcal{L}_x^{aug} + \lambda_4 \mathcal{L}_{Dual} \quad (15)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the factor ratios of each loss component.

In the fine-tuning stage, the average feature prediction branch is discarded, and a projection layer, followed by a softmax layer, is added after the bottleneck layer for KWS prediction. The original network’s parameters can either be kept fixed or adjusted during fine-tuning. Our experiments indicate that adjusting all parameters enhances performance, so we choose to update all parameters during this phase.

5 Experiment Setup

In this section, we evaluated the proposed method on keyword spotting tasks by implementing our CNN-Attention model with supervised training and comparing it to Google’s model. An ablation study was conducted to examine the impact of speed and volume augmentation on unsupervised learning. Additionally, we compared our approach with other unsupervised learning methods, including CPC, APC, and MPC, using their published networks and hyperparameters without applying any additional experimental tricks [23]-[25]. We also analyzed how varying pre-training steps influence the performance and convergence of the downstream KWS task.

5.1 Datasets

We used Google’s Speech Commands V2 Dataset [23] for evaluating the proposed models. The dataset contains more than 100k utterances. Total 30 short words were recorded by thousands of different people, as well as background noise such as pink noise, white noise, and human-made sounds. The KWS task is to discriminate among 12 classes: “yes”, “no”, “up”, “down”, “left”, “right”, “on”, “off”, “stop”, “go”, unknown, or silence. The dataset was split into training, validation, and test sets, with 80% training, 10% validation, and 10% test. This results in about 37000 samples for training, and 4600 each for validation and testing. We applied the HuNonspeech ¹ real noisy data to degrade the original speech. In our experiments, this strategy was executed using the Aurora4 tools ². Each utterance was randomly corrupted by one of 100 different types of noise from the HuNonspeech dataset. The Signal Noise Ratio (SNR) for each utterance ranged from 0 to 20 dB, with an average SNR of 10 dB across all datasets (Table 1).

Table 1. Results comparison of KWS Model, Classification Accuracy (%)

Model Name	Supervised Training Data	Dev	Eval
Sainath and Parada (Google)	Speech Commands	-	84.7
CAB-KWS (w/o volume)	Speech Commands	86.4	85.3
CAB-KWS & speed augment	Speech Commands	87.3	85.8

Table 2. Ablation study, the effect of speed and volume augmentation, classification accuracy (%)

Model Name	Pre-training Data	Fine-tuning Data	Dev	Eval
CAB-KWS + vo-pre.	Speech Commands	Speech Commands	86.1	85.9
CAB-KWS + sp-pre.	Speech Commands	Speech Commands	87.8	86.9
CAB-KWS + vo-sp-pre.	Speech Commands	Speech Commands	87.9	87.2
CAB-KWS + vo-sp-pre-contras.	Speech Commands	Speech Commands	88.1	88.3
CAB-KWS + vo-pre.	Librispeech-100	Speech Commands	86.3	86.0
CAB-KWS + sp-pre.	Librispeech-100	Speech Commands	87.9	87.9
CAB-KWS + vo-sp-pre.	Librispeech-100	Speech Commands	88.2	88.1
CAB-KWS + vo-sp-pre-contras &	Librispeech-100	Speech Commands	88.4	88.5

“vo-pre.” means volume pre-training; “sp-pre.” is speed pre-training; “vo-sp-pre.” indicates volume & speed pre-training; “contras.” is contrastive learning.

As with other unsupervised approaches, a large unlabeled corpus, consisting of 100 h of clean Librispeech [14] audio, was used for network pre-training through unsupervised learning. Initially, the long utterances were divided into 1-second segments to align with the Speech Commands dataset. Following this, the clean segments were mixed with noisy HuNonspeech data using Aurora 4 tools, employing the same corruption mechanism as the Speech Commands.

5.2 Model Setup

The model architecture consists of:

- CNN blocks with 2 layers, a 3x3 kernel size, 2x2 stride, and 32 channels.
- Transformer layer with 2 layers, a 320-dimensional embedding space, and 4 attention heads.
- Feature Selecting Layer retains the last 2 frames with a 2x320 dimension.
- Bottleneck Layer with a single fully connected (FC) layer of 800 dimensions.
- Project Layer with one FC layer outputting a 12-dimensional softmax.
- Reconstruct Layer with one FC layer outputting a 40-dimensional softmax.

The factor ratio is set to $\lambda_1 = 0.8$, $\lambda_2 = 0.05$, $\lambda_3 = 0.05$, and $\lambda_4 = 0.1$.

To demonstrate the effectiveness, we compared with other approaches:

- Supervised Learning: Used Google’s Sainath and Parada’s model as baseline.
- Unsupervised Learning:
 - Contrastive Predictive Coding (CPC): Learns representations via next step prediction.
 - Autoregressive Predictive Coding (APC): Optimizes L1 loss between input and output sequences.
 - Masked Predictive Coding (MPC): Utilizes Transformer with Masked Language Model (MLM) structure for predictive coding, incorporating dynamic masking.

6 Experimental Results

6.1 Comparison of KWS Model (RQ1)

The table compares the classification accuracy of three different KWS models: (1) the model by Sainath and Parada (Google), (2) the CAB-KWS model without volume augmentation, and (3) the CAB-KWS model with speed augment. It can be observed that the CAB-KWS model with speed augment achieved the highest classification accuracy on both the development (Dev) and evaluation (Eval) datasets. This research question aims to investigate how the inclusion of data augmentation techniques, specifically speed augment in this case, improves the performance of KWS models compared to models without these techniques. The results could be used to guide future development of KWS models and to optimize their performance for various applications.

6.2 Ablation Study (RQ2)

The CAB-KWS keyword spotting model is an advanced solution designed to improve the classification accuracy of speech recognition tasks. The ablation study presented in the table focuses on evaluating the impact of different pre-training techniques, such as volume pre-training, speed pre-training, combined volume and speed pre-training, and combined volume, speed, and contrastive learning pre-training, on the model's performance. By comparing the classification accuracy of CAB-KWS when fine-tuned on two datasets, Speech Commands and Librispeech-100, we can better understand the effectiveness of these pre-training techniques and their combinations.

Firstly, Table 2 shows that the CAB-KWS model with speed pre-training (sp-pre.) outperforms the model with volume pre-training (vo-pre.) in both datasets. This result indicates that speed pre-training is more effective in enhancing the model's classification accuracy than volume pre-training. However, the combination of volume and speed pre-training (vo-sp-pre.) further improves the model's performance, demonstrating that utilizing both techniques can lead to better keyword spotting results.

Moreover, the inclusion of contrastive learning (contras.) in the pre-training process yields the highest classification accuracy in both Speech Commands and Librispeech-100 datasets. The CAB-KWS model with combined volume, speed, and contrastive learning pre-training (vo-sp-pre-contras.) outperforms all other models, highlighting the benefits of incorporating multiple pre-training methods. This result emphasizes the goodness of the CAB-KWS model, as it demonstrates its adaptability and capability to leverage various pre-training techniques to enhance its performance.

The CAB-KWS model's strength lies in its ability to capitalize on different pre-training methods, which can be tailored to suit specific datasets and tasks. By combining these techniques, the model can learn more robust and diverse representations of the data, leading to improved classification accuracy. This adaptability makes the CAB-KWS model particularly suitable for a wide range

of applications in keyword spotting and speech recognition tasks, where performance and generalizability are of utmost importance.

In conclusion, the goodness of the CAB-KWS keyword spotting model is showcased through its ability to integrate various pre-training techniques, such as volume pre-training, speed pre-training, and contrastive learning, to improve classification accuracy. The ablation study demonstrates that the combination of these methods leads to the highest performance across different datasets, highlighting the model’s adaptability and effectiveness in handling diverse keyword spotting tasks. This advanced model, with its robust pre-training methods and fine-tuning capabilities, offers a promising solution for speech recognition applications and can contribute significantly to advancements in the field.

6.3 Comparison with Unsupervised Models (RQ3)

Table 3. Comparison results in accuracy (%)

Model Name	Pre-training Data	Fine-tuning Data	Dev	Eval
Contrastive Predictive Coding (CPC)	Speech Commands	Speech Commands	87.6	86.9
Autoregressive Predictive Coding (APC)	Speech Commands	Speech Commands	87.2	86.5
Masked Predictive Coding (MPC)	Speech Commands	Speech Commands	87.0	86.7
CAB-KWS (full)	Speech Commands	Speech Commands	88.1	88.3
Contrastive Predictive Coding (CPC)	Librispeech-100	Speech Commands	87.8	87.4
Autoregressive Predictive Coding (APC)	Librispeech-100	Speech Commands	87.7	87.5
Masked Predictive Coding (MPC)	Librispeech- 100	Speech Commands	87.9	87.0
CAB-KWS(full)	Librispeech-100	Speech Commands	88.4	88.5

The CAB-KWS model is a sophisticated keyword spotting solution that integrates multiple pre-training techniques to improve classification accuracy in speech recognition tasks. The Table 3 provided presents a comparison of the CAB-KWS model with three other models that employ individual pre-training methods, namely Contrastive Predictive Coding (CPC), Autoregressive Predictive Coding (APC), and Masked Predictive Coding (MPC). By comparing the performance of these models, we can gain insights into the effectiveness of the CAB-KWS model and highlight its advantages over models based on single pre-training techniques.

The comparison in the table reveals that the CAB-KWS model consistently achieves the highest classification accuracy on both the development (Dev) and evaluation (Eval) datasets when fine-tuned on Speech Commands, regardless of the pre-training data source (Speech Commands or Librispeech-100). This result underlines the goodness of the CAB-KWS model as it demonstrates its ability to effectively utilize multiple pre-training techniques to outperform models that rely on individual pre-training methods.

The CAB-KWS model’s superior performance can be attributed to its ability to integrate and capitalize on the strengths of various pre-training techniques. By combining different methods, the model can learn more diverse and robust representations of the data, which in turn leads to improved classification accuracy. This adaptability makes the CAB-KWS model particularly suitable for a wide range of applications in keyword spotting and speech recognition tasks, where performance and generalizability are crucial.

Furthermore, the CAB-KWS model’s consistent performance across different pre-training data sources indicates its flexibility and robustness. It is not limited by the choice of pre-training dataset, which is an essential aspect of its goodness. This characteristic allows the model to be adaptable and versatile, enabling its use in various speech recognition applications with different data sources.

In summary, the CAB-KWS keyword spotting model showcases its goodness by effectively combining multiple pre-training techniques to achieve superior classification accuracy compared to models based on individual pre-training methods. Its consistent performance across different pre-training data sources highlights its adaptability, making it a promising solution for diverse speech recognition tasks. The CAB-KWS model’s ability to harness the strengths of various pre-training techniques and deliver enhanced performance demonstrates its potential to contribute significantly to advancements in the field of speech recognition (Fig. 2).

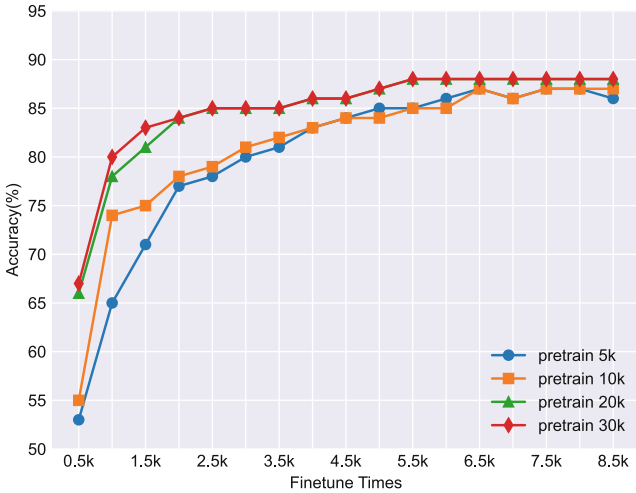


Fig. 2. Comparison of results with different pre-training steps. The number of pre-training steps in unsupervised learning significantly impacts accuracy and fine-tuning convergence. In our experiments, pre-training for 30K steps achieved the best classification accuracy and the quickest convergence.

7 Conclusion

This paper presents a robust approach for the Keyword Spotting (KWS) task. Our CNN-Attention architecture, in combination with our unsupervised contrastive learning method, CABKS, utilizes unlabeled data efficiently. This circumvents the challenge of acquiring ample labeled training data, particularly beneficial when target keywords change or when positive samples are scarce. Furthermore, our speech augmentation strategy enhances the model’s robustness, adapting to variations in keyword utterances. By using contrastive loss within mini-batches, we’ve improved training efficiency and overall performance. Our method outperformed others such as CPC, APC, and MPC in experiments. Future work could explore this approach’s application to other speech tasks and investigate other augmentations or architectures to enhance performance. This work marks a significant step towards more reliable voice-controlled systems and interactive agents.

References

1. Chung, Y.A., Hsu, W.N., Tang, H., Glass, J.: An unsupervised autoregressive model for speech representation learning. arXiv preprint [arXiv:1904.03240](https://arxiv.org/abs/1904.03240) (2019)
2. De Andrade, D.C., Leo, S., Viana, M.L.D.S., Bernkopf, C.: A neural attention model for speech command recognition. arXiv preprint [arXiv:1808.08929](https://arxiv.org/abs/1808.08929) (2018)
3. Garcia, A., Gish, H.: Keyword spotting of arbitrary words using minimal speech resources. In: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 1, pp. I–I. IEEE (2006)
4. Hannun, A., et al.: Deep speech: scaling up end-to-end speech recognition. arXiv preprint [arXiv:1412.5567](https://arxiv.org/abs/1412.5567) (2014)
5. Jaitly, N., Hinton, G.E.: Vocal tract length perturbation (VTLP) improves speech recognition. In: Proc. ICML Workshop on Deep Learning for Audio, Speech and Language, vol. 117, pp. 21 (2013)
6. Jiang, D., et al.: Improving transformer-based speech recognition using unsupervised pre-training. arXiv preprint [arXiv:1910.09932](https://arxiv.org/abs/1910.09932) (2019)
7. Kharitonov, E., et al.: Data augmenting contrastive learning of speech representations in the time domain. In: 2021 IEEE Spoken Language Technology Workshop (SLT), pp. 215–222. IEEE (2021)
8. Ko, T., Peddinti, V., Povey, D., Khudanpur, S.: Audio augmentation for speech recognition. In: Sixteenth Annual Conference of the International Speech Communication Association (2015)
9. Li, B., et al.: Acoustic modeling for google home. In: Interspeech, pp. 399–403 (2017)
10. Li, P., Liang, J., Xu, B.: A novel instance matching based unsupervised keyword spotting system. In: Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007), pp. 550–550. IEEE (2007)
11. Luo, J., Wang, J., Cheng, N., Jiang, G., Xiao, J.: End-to-end silent speech recognition with acoustic sensing. In: 2021 IEEE Spoken Language Technology Workshop (SLT), pp. 606–612. IEEE (2021)
12. Majumdar, S., Ginsburg, B.: MatchboxNet: 1D time-channel separable convolutional neural network architecture for speech commands recognition. arXiv preprint [arXiv:2004.08531](https://arxiv.org/abs/2004.08531) (2020)

13. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint [arXiv:1807.03748](https://arxiv.org/abs/1807.03748) (2018)
14. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an ASR corpus based on public domain audio books. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5206–5210. IEEE (2015)
15. Park, D.S., et al.: SpecAugment: a simple data augmentation method for automatic speech recognition. arXiv preprint [arXiv:1904.08779](https://arxiv.org/abs/1904.08779) (2019)
16. Park, D.S., et al.: Improved noisy student training for automatic speech recognition. arXiv preprint [arXiv:2005.09629](https://arxiv.org/abs/2005.09629) (2020)
17. Park, H.J., Zhu, P., Moreno, I.L., Subrahmanya, N.: Noisy student-teacher training for robust keyword spotting. arXiv preprint [arXiv:2106.01604](https://arxiv.org/abs/2106.01604) (2021)
18. Schalkwyk, J., et al.: “your word is my command”: Google search by voice: a case study. *Adv. Speech Recogn. Mob. Environ. Call Centers Clin.* 61–90 (2010)
19. Tejedor, J., et al.: Search on speech from spoken queries: the multi-domain international albayzin 2018 query-by-example spoken term detection evaluation. *EURASIP J. Audio Speech Music Process.* **2019**(1), 1–29 (2019)
20. Tejedor, J., et al.: Albayzin query-by-example spoken term detection 2016 evaluation. *EURASIP J. Audio Speech Music Process.* **2018**, 1–25 (2018)
21. Varol, G., Momeni, L., Albanie, S., Afouras, T., Zisserman, A.: Scaling up sign spotting through sign language dictionaries. *Int. J. Comput. Vision* **130**(6), 1416–1439 (2022)
22. Vygon, R., Mikhaylovskiy, N.: Learning efficient representations for keyword spotting with triplet loss. In: *Speech and Computer: 23rd International Conference, SPECOM 2021, St. Petersburg, Russia, September 27–30, 2021, Proceedings 23*, pp. 773–785. Springer (2021)
23. Warden, P.: Speech commands: a dataset for limited-vocabulary speech recognition. arXiv preprint [arXiv:1804.03209](https://arxiv.org/abs/1804.03209) (2018)
24. Wei, Y., Gong, Z., Yang, S., Ye, K., Wen, Y.: EdgeCRNN: an edge-computing oriented model of acoustic feature enhancement for keyword spotting. *J. Ambient Intell. Humanized Comput.* 1–11 (2022)
25. Zhang, Y., Glass, J.R.: Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In: 2009 IEEE Workshop on Automatic Speech Recognition & Understanding, pp. 398–403. IEEE (2009)



Deep Learning in Automated Worm Identification and Tracking for *C. Elegans* Mating Behaviour Analysis

Chukwuma Hilary Akpu, Hong Wei^(✉) , and Xia Hong 

University of Reading, Reading RG6 6AY, UK
{h.wei,x.hong}@reading.ac.uk
<https://www.reading.ac.uk/computer-science/>

Abstract. This study is concerned with computer vision technologies applied in *C. elegans* (diminutive nematodes) mating behaviour analysis, more specifically object detection and tracking to find contacts of male and female worms in worm mating videos. Advanced deep learning algorithms, such as YOLOv8 and DeepSORT, are adapted in the automated worm identification and tracking system. A modified DeepSORT algorithm is developed to cope with appearance similarity of *C. elegans* for improving the tracking accuracy. In addition, a male worm detection and tracking algorithm, utilising the male worm's mobility characteristic, assists the modified DeepSORT in accurate male worm tracking. Finally, worm contact detection is implemented by calculating the Euclidean distance between the male and female worms. The developed system, named as M1 and M2, is trained and evaluated under two sets of data, bounding boxes and segmented worms, respectively. Furthermore, we compared the effectiveness of including SAM segmentation optional module in experiments. The evaluation results have shown that YOLOv8 has excellent performance in worm detection to cope with deformable worm shape, and the modified DeepSORT significantly outperforms the default DeepSORT in worm tracking.

Keywords: Object detection and tracking · *C. elegans* mating behaviour analysis · deep learning

1 Introduction

Background. This study is concerned with applying computer vision technologies to discover mating behaviours of *Caenorhabditis elegans* (*C. elegans*) worms. *C. elegans* are diminutive and free-living nematodes that have emerged as vital model organisms across diverse scientific disciplines, including neurobiology, developmental biology, and genetics [19]. In the short developmental life cycle, typically three days, *C. elegans* undergo complete development from an embryo to a sexually mature adult [1]. This significant characteristic of *C. elegans* has made it popular in investigations to unveil correlations between *C.*

C. elegans behaviour and the presence of environmental toxins in the soil [8]. By analysing *C. elegans* mating behaviours, scientists intend to reveal soil conditions with regard to pollution [4]. Carefully setting experiments with a camera to capture *C. elegans* mating behaviours made the analysis possible purely based on videos. However, manual methods to observe the whole process are time-consuming, which restrict the scope and efficiency of comprehensive studies. An automated analysis tool based on computer vision technologies, such as object detection and tracking is sought after to discover the mating behaviours.

In recent years, with the development of computer vision technologies, deep neural networks are adapted in object detection and tracking. The emergence of YOLO [23], from the original YOLO to YOLOv8, enables accurate real-time object detection for robotics, autonomous vehicles, and video monitoring applications [27]. DeepSORT [28] has further advanced multiple object tracking with a focus on simple and effective algorithms, which integrate appearance information to improve the performance of the original simple online and real-time tracking (SORT) algorithm [3]. Alternatively, the Segment Anything Model (SAM) is an attempt to lift image segmentation into the era of foundation models [17].

Problem Statement. The videos taken in the *C. elegans* reproductive experiments contain six worms, ideally, in each frame, five females and one male. The duration of each video is about 18 min. The male worm is active and moving around to approach females, whilst female worms are relatively inactive but with motions. Figure 1(a) shows six worms in a frame and Fig. 1(d) demonstrates the male worm has contacted a female worm after a few minutes in the video. From Fig. 1, it can be seen that positions and shapes of these worms are changing in the course of the process. This leads to a problem of deformable object detection and tracking. The active male worm may move out of the scene (see Fig. 1(b)), and imprints may appear in the scene (see Fig. 1(b), (c), and (d)). These challenge the traditional approaches, e.g. appearance based object detection, motion based tracking, etc. [5–7, 21].

With the final aim of automated analysis of *C. elegans* mating behaviours, the initial objective of this study is to detect and record the contacting points between the male and a female worm in the 18 min videos so that researchers can skip to these points to observe the mating behaviours. Deep learning methods,

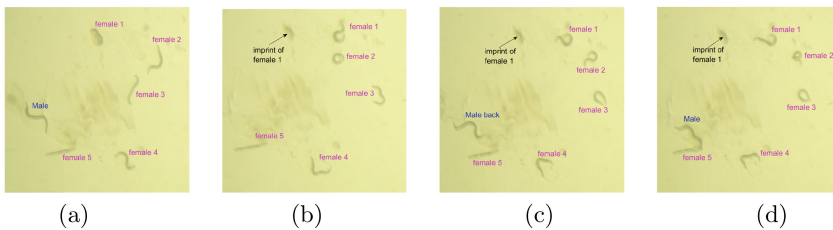


Fig. 1. Examples of *C. elegans* positions in experiments (a) Initial positions, (b) Male left the scene, (c) Male back to the scene, (d) Male touching female 5

based on both YOLOv8 and DeepSORT, are considered in object detecting and tracking to investigate how the aforementioned challenges can be dealt with by advanced technologies.

Related Work. Using a computer to monitor nematodes movement was attempted in the 80's of the last century [9] with a 6809 microprocessor programmed in assembly language under a lighting condition of high contrast. In the area of *C. elegans* behavioural research, an array of innovative tracking systems and methodologies have emerged, each with distinct attributes and capabilities. Ramot, et al. [22] developed the Parallel Worm Tracker, a platform for quantitative analysis of *C. elegans* locomotion. This system is capable of tracking multiple worms in sequential video frames and recording their centroid positions. It is also adept at calculating the worm's speed and angular velocity. Simonetta, et al. [26] proposed an automated system that tracks the locomotor activity of *C. elegans* which is also suitable for circadian locomotion recording and research on aging mechanisms. The system utilizes light microbeams to detect worm movement and convert the frequency of the signal, allowing for a sophisticated analysis of locomotion patterns. Jaensch, et al. [13] presented an automated tracking and analysis system that offers exceptional accuracy in quantifying and tracking the size of Green Fluorescent Protein (GFP)-labelled centrosomes in early *C. elegans* embryos. It proves its efficiency by effectively processing large datasets with only minor manual corrections required. Dzyubackyk, et al. [10] introduced an algorithm designed for tracking *C. elegans* embryogenesis utilizing fluorescence microscopy images. The algorithm demonstrated successful segmentation and tracking of nuclei in the image sequence, surpassing the performance of existing methods. It was found to be efficient and user-friendly, employing graph-cut-based energy minimization for improved results. Restif, et al. [24] introduced CeleST, a sophisticated computer vision software tailored for automated tracking and in-depth analysis of *C. elegans* swimming behaviour. This innovative approach employs adaptive background subtraction to effectively discern and track individual nematodes within video frames, surmounting the challenges posed by intricate and diverse background environments. Despite its impressive capabilities, it is noteworthy that CeleST is designed to exclude worms that are in contact during the tracking process. This unique exclusion, while advantageous in some contexts, may present limitations when applied to specific studies that aim to explore the nuances of worm interactions and contact behaviour.

Javer, et al. [14], introduced the Tierpsy Tracker, a Python-based multi-worm tracker that extracts postural information from worm behaviour videos. By offering enhanced head-tail detection and locally calculated thresholds, the system provides an improved tracking accuracy. Lorimer, et al. [20] developed an approach that excels in detecting changes in worm locomotion behaviour through prediction error analysis. The algorithm localizes changes in worm locomotion behaviour and offers flexibility and sensitivity. Leonard and Vidal-Gadea [20] proposed a cost-efficient and user-friendly *C. elegans* tracking system. Although designed for classroom usage, the system delivers results almost rivalling more

expensive professional systems, making it a cost-effective option for basic worm behaviour studies. Deep learning methods were employed by Banerjee, et al. [2] in their deep-worm-tracker for accurate detection and tracking *C. elegans* in worm behavioural studies.

Contributions. Different from the existing tracking systems, the uniqueness of this study lies in detecting and monitoring *C. elegans* pairs engaged in mating interactions. This requires identifying individual worms and tracking them throughout their movements with position and appearance changes until the male worm touches one of the female worms. Limitations of robustness and reliability are still issues in dealing with occlusion and object lost/back in the scene by using traditional approaches. Curiously, this study investigates how an integrated deep learning approach copes with the limitations. In this paper, we introduce an automated system that aids the study of *C. elegans* mating behaviours analysis. The developed system, adapting YOLOv8 and modified DeepSORT, addresses the challenging task of detecting and tracking individual deformable worms, monitoring the trajectory of the male worm, and identifying mating occurrences in recorded videos. A contact detection mechanism is implemented efficiently reporting instances where worms overlap or make contact. The system’s overall performance achieves a substantial level of accuracy in worm tracking and contact detection. The technical innovation of the integrated system is demonstrated in developing algorithms to modify DeepSORT, which significantly improves the performance in worm tracking and contact detection.

The rest of the paper is organised as below. Section 2 briefly describes YOLOv8 and DeepSORT. Section 3 introduces our technical approaches, including the system framework, data annotation, and details of implementation. Relevant experiments and testing results are demonstrated in Sect. 4. Section 5 concludes the study and lays the future work.

2 Preliminary: YOLOv8 and DeepSORT

YOLOv8: YOLO was introduced by Redmon, et al. [23], representing a significant leap forward in object detection efficiency and effectiveness. The key innovation of YOLO lies in its ability to perform object detection in a single forward pass of various deep neural networks as backbones for different versions, from the original YOLO to recent YOLOv8, thereby achieving real-time processing speed. The network divides the input image into a grid, and each grid cell is responsible for predicting bounding boxes (objects). This grid-based approach significantly reduces the computational overhead, making YOLO highly efficient and suitable for real-time applications.

Essentially, the object detection task is framed as a regression problem, where the neural network predicts bounding boxes and class probabilities directly from the input image. Non-Maximum Suppression (NMS) is then used, which is a post-processing technique [12] to reduce the number of overlapping bounding boxes and improve the overall detection quality. From the original YOLO to recent

YOLOv8, the algorithm also introduces anchor boxes to improve the accuracy of object detection. Anchor boxes are predetermined shapes of different aspect ratios that are used to refine the predicted bounding boxes. By introducing anchor boxes, recent versions, such as YOLOv8, are better equipped to handle objects of various shapes and sizes leading to improved localization accuracy.

As a state-of-the-art model, YOLOv8 [15] by Ultralytics was used in this study. Object detection algorithms of YOLOv8 generate multiple bounding boxes around the same object with different confidence scores, followed by NMS which filters out redundant and irrelevant bounding boxes, keeping only the most accurate ones. From input images, the YOLOv8 efficiently outputs a set of detected bounding boxes together with their sufficiently high confidence scores of object detection.

DeepSORT: DeepSort stands for Deep Learning-based SORT (Simple Online and Realtime Tracking), is an advanced object tracking algorithm that combines the principles of deep learning and traditional SORT to achieve highly accurate and efficient tracking results in real-time video sequences. Extended from SORT, a traditional online tracking method that uses Kalman filtering [16] and the Hungarian algorithm [18] for data associations, DeepSORT addresses the limitation of the SORT algorithm by incorporating two pieces of information, motion and appearance into its framework [28], named as $d^{(1)}$ and $d^{(2)}$, respectively.

The motion metric $d^{(1)}$ is implemented with the Mahalanobis distance between predicted Kalman states vector \mathbf{d}_j of j th bounding boxes (consisting of its center position (u, v) , aspect ratio r , height h as well as their respective velocities in image coordinates) and newly arrived measurements \mathbf{y}_i of i th track.

$$d^{(1)}(i, j) = (\mathbf{d}_j - \mathbf{y}_i)^T \mathbf{S}_i^{-1} (\mathbf{d}_j - \mathbf{y}_i) \quad (1)$$

where the i -th track distribution is projected into the measurement space by $(\mathbf{y}_i, \mathbf{S}_i)$.

For each bounding box detection \mathbf{d}_j , an appearance descriptor \mathbf{a}_j with $\|\mathbf{a}_j\| = 1$, is calculated based on a pre-trained convolution neural network(CNN) [28]. This approach trains appearance features offline with a large number of training samples on a convolution neural network. The appearance metric $d^{(2)}$ measures the smallest cosine distance between the i -track and j -th detection in the image space.

$$d^{(2)}(i, j) = \min\{1 - \mathbf{a}_j^T \mathbf{a}_k^{(i)} | \mathbf{a}_k^{(i)} \in \mathfrak{R}_i\} \quad (2)$$

where \mathbf{a}_j denotes an appearance descriptor in j -th detection, and $\mathbf{a}_k^{(i)}$ represents appearance descriptors in the i -th image space \mathfrak{R}_i , which is maintained as the recent 100 associated appearance descriptors for each track.

Finally, the two metrics are combined as:

$$C_{i,j} = \lambda d^{(1)}(i, j) + (1 - \lambda) d^{(2)}(i, j) \quad (3)$$

where $0 \leq \lambda \leq 1$ is a hyperparameter. $C_{i,j}$ are used to determine if the detected bounding boxes are assigned to each track. For the details of DeepSORT implementation, refer to [28].

3 Methodology

Relevant system frameworks and algorithms are developed to overcome challenges that arise when dealing with visually similar entities such as *C. elegans* for their mating behaviours analysis based on videos. Figure 2 depicts the system framework. YOLOv8, fine-trained by annotated *C. elegans* data, is for *C. elegans* detection. It automatically resizes and rescales the input image to match that of the images used for training the detector. The locations of objects detected in the input image are returned as a set of bounding boxes. A modified DeepSORT algorithm is implemented for multiple worm tracking. In the framework, SAM (Segment Anything Model) is optionally attempted to assist DeepSORT in accurate tracking with binary images [17]. For the technical details of SAM, which is beyond the scope of this paper, the readers are referred to [17]. In Sect. 4, we carried out an ablation study of SAM to obtain empirical results with or without SAM module in Fig. 2. The male worm is identified and tracked in each frame, even when it leaves and comes back to the scene. The system outputs two text files with information on the contact times of the male worm and a female worm, and coordinates of the male worm’s trajectory, as well as videos with detected worms in each frame.

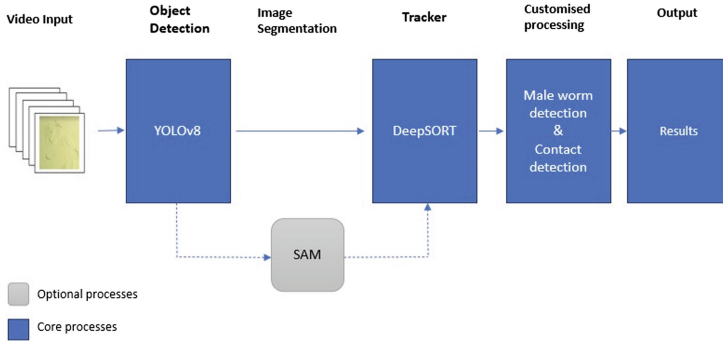


Fig. 2. Framework of the worm detection and tracking system.

3.1 Data Annotation

To fine-train YOLOv8, annotated training samples are prepared from the *C. elegans* mating videos. This endeavour is expedited through the utilization of the RoboFlow platform [25], which eased the annotation and data augmentation process. Two different versions of annotated training samples are collected, bounding boxes (version 1: V1) and segmented encapsulations (version 2: V2), as shown in Fig. 3 (a) and (b), respectively. The initial step involves the conversion of videos into images. This task was achieved through the utilization of the

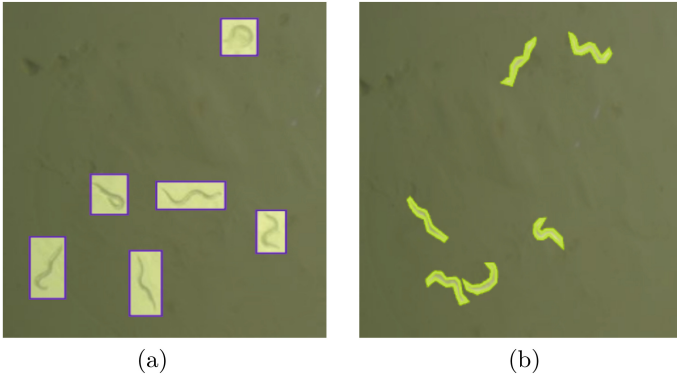


Fig. 3. Training samples for YOLO fine-train (a) Bounding box, (b) Segmented

ffmpeg library [11], where images or frames are meticulously extracted from the videos at intervals, typically spanning every 5 to 10 s.

In the selection of training samples, diversity is the key element to ensure the robustness of a trained model. V1 contains 846 training samples, whilst V2 only has 208 training samples due to the process being highly time-consuming.

3.2 Modification of DeepSORT

A modified DeepSORT algorithm was proposed in this study to enhance the tracking accuracy. Due to high visual similarities between the worms, identity switching often occurs when these worms come in contact with each other or occlude themselves. This, in turn, affects tracking accuracy. In other instances, it would create a new ID for an existing worm. Knowing that videos in this study consist of 6 worms at maximum, we modified DeepSORT to limit the frequency in which new worm IDs are created after all 6 worms in the video have been identified and are being tracked.

To mitigate identity switches during worm interactions, where visual similarities can lead to confusion, we introduced dynamism to the number of frames for new track identification. The DeepSORT algorithm initiates a new track hypothesis for each detection that cannot be associated with an existing track. These nascent tracks are initially classified as tentative with the algorithm anticipating their consecutive appearance in a specified number of frames before their inclusion in the track set. Failure to meet this criterion results in discarding these tentative tracks. This predefined number for track identification can be adjusted prior to the program's execution. In modifying DeepSORT, we made the predefined number for track identification adjustable during program execution. At the program's onset, this number is set to a relatively low value (e.g., 10 frames) and once all six worms in the video are integrated into DeepSORT's track set, we dynamically increase the number of frames for new track identification to a substantial value (e.g., 400 frames). This means that after all 6 worms are

identified if the YOLO model detects a *7th* object (maybe a worm imprint as shown in Fig. 1), this *7th* object (wrongly detected) will not be easily included in our modified DeepSORT's track set as opposed to the default DeepSORT. This dynamic adjustment significantly enhances tracking accuracy by minimizing the occurrence of identity switches.

SAM (Segment Anything Model) [17] is explored aiming to improve DeepSORT tracking accuracy. Although experiments with SAM had better tracking performance, the time taken to run such experiments is double that of the experiments without SAM (see Table 2). For this reason, we left SAM to be an optional process, for instances where speed is prioritized. YOLOv8 outputs bounding boxes after *C. elegans* detection. SAM is applied to each bounding box to segment *C. elegans* within it. Using this way, the segmented image and the bounding boxes are passed into the DeepSORT tracker. Figure 4 depicts the outputs of YOLO and SAM in the system.

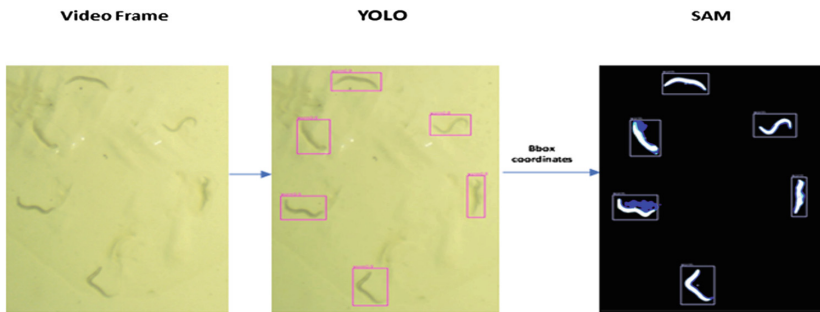


Fig. 4. Outputs of YOLO and SAM in the system

3.3 Male Worm Detection and Contact Detection

An integral objective of this worm detection and tracking system is to provide scientists with valuable insights into the moments when contact occurs between male and female worms. The visual similarity between male and female worms may cause problems for the appearance metric used in DeepSORT. A male worm detection algorithm is developed and implemented. As aforementioned, the male worm tends to have higher mobility and move more rapidly compared to those female worms in the videos taken from *C. elegans* mating events. This behavioural disparity presents a unique opportunity to leverage mobility as a discriminating factor for distinguishing between male and female worms. The approach for male worm detection is presented in Algorithm 1.

For contact detection, the initial way involves calculating the Euclidean distance between the centre points of two bounding boxes of worm pairs (one is the male worm). If this distance falls below a specified threshold, it unequivocally

Algorithm 1. Male Worm Identification by Mobility

Require: Coordinates of bounding boxes in each frame. Total number of bounding boxes m . A specified frame number n .

Ensure: The male worm coordinates are identified.

1. For bounding boxes $k = 1, \dots, m$ in a frame of the video
 - (a) Obtain the centre coordinates $u_k^{(i)}$ and $v_k^{(i)}$ in the initial frame.
 - (b) At a specified frame number (n):
 - i. Obtain the current centre coordinates $u_k^{(c)}$ and $v_k^{(c)}$.
 - ii. Compute the Euclidean distance D_k between the current and initial coordinates.
- $$D_k = \sqrt{(u_k^{(c)} - u_k^{(i)})^2 + (v_k^{(c)} - v_k^{(i)})^2} \quad (4)$$
2. At frame ($n+1$), identify the worm with the highest Euclidean distance $\max\{D_k\}$ as the male worm.
 3. Return
-

signals contact between the worms. The time of contact in the video was determined by multiplying the frame number and the frames per second of the video. While this method provides a quick, straightforward, and intuitive way for contact detection, it also demonstrates certain limitations. In scenarios where two worms are positioned in parallel, their bounding box centre points may exhibit proximity, leading to potential false positives in the contact detection process. To address the limitations, worm segments as connected components are attempted in contact detection. If any segmentation point from the male worm overlaps or is in proximity to any female segmentation point, it is indicative of contact between the two worms, and the time is recorded for the occurrence. The second method highly depends on the accurate segmentation of each worm in the frame. Otherwise, it may introduce contact detection errors.

4 Evaluation and Results

Evaluation takes place in three folders, *i.e.* worm detection, tracking, and contact detection. A set of metrics is adopted. Mean Average Precision (mAP) and $mAP(50-95)$ are for worm detection, whilst the Multiple Object Tracking Accuracy ($MOTA$) is used for evaluating worm tracking accuracy. The contact detection is done by comparing the system output with the ground truth manually obtained, and contact accuracy and contact $F1$ score are used as metrics.

4.1 Evaluation Metrics

Mean Average Precision (mAP) is a commonly used evaluation metric in object detection tasks. It is defined in Eq. 5

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (5)$$

where AP_i is the precision for detection class i , defined as $AP_i = \frac{TP_i}{TP_i + FP_i}$. N is the number of detection classes, in this case, six classes represent six worms in a scene. A higher mAP score indicates better performance in object detection, implying that the detection model excels in both precision and recall, where $recall_i = \frac{TP_i}{TP_i + FN_i}$. TP , FP , and FN stands for true positive, false positive, and false negative, respectively.

$mAP(50-95)$ is an extension of the mAP metric, taking Intersection over Union (IoU) in the calculation. IoU is defined as:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (6)$$

$mAP(50-95)$ measures the mean Average Precision over a range of IoU thresholds, typically from 0.5 to 0.95, in increments of 0.05. The following steps are involved in the calculation.

1. For each detection class and for each IoU threshold (beginning from 0.5, incrementing by 0.05 up to 0.95), we compute the AP .
2. The $mAP(50-95)$ score is then obtained by averaging these AP values across all detection classes and across all IoU thresholds (from 0.5 to 0.95).

A high $mAP(50-95)$ score indicates that the model performs well in object detection across various IoU thresholds. This metric is more reliable than the mAP which only uses an IoU value of 0.5, thus indicating that a higher $mAP(50-95)$ score will result in more accurate bounding box predictions.

Multiple Object Tracking Accuracy ($MOTA$) is an evaluation metric in the field of object tracking that provides a holistic assessment of a tracking system's ability to accurately track multiple objects over time. It is defined as:

$$MOTA = 1 - (FN + FP + M_{sw})/GT \quad (7)$$

where

- False Negatives (FN): quantifies the number of worms present in the frame which the tracking system fails to detect.
- False Positives (FP): encompasses objects erroneously identified as worms by the tracker, quantifying instances of incorrect worm detection.

- Male worm switches (M_{sw}): indicates whether the male worm is detected or switched with other worms in a given frame. This parameter carries a weight of 0.25 greater (+25%) than that of identity switches occurring within female worms.
- Ground Truth (GT): represents the actual number of worms present in the frame.

A high *MOTA* score suggests that the tracking system is performing well in terms of accuracy in object tracking. This implies that it effectively tracks the majority of objects while minimizing missed detections (FN), false alarms (FP), and failure in detecting a male worm (M_{sw}) in a frame.

Contact Accuracy in contact detection is defined as

$$\text{Contact accuracy} = \frac{\text{contacts detected}}{\text{overall contacts}} \quad (8)$$

In Eq. 8, contacts detected refers to the number of contacts (between male and female worms) that were detected by the system in the testing video, and overall contacts is derived from the ground truth showing the real contact number.

Contact F1 Score is expressed in Eq. 9.

$$\text{Contact } F1 \text{ score} = \frac{2TP}{2TP + FP + FN} \quad (9)$$

where TP refers to contacts detected as contacts, FP ; non-contacts detected as contacts, and FN , contacts that were not detected by the system.

4.2 System Evaluation

Evaluation experiments were conducted by using a system with a 7th-generation Intel processor, 16GB RAM, and a 6GB Nvidia Rtx2060 GPU. A pre-trained YOLOv8n (a nano-sized model) was adapted in the experiments. The fine-tuning of the model utilised the two sets of training samples, V1 and V2, described in Sect. 3.1. The learning rate was set as 0.01 and the default Adam optimizer is used in training. Two YOLOv8 models, M1 and M2, are established with V1 and V2, respectively. The training took less than an hour for M1 and over 4 h for M2. Figure 5 shows the losses against epochs in their training. In M1, box losses and class losses were involved in the training, whilst for M2, segmentation losses were also taken into account. Testing was conducted on both M1 and M2.

Worm Detection Evaluation. In the evaluation experiments, we had three sets of testing data in each model. For Model M1, associated with the V1 dataset, 133 samples were derived along with the training samples from the three videos as described in Sect. 3.1; 167 samples were annotated from a new set of videos,

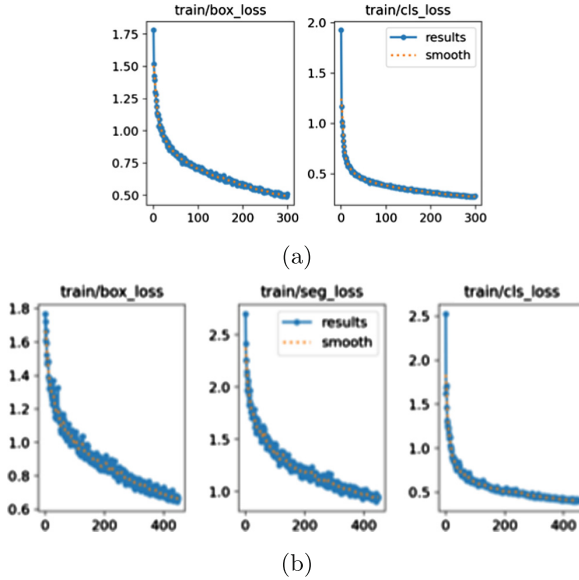


Fig. 5. Losses against epochs in training (a) Model M1, (b) Model M2

and then we applied data augmentation to increase the testing dataset to 298 samples. For Model M2, associated with dataset V2, 46 testing samples were derived along with the training samples; 48 samples from a new set of videos, and 71 samples from data augmentation. As demonstrated in Table 1, M2 shows the consistency of mAP and $mAP(50-95)$ scores for the three testing datasets. Interestingly for M1, the testing dataset derived from the same videos as the training samples produces a much better $mAP(50-95)$ score than those from the testing datasets extracted from different videos. This may be caused by bounding boxes with larger percentage overlaps decreasing in these two testing datasets. Meanwhile, M1 yields superior mAP scores. The performance divergence can be attributed to several factors, mainly because of the dissimilarity in training data. It is worth noting that both the V1 and V2 training datasets comprise images extracted from three distinct *C. elegans* mating videos. However, a pivotal distinction emerges in the volume of training samples. The V1 dataset boasts a larger training sample size, amounting to 846 samples, in contrast to the V2 dataset, which comprises a modest 208 samples. On the one hand, it may indeed imply an enhancement in model performance, as a larger training dataset generally fosters better learning outcomes. Conversely, this scenario can also potentially lead to overfitting, where the model becomes excessively tailored to the idiosyncrasies of the three specific videos present in the V1 dataset. It underscores the significance of dataset diversity in achieving robust model generalization.

Table 1. Worm detection performance

Model	Training sample	Epochs in training	Batch size	Testing sample	mAP	mAP (50–95)
M1	846	300	16	133	0.992	0.858
				167	0.981	0.591
				298	0.980	0.587
M2	208	450	1	46	0.975	0.765
				48	0.960	0.714
				71	0.950	0.713

Worm Tracking Evaluation. Building up on the YOLOv8 models is the DeepSORT for worm tracking with or without optional modular SAM. Ablation experiments were conducted with three different scenarios in worm tracking evaluation.

1. A system that uses the default DeepSORT algorithm.
2. A system that uses the modified DeepSORT algorithm.
3. A system that incorporates the Segment Anything Model (SAM) in conjunction with the modified DeepSORT algorithm.

Overall, 113,400 frames were involved in the worm tracking and contact evaluation, about 10.5 fps (frame per second). The results presented in Tables 2 and 3 were from a test of an eighteen-minute video. With different algorithms applied, the testing time was different. From Table 2, it is observed that with default DeepSORT, the M2 model, trained on a dataset comprising 208 images has the lowest *MOTA*. However, it attains the most commendable performance when coupled with the modified DeepSORT tracker and SAM. Table 2 has clearly shown that in both M1 and M2, the modified DeepSORT outperforms the default DeepSORT significantly. The modified DeepSORT with SAM incorporated gives better *MOTA* scores, but the tracking operation took a longer time.

Table 2. Worm tracking performance

Model	Tracker	<i>MOTA</i>	M_{sw}	Testing time
M1	Default DeepSORT	61.5%	9	50:16
	Modified DeepSORT	82.2%	4	51:06
	Modified DeepSORT (SAM)	85.2%	3	2:12:45
M2	Default DeepSORT	21.4%	30	51:44
	Modified DeepSORT	97.6%	4	56:23
	Modified DeepSORT (SAM)	97.9%	4	2:23:45

Worm Contact Evaluation. Detecting contact between male and female worms was evaluated by implementing a manual recording process to establish the ground truth. It documents the time frames encompassing the male worm’s interaction with a female worm, including the initiation and cessation of contact. A video of *C. elegans* mating behaviours was involved in the testing. Contact accuracy and F1 score defined in Eqs. 8 and 9 are demonstrated in Table 3. Due to the low performance in tracking, the default DeepSORT algorithm was excluded from the tests for both M1 and M2, only modified variants are evaluated for contact detection.

Table 3. Contact detection performance

Model	Tracker	Contact accuracy	Contact <i>F1</i> score
M1	Modified DeepSORT	91.7%	0.861
	Modified DeepSORT (SAM)	93.7%	0.898
M2	Modified DeepSORT	85.3%	0.738
	Modified DeepSORT (SAM)	73.4%	0.347

The M1 model uses bounding boxes for contact detection since it does not have segmentation coordinates while the M2 model uses the segmentation-based contact detection. Although the M2 model coupled with Modified DeepSORT and SAM has shown the highest performance in worm tracking, it exhibits an anomaly, which affects the contact detection accuracy. The reason for this misclassification is due to how image segmentation was implemented, which renders the worms in stark white against a pitch-black background. Thus, when the worms, displayed as white entities, converge, their contours are not clearly visible due to the overlapping white pixels, making it difficult to identify individual worms in contact. Although worm segmentation has shown an advantage in tracking isolated worms, it occasionally makes mistakes in detecting worms during contact. We observe that the bounding-box-based contact detection (with SAM) performs better than the segmentation-based methods. Interestingly, the M1 model with modified DeepSORT tracker and SAM produced the best results with respect to both contact accuracy and contact *F1* score on the video used in evaluation.

5 Conclusion and Future Work

In this study, we developed a computer vision system which works on worm mating videos to identify and track each worm involved, leading to worm contact detection. By adapting YOLOv8 and DeepSORT algorithms in the system, the modified tracker associated with the developed male worm detection and contact detection algorithms has achieved commendable contact detection accuracy on

the evaluated video recording worm mating behaviours. The research problem involves deformable object detection and tracking. It has proved that the integrated deep learning approach can cope with the difficulty. The evaluation results have shown that YOLOv8 has excellent performance in worm detection to cope with deformable worm shapes in both M1 and M2. By incorporating SAM in the tracker, excellent tracking performance was achieved in M2 although substantial time is required in the operation, and the contact detection accuracy is also improved in M1. Future study includes two aspects, investigating a larger number of small sequences to gain more comprehensive views on mating behaviours and to investigate the role of segmented objects in contact detection. With the two approaches, it is expected to make the system more robust and efficient in coping with different scenarios.

Acknowledgements. The project was initiated with the collaboration supported by the grant of the University of Reading NERC Discipline Hopping for Environmental Solutions. We would like to express our thanks to Nandini Vasudvan, Zuowei Wang, Eva Kevei, and Xingchen Zhai for providing the data and discussions over the project.

References

1. Altun, Z.F., Hall, D.H.: Introduction to *C. elegans*. In: WormAtlas (2009). <https://doi.org/10.3908/wormatlas.1.1>
2. Banerjee, S., Khan, K., Sharma, R.: Deep-worm-tracker: deep learning methods for accurate detection and tracking for behavioral studies in *C. elegans*. *Appl. Anim. Behav. Sci.* **266**, 106024 (2023)
3. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468 (2016)
4. Bhagat, J., Nishimura, N., Shimada, Y.: Worming into a robust model to unravel the micro/nanoplastic toxicity in soil: a review on *Caenorhabditis elegans*. *TrAC, Trends Anal. Chem.* **138**, 116235 (2021)
5. Bradley, D., Roth, G.: Adapting thresholding using the integral image. *J. Graph. GPU Game Tools* **12**(2), 13–21 (2007)
6. Chauhan, A., Krishan, P., Kumar, D.: Moving object tracking using gaussian mixture model and optical flow. *Int. J. Adv. Res. Comput. Sci. Software Eng.* **3**(4), 243–246 (2013)
7. Chen, X., Wang, X., Xuan, J.: Tracking multiple moving objects using unscented kalman filtering techniques. In: International Conference on Engineering and Applied Science (ICEAS 2012) (2012)
8. Donkin, S.G., Dusenbery, D.B.: A soil toxicity test using the nematode *Caenorhabditis elegans* and an effective method of recovery. *Arch. Environ. Contam. Toxicol.* **25**, 145–151 (1993)
9. Dusenbery, D.B.: Using a microcomputer and video camera to simultaneously track 25 animals. *Comput. Biol. Med.* **15**(4), 169–175 (1985)
10. Dzyubachyk, O., Jelier, R., Lehner, B., Niessen, W., Meijering, E.: Model-based approach for tracking embryogenesis in *Caenorhabditis elegans* fluorescence microscopy data. In: Proceeding of Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5356–5359 (2009)

11. FFmpeg-Developers: FFmpeg tool (version be1d324) [software] (2016). <http://ffmpeg.org/>. Accessed 28 June 2024
12. Hosang, J., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6469–6477 (2017)
13. Jaensch, S., Decker, M., Hyman, A.A., Myers, E.W.: Automated tracking and analysis of centrosomes in early *Caenorhabditis elegans* embryos. *Bioinformatics* **26**(12), i13–i20 (2010)
14. Javer, A., et al.: An open-source platform for analyzing and sharing worm-behavior data. *Nat. Methods* **15**, 645–646 (2018)
15. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics yolov8, [software] (2023). <https://github.com/ultralytics/ultralytics/>. Accessed 28 June 2024
16. Karavasilis, V., Nikou, C., Likas, A.: Visual tracking by adaptive kalman filtering and mean shift. In: Konstantopoulos, S., Perantonis, S., Karkaletsis, V., Spyropoulos, C., Vouros, G. (eds.) *Artificial Intelligence: Theories, Models and Applications*, pp. 153–162 (2010)
17. Kirillov, A., et al.: Segment anything. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026 (2023)
18. Kuhn, H.: The hungarian method for the assignment problem. *Naval Res. Logistics Quarterly* **2**, 83–97 (1955)
19. Leung, M.C., et al.: *Caenorhabditis elegans*: an emerging model in biomedical and environmental toxicology. *Toxicological Sci.* **106**(1), 5–28 (2008)
20. Lorimer, T., et al.: Tracking changes in behavioral dynamics using prediction error. *PLoS One* **16**(5), e0251053 (2021)
21. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybernetics* **9**(1), 62–66 (1979)
22. Ramot, D., Johnson, B.E., Berry, B.J., Carnell, L., Goodman, M.B.: The parallel worm tracker: a platform for measuring average speed and drug-induced paralysis in nematodes. *PLoS One* **3**(5), e2208 (2008)
23. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788 (2016)
24. Restif, C., Ibáñez Ventoso, C., Vora, M.M., Guo, S., Metaxas, D., Driscoll, M.: CeleST: computer vision software for quantitative analysis of *C. elegans* swim behavior reveals novel features of locomotion. *PLoS One* **10**(7), e1003702 (2014)
25. Roboflow: everything you need to build and deploy computer vision models (2023). <https://roboflow.com/> Accessed 28 June 2024
26. Simonetta, S.H., Golombek, D.A.: An automated tracking system for *Caenorhabditis elegans* locomotor behavior and circadian studies application. *J. Neurosci. Methods* **161**, 273–280 (2007)
27. Terven, J., Córdova-Esparza, D.M., Romero-González, J.A.: A comprehensive review of YOLO architectures in computer vision: from yolov1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **5**(4), 1680–1716 (2023)
28. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645–3649. IEEE (2017)



Interactive-Time Text-Guided Editing of 3D Face

Yeon-Jeong Lee¹, Yeong-Hun Song¹, Sang Wook Yoo²,
and Joon-Kyung Seong^{1,3}

¹ Department of Artificial Intelligence, College of Informatics, Korea University,
Seoul 02841, South Korea

{bbcdgg1920, syh6087, jkseong}@korea.ac.kr

² LULULAB, Seoul 06054, South Korea

sangwook.yoo@lulu-lab.com

³ School of Biomedical Engineering, College of Health Science, Korea University,
Seoul 02841, South Korea

Abstract. Manipulating 3D faces using text is an important technology in the entertainment industry. However, text-based manipulation of 3D faces remains a challenging area due to the scarcity of data pairs consisting of 3D faces and corresponding text. Additionally, inference for manipulating 3D faces using text prompts often requires several minutes due to the large model sizes or the optimization process to fit the text prompt. In this paper, we propose the ITFaceEdit model, a text and image-based 3D face manipulation model. ITFaceEdit constructs a framework trainable only with image and text data pairs, allowing it to learn a direct relationship between the text latent space and the 3D face latent space. By utilizing vectors from the learned text embeddings, we can manipulate 3D faces, employing face parsing for disentangled manipulation. Through this approach, we not only extend the reconstructed 3D face space using images with text-based manipulation but also configure an inference process without relatively heavy model structures and optimization steps, enabling 3D face manipulation in a few seconds. We demonstrate the superiority of our proposed method through comparisons with existing methods in various ways.

Keywords: 3D face manipulation · Text-driven 3D face manipulation · Text-driven 3D face animation

1 Introduction

Manipulating 3D faces is crucial in various industries such as film, gaming, and beauty. Since the emergence of CLIP [27], many studies have attempted to generate or manipulate facial images or 3D faces from text using the joint embedding

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78122-3_9.

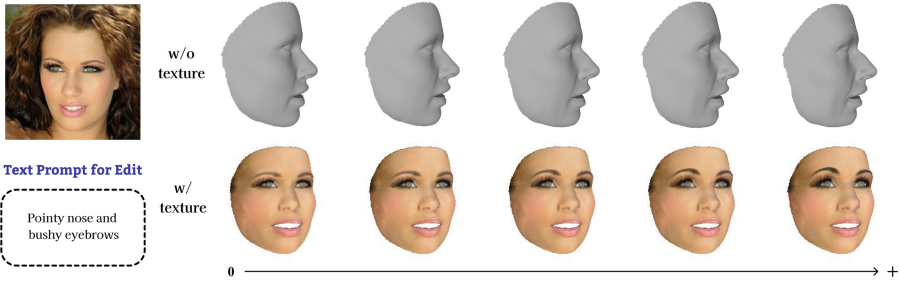


Fig. 1. Text-based 3D face manipulation with ITFaceEdit. The ITFaceEdit model identifies the direction of 3DMM coefficient manipulation from the given text and image, then manipulates the reconstructed 3D face from the image in that direction. ITFaceEdit allows for manipulation of the identity, expression, and texture of the face individually, and this process can be executed within seconds.

space learned by CLIP. However, text-guided manipulation of 3D faces remains challenging due to the scarcity of paired 3D face-text data and the difficulty in geometric manipulation of 3D faces.

Previous research has utilized image-text data to supplement the lack of 3D face-text data pairs for training 3D face manipulation models. However, using facial images alone makes it easy to distinguish differences in expressions and colors, but difficult to learn geometric details such as eyes, nose, and mouth due to the loss of geometric information. Therefore, previous studies have focused on changing expressions and colors while maintaining facial identity. Additionally, many models in text-based 3D face manipulation rely on outputs from StyleGAN [17]-base and Diffusion [13]-based generative models and optimization processes using text prompts, resulting in manipulation processes taking several minutes or more.

To address these challenges, we propose a new model, ITFaceEdit (Fig. 1). We utilized the Deep3DFaceRecon [7] model to extract coefficients for Basel Face Model (BFM) [10]-based 3D face reconstruction from images, thereby incorporating 3D geometric information that extends beyond the capabilities of facial images alone. This approach facilitates the manipulation of identity aspects, which were previously difficult to manage. Additionally, to achieve rapid manipulation, we trained a network that directly maps text to the BFM coefficient space, circumventing the need for optimization techniques. This network effectively maps the specific parts of the coefficients associated with the text prompts. During training, we segmented each text into identity, expression, and texture components, and independently mapped these to the respective coefficients. This method enables swift text-based 3D face manipulation, thereby enhancing both the efficiency and diversity of manipulations. Furthermore, we employed texture maps obtained through differential rendering and facial parsers to achieve higher quality and more visually realistic manipulation results.

We use datasets such as Multi-Modal CelebA-HQ [15, 19, 23], AffectNet [25], along with Cleaned Face Datasets [15, 19, 23] and BFM coefficients obtained from the Deep3DFaceRecon [7] model, for training and evaluation. Our model can manipulate 3D faces reconstructed from various images based on values mapped to BFM coefficients from text input, and can handle manipulation for entire sentences, not just single words. In summary, the contributions of our paper are as follows:

- We propose a new model, ITFaceEdit, structured as a single pipeline (see Fig. 3). This model takes image and text prompts as inputs and effectively manipulates 3D faces with simplicity. Specifically, the Text-BFM Mapper within our model directly maps text features to the 3D coefficient space, offering a new method distinct from prior optimization-based training. We demonstrate in Table 3 that this approach can significantly reduce the time for 3D face manipulation to about 10s and enables simultaneous manipulation of multiple 3D faces with the same text.
- Unlike many studies that target expression or texture for manipulation, our model divides the Text-BFM Mapper into identity, expression, and texture components for training, allowing for detailed manipulation that includes facial identity. Moreover, unlike previous methods that required adjusting manipulation intensity through iterations of optimization-based training, our model provides three parameters that independently control the identity, expression, and texture aspects of the 3D face. These parameters enable users to manipulate and animate faces in real-time with the desired intensity.
- Lastly, we applied texture maps and localized manipulation using parsers in ITFaceEdit, enabling more visually realistic and natural manipulations. We demonstrate the effectiveness of our approach using FID and CLIP scores in Table 1, as well as visual evidence in Table 2.

2 Related Work

2.1 Parameter Space of 3D Faces

The development of 3D Morphable Models(3DMM) has been driven by significant research contributions over the years. [4] introduced the foundational concept by leveraging Principal Component Analysis (PCA) [1] to model facial shape and texture variations. Cao et al. [5] expanded the scope with FaceWarehouse, offering an extensive database of facial expressions, fostering further research in 3D face fitting and animation. Additionally, models like BFM [10] and FLAME (Faces Learned with an Articulated Model and Expressions) [22] have refined the representation of facial features, enabling more accurate modeling and animation. In recent years, research on learning the parameter space of 3DMM and reconstructing 3D facial parameters from images has seen a

rapid increase. Particularly, models like Deep3DFaceRecon [7] utilize deep neural networks to directly estimate 3DMM parameters from images, enabling highly accurate and detailed 3D facial reconstruction. HRN [21] adopt a hierarchical approach to extract facial features and predict 3DMM parameters for multi-step reconstruction. Additionally, methods like HiFace [6] and DECA [8] have enabled the restoration of detailed facial components and the animation of 3D faces. From these studies, extracting parameters for 3D face reconstruction from images has become feasible, indirectly addressing one of the major challenges in 3D facial research, which is the scarcity of 3D facial datasets. Drawing inspiration from such studies, we trained our model using pairs of 3DMM coefficients of 3D faces obtained from images and corresponding text, instead of insufficient 3D face-text data pairs. By utilizing the 3DMM coefficients extracted from images rather than using the images directly, we were able to incorporate geometric information of 3D that images alone could not capture into our model.

2.2 Text-Driven 3D Manipulation

Since the emergence of CLIP [27], text-based image manipulation has become increasingly prevalent, leveraging the shared embedding space of images and text. In the domain of 3D face manipulation, efforts have been made to optimize the rendering image of 3D faces and the CLIP embedding values of given text to align, aiming to manipulate 3D faces based on text descriptions. Latent3D optimizes input 3D faces in the direction of text using TBGAN [9] and CLIP, while ClipFace [2] utilizes StyleGAN-ADA [16] and CLIP to optimize the parameters of 3DMM and the texture style in the direction of text, thus transforming facial expressions and textures. Additionally, TG-3DFace [30] draws inspiration from StyleGAN2 [18] to learn text styles and generate corresponding 3D faces using Tri-plane, proposing a method to manipulate them with CLIP. Moreover, approaches like DreamFace [31] and HeadSculpt [12] explore text-based manipulation of 3D faces using latent diffusion model [28] and CLIP, while FaceCLIP-NeRF [14] presents a method to deform expressions in the hyper space of NeRF using CLIP. However, these methods often allow manipulation of only certain aspects of 3D faces, and suffer from issues such as the presence of CLIP-based optimization processes or long inference times due to model size during the face manipulation process. We introduce a novel approach to learning the shared embedding space between 3DMM parameters and text, achieving extremely fast inference manipulation within 10s, without the use of complex model structures or optimization processes.

3 Overview

Our goal is to manipulate reconstructed 3D faces from images according to the direction indicated by the text. Our research proceeded in two steps: (1) Unlike previous methods, which iteratively align text and rendered images of 3D faces using CLIP’s image-text joint embedding space, we trained a text-to-parametric 3D face mapper that directly maps text to the parametric 3D

face space. Figure 2b illustrates the entire training process. We trained three mapping networks to handle the identity, expression, and texture aspects of the text, respectively, in order to map the text to the parametric 3DMM space (the orange mapping networks). This was achieved using a pre-trained image-parametric 3D face space mapper (the pink Image encoder) and an image-text joint embedding space (the green CLIP encoder). This framework eliminates the need for optimization during manipulation and allows for the manipulation of multiple 3D faces simultaneously using a single text. (2) After training the model, we use the trained three mapping networks to manipulate 3D faces based on text. The overall manipulation process is shown in Fig. 3. The Text-BFM Mapper shown in Fig. 3 is the same as the one in Fig. 2b. During manipulation, we use texture maps to obtain higher-quality textured 3D faces and employ a face parser for more disentangled manipulation. Those two steps are detailed in Sect. 4 and Sect. 5, respectively.

4 Training the Mapping Between Text and 3DMM Coefficients

To enable text-based 3D face manipulation, we first learn the relationship between the text latent space and the parametric 3D face space. We were inspired by the idea that the parameter space of 3D faces is a concatenation of parameter spaces containing the meaning of identity, expression, and texture of the face. Based on this insight, we aimed to create a model that can be more fittingly manipulated according to text. To achieve this, we reclassified the facial features into three parts: identity, expression, and texture, based on the existing dataset, and used them for training. Additionally, we designed the model to be capable of separately training each part.

4.1 Text Generation

As depicted in Fig. 2a, we constructed a new text dataset corresponding to the three elements (identity, expression, texture) constituting the parametric 3D face space. We directly separated the features of images into identity, expression, and texture parts using the image annotation of the Multi-Modal CelebA-HQ dataset [19]. Since our goal was to manipulate only the front part of the face, annotations for accessories, hair, and other parts not relevant to the manipulation were not utilized during the separation process. Additionally, to supplement the relatively sparse information on expression in the Multi-Modal CelebA-HQ dataset, we also utilized the AffectNet dataset. The separated annotations were then converted into sentences using an LLM (Large Language Model) and used as input values. Specifically, we employed the Qwen [3] and Llama2 [29] models for this task. In cases where the annotation information corresponding to identity, expression, or texture was missing for an image, the sentence for that particular part was masked with a None value.

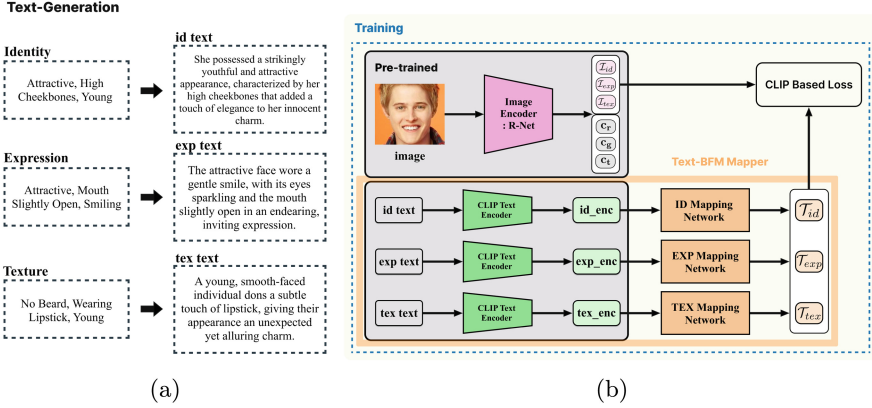


Fig. 2. (a) An example of generated text data. Using the face image annotation and LLM model, we generated new text data related to each element comprising the 3D facial parameter space and utilized it for training. (b) An overview of our training framework. We train a text encoder using image-text pairs to embed the 3D face representation implied by text. Images are embedded into 3DMM coefficients through a pre-trained image encoder, which serve as features of the 3D face. We then freeze the pre-trained image encoder and text encoder before training the mapping network.

4.2 Training

Based on this image-text dataset, we train the three 4-layers MLP networks showed in Fig. 2b. Each of these mapping networks takes associated text as input and maps it to the parametric 3D face space. To enable direct mapping between text and 3D face parameters, as mentioned in Sect. 3, we utilize a pre-trained network capable of mapping facial images to the parametric 3D face space and the image-text joint embedding space of CLIP.

In Fig. 2b, the R-Net is a pre-trained CNN released by Deep3DFaceRecon [7] with weakly-supervised learning that takes a single image and outputs the coefficients for a 3DMM. The 3DMM coefficients are represented by the vector $coeff = (c_{id}, c_{exp}, c_{tex}, c_r, c_g, c_t) \in \mathbb{R}^{257}$; $c_{id} \in \mathbb{R}^{80}$, $c_{exp} \in \mathbb{R}^{64}$, $c_{tex} \in \mathbb{R}^{80}$, $c_r \in \mathbb{R}^3$, $c_g \in \mathbb{R}^{27}$, and $c_t \in \mathbb{R}^3$. The six components represent the encoded face identity, expression, texture, the degree of facial rotation, lighting coefficient, and position of the face in the image, respectively. And coefficients are used to generate the 3D face mesh M with face shape S and face texture T as follows:

$$\begin{aligned}
 M &= (S, T) \\
 S &= Recon_S(c_{id}, c_{exp}) = \bar{S} + B_{id} \cdot c_{id} + B_{exp} \cdot c_{exp} \\
 T &= Recon_T(c_{tex}) = \bar{T} + B_{tex} \cdot c_{tex}
 \end{aligned} \tag{1}$$

where \bar{S} and \bar{T} mean the average face shape and texture. B_{id} , B_{exp} , and B_{tex} are the PCA bases of identity, expression, and texture respectively, which are all scaled with standard deviations. [7] using the Basel Face Model(BFM) [26]

for \bar{S} , B_{id} and \bar{T} and use the expression bases B_{exp} of [11] which are built from FaceWarehouse [5].

We employ R-Net and the CLIP text encoder to train a new mapping network for establishing the correspondence between the 3DMM coefficients space and the latent space of text. Notably, camera and lighting information of the coefficients was not utilized during training. The mapping network comprises linear layers and activation functions, and during the training phase, we calculate the loss for each mapping network based on the CLIP loss, and the total loss is obtained as the sum of these losses:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{id} + \mathcal{L}_{exp} + \mathcal{L}_{tex} & (2) \\ \mathcal{L}_{id} &= (\mathcal{L}_{I_{id}} + \mathcal{L}_{T_{id}})/2 \\ \mathcal{L}_{I_{id}} &= -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(S(\mathcal{I}_i, \mathcal{T}_{id,i})/\sigma)}{\sum_{j=1}^N \exp(S(\mathcal{I}_i, \mathcal{T}_{id,j})/\sigma)} \\ \mathcal{L}_{T_{id}} &= -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(S(\mathcal{T}_{id,i}, \mathcal{I}_i)/\sigma)}{\sum_{j=1}^N \exp(S(\mathcal{T}_{id,i}, \mathcal{I}_j)/\sigma)} \end{aligned} \quad (3)$$

where \mathcal{I} and \mathcal{T} are the embedded coefficients for the image and text, respectively. Since the calculation method for the loss of the three mapping networks is the same, we have provided an example using the id mapping network here, and the same process is followed to calculate the remaining two losses. Note that, \mathcal{I}_i denotes the portion of the embedded value that influences facial features (the pink part of the embedding vector in Fig. 2b). $S(\cdot, \cdot)$ is a function that calculates the cosine similarity. N is the mini-batch size for the image-text pairs. And σ , a learnable parameter, is a logit scale value multiplied after calculating the cosine distance between image embedding vectors and text embedding vectors. For training, we utilized the CLIP text encoder pre-trained with the ‘ViT-B/16’ image encoder. We employed the Adam optimizer with learning rates set to 1e-5 and weight decay set to 0.2. Additionally, training was conducted with a batch size of 1024 on a RTX A6000 GPU.

5 Text-Driven 3D Face Manipulation

5.1 Manipulation Through Text Embeddings

The user can input an image of the face they wish to manipulate and text prompts into the ‘ITFaceEdit’ framework, which includes a pre-trained ‘Text-BFM Mapper’, to adjust the reconstructed 3D face according to the directions specified by the text. The text prompts can be divided into identity, expression, and texture, allowing for repetition based on the user’s criteria. For parts without corresponding text prompts, the value ‘None’ is used. When provided with BFM coefficients obtained from the image, the model manipulates the 3D face using the embedding values of the learned text through the Text-BFM Mapper.

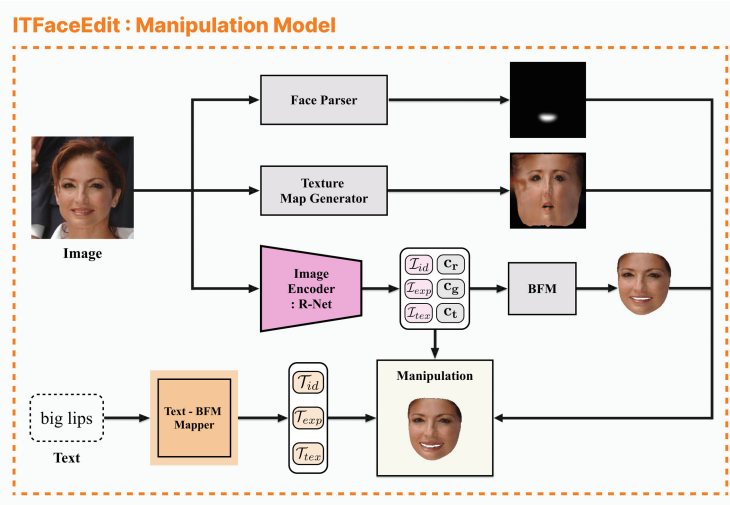


Fig. 3. An overview of the manipulation of 3D faces with a text prompt. (ITFaceEdit) When a text prompt is entered, the trained model maps the text to the parametric 3D face space and then uses it to manipulate the 3D face. Texture maps are used for higher resolution during manipulation, and a face parser is employed for more disentangled manipulations.

We manipulate the shape and color of the 3D face independently. Particularly for the color manipulation, we utilize texture maps generated from the input image to achieve higher-quality 3D face editing. Denoting the BFM coefficients obtained from the image for manipulation as \mathcal{I} , the embedding value of the text prompt as \mathcal{T} , and the resulting 3D face generated by the manipulation as M' , our manipulation processes for shape and color are defined as follows:

$$\begin{aligned}
 M' &= (S', T') \\
 S' &= Recon_S(\mathcal{I}_{id} + \alpha \cdot \mathcal{I}_{id} / |\mathcal{I}_{id}|, \mathcal{I}_{exp} + \beta \cdot \mathcal{I}_{exp} / |\mathcal{I}_{exp}|) \\
 T' &= C_{img} + Recon_T(\mathcal{I}_{tex} + \gamma \cdot \mathcal{I}_{tex} / |\mathcal{I}_{tex}|) - Recon_T(\mathcal{I}_{tex})
 \end{aligned} \tag{4}$$

where α, β , and γ determine the extent of deformation of the 3D face along the direction embedded by the text T , controlling the degree of change for identity, expression, and texture respectively. And C_{img} represents the color obtained from the texture map of the image. This texture map is generated through a pre-trained network of the HRN [21] model and used accordingly. Through our experiments, we found that the appropriate values for α, β , and γ are approximately between 0 and 20, 0 and 30, and 0 and 10 respectively.

5.2 Localization by Face Parsing

By individually training the mapping networks for each feature, we were able to achieve some level of disentanglement within the feature space. However, within the same feature space (such as identity), similar issues could arise. For instance, if the statistical analysis of the training data showed a correlation between large noses and large mouths, inputting ‘large nose’ might inadvertently result in a larger mouth. To address this entanglement issue within a single mapping network, we introduced a parser to further disentangle the features.

We employ a pre-trained face parsing model from CelebAMask-HQ [20] to obtain localization maps. This model already contains text pairs (classes) corresponding to each facial feature. Thus, when a word, synonym, or similar term related to a class is input, we mask and utilize the corresponding part. If it does not match any class, we use the entire face. Additionally, to address potential discontinuities in 3D manipulation arising from the use of the parser for natural manipulation, we mitigate this issue by applying Gaussian blur.

$$\begin{aligned}
 M' &= (S', T') \\
 S' &= \begin{cases} w_v \cdot \text{Recon}_{S;v}(\mathcal{I}_{id} + \alpha \cdot \mathcal{T}_{id} / |\mathcal{T}_{id}|, \mathcal{I}_{exp} + \beta \cdot \mathcal{T}_{exp} / |\mathcal{T}_{exp}|) \\ +(1 - w_v) \cdot \text{Recon}_{S;v}(\mathcal{I}_{id}, \mathcal{I}_{exp}) \\ \text{Recon}_{S;v}(\mathcal{I}_{id}, \mathcal{I}_{exp}) \end{cases} & \begin{array}{l} \text{if } v \in V_{\text{parsing}} \\ \text{otherwise} \end{array} \\
 T' &= \begin{cases} C_{img;v} + w_v \cdot (\text{Recon}_{T;v}(\mathcal{I}_{tex} + \gamma \cdot \mathcal{T}_{tex} / |\mathcal{T}_{tex}|) \\ - \text{Recon}_{T;v}(\mathcal{I}_{tex})) \\ C_{img;v} \end{cases} & \begin{array}{l} \text{if } v \in V_{\text{parsing}} \\ \text{otherwise} \end{array}
 \end{aligned} \tag{5}$$

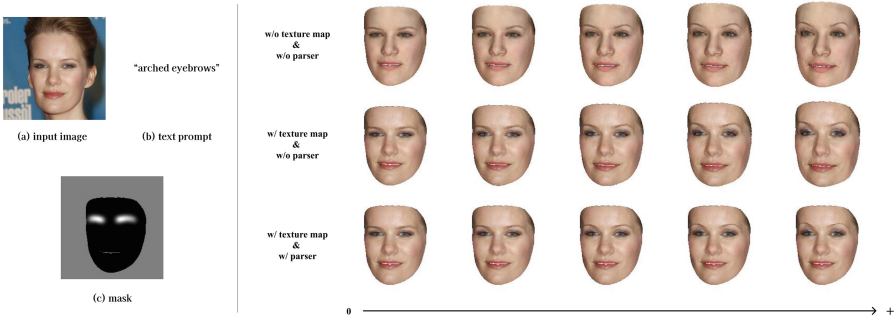


Fig. 4. Results of text-based 3D face manipulation. (a) is the image after reconstruction and the desired manipulation, (b) is the text prompt used for manipulation, and (c) is the mask of the face parser for the text prompt in (b). From the resulting images on the right, it’s evident that employing the texture map leads to much more realistic outcomes, while the use of the parser effectively suppresses the alteration of facial contours beyond the desired areas, such as eyebrows, observed previously. The leftmost image in the same row is the result of reconstruction the original image, and moving to the right, the results are obtained by increasing the values of α , β , and γ according to the text prompt.

In the above equation, v represents the set of mesh vertices corresponding to the areas masked using face parsing in the image for the prompt text. Additionally, w_v denotes the weights indicating the degree of manipulation for each v . The bottom row of images in Fig. 4 illustrates the results of manipulating the face mesh using parsing.

6 Result

Figure 5 presents the results of single-text-based 3D face manipulation using both texture maps and parsers. It demonstrates that our model performs well with various texts, reflecting the text accurately in the manipulation of identity, expression, and texture aspects. The last two columns in Fig. 5 depict the manipulation results from the mean face of BFM and omit the texture to emphasize geometric changes. This allows us to observe the direction of change from the mean face due to the text. Our model finds a mapping between the parametric 3D face space and a single text prompt without the need for inference for each text-image pair, enabling simultaneous application to multiple images, thereby allowing the manipulation of a large number of 3D faces concurrently. Additionally, our model adeptly manipulates 3D faces not only with a single text but also with multiple text prompts. Figure 6a demonstrates that our model performs well even with multi-text prompts. Figure 6b demonstrates that by using the three parameters α , β , and γ , we can manipulate or animate the 3D face to match the desired text to the preferred extent. The figure illustrates how each parameter appropriately adjusts the identity, expression, and texture aspects of the face.

Lastly, we present the results of an ablation study conducted on approximately 16,000 manipulated 3D faces in Table 1b. The FID scores were calculated using images rendered from manipulated 3D faces via CelebA-HQ and ITFaceEdit. The results confirmed that the use of texture maps contributed to the overall quality improvement of the final outcomes, as evidenced by the FID scores. The CLIP score represents the similarity between the rendered images and the corresponding text prompts, with the values in parentheses next to the CLIP scores indicating the increase in scores compared to the original images and text prompts. Through our experiments (Table 1a), we observed that 3D face reconstructions using texture maps resulted in visually superior outcomes, despite lower CLIP scores. This suggests that the CLIP model may not be perfectly aligned with human visual perception, potentially due to internal biases and the fact that it is not specifically trained on facial images and features. [2] also discusses that a high CLIP score does not necessarily reflect perceptually high-quality results and may have a bias towards color. While CLIP scores may not be an entirely accurate measure for 3D face manipulation, we still used them for our analysis as CLIP is one of the best existing models for capturing the correspondence between images and text. Therefore, for a more accurate analysis, we calculated the increase in CLIP scores based on the presence or

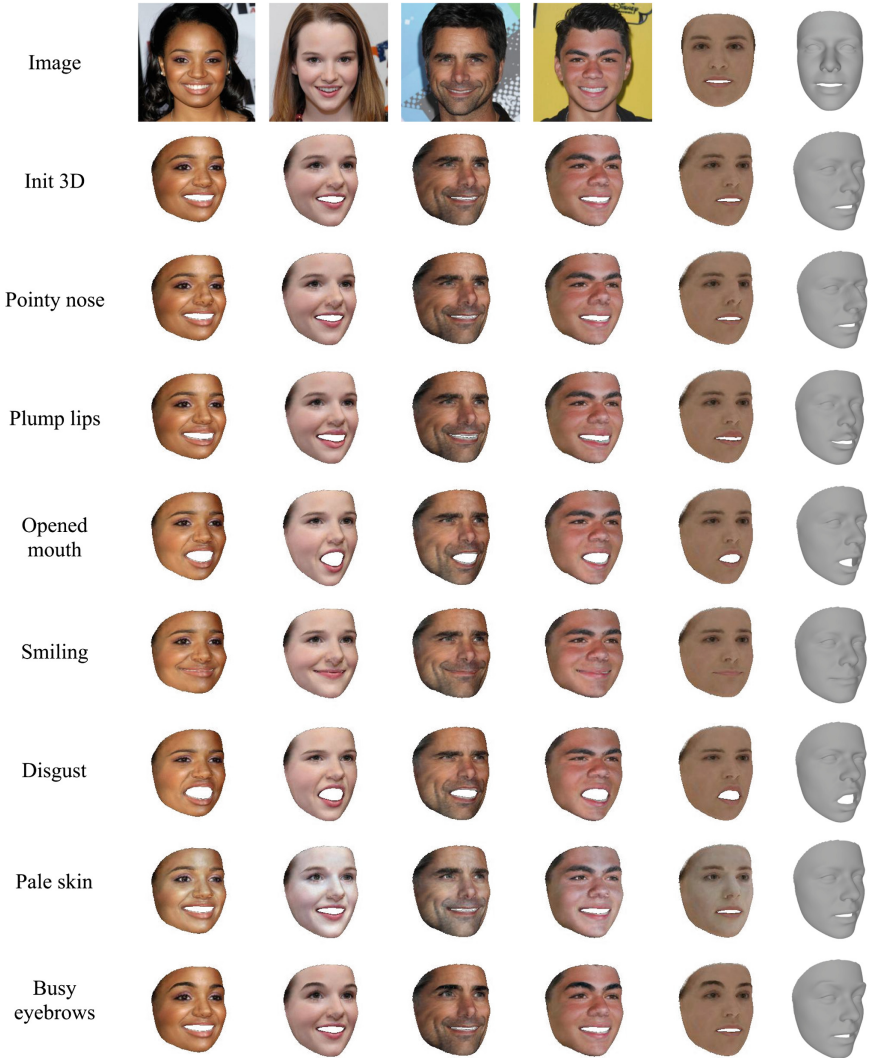
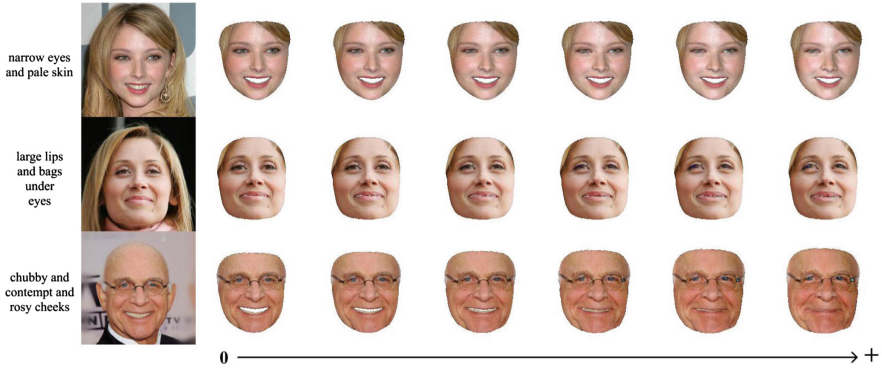


Fig. 5. Results of 3D face manipulation with various single text prompts. It can be observed that identity, expression, and texture are all being manipulated appropriately. The rightmost two columns show the manipulation results for the mean face of the BFM.

absence of texture maps before manipulation. Ultimately, we found that CLIP scores increased after 3D face manipulation in all cases, and that the combined metric for FID values and the increase in CLIP scores was highest when both texture map and parser were used.



(a) Results of manipulations from multi-text prompts



(b) Controlled study on parameters

Fig. 6. (a) Results of 3D face manipulation in response to multi-text prompts. The leftmost image shows the 3D face reconstructed from an image, and moving to the right, the results are obtained by linearly increasing the values of α (0 to 20), β (0 to 30), and γ (0 to 10) simultaneously. This demonstrates that we can animate the 3D face using these parameters. (b) The results of the controlled study for the bottom-most outcome in Figure (a) are shown. From the first row, the parameters α , β , and γ are manipulated one by one while fixing the other parameters at 0. As indicated by the meaning of each parameter, α focuses on chubby, β on contempt, and γ on rosy cheeks. It can be observed that each parameter is well-manipulated to reflect its specific focus.

Table 1. (a) The CLIP score between the original image, the 3D reconstruction using BFM without any manipulation, and the text. This shows that environmental variations can result in different CLIP scores despite the absence of any manipulations. (b) Ablation study on Multi-Modal CelebA-HQ dataset, where \downarrow means the lower the better, \uparrow means the higher the better. In the experiment, we used values of 20 for α , 30 for β , and 10 for γ . The increase in CLIP score was calculated based on the CLIP score corresponding to the presence or absence of the texture map before manipulation.

Resources	CLIP score \uparrow	
2D origin image	21.5977	
3D rendering w/o texture map	23.5648	
3D rendering w/ texture map	22.4195	
(a) Before manipulation		
Methods	FID \downarrow	CLIP score \uparrow (Increase)
w/o texture map, w/o parser	61.5505	24.3120 (0.7472)
w/ texture map, w/o parser	50.6994	23.8908 (1.4713)
w/ texture map, w/ parser	46.8236	23.9273(1.5078)
(b) After manipulation		

7 Comparison

To demonstrate the superiority of our model, we compare our results with existing models such as Text2Mesh and CLIP face (Fig. 7). Compared to basic text-based 3D mesh generation models, we exhibit better performance. Furthermore, when compared to ClipFace, we show stronger capabilities in manipulating the identity aspect.

One of the greatest strengths of our model is its fast learning speed and execution time. By leveraging the powerful 3D and text representations inherent in CLIP embeddings and 3DMM parameter space, we were able to construct the learning and manipulation processes with simpler networks. Table 3 illustrates the number of parameters among the models and the time taken for manipulating text-based 3D faces. It also includes TG-3DFace, a model whose comparison in terms of manipulation results is currently not feasible. As there have been no cases achieving manipulation in less than one minute so far, we consider this point to be our strong advantage. For the runtime calculations, we use a single RTX A6000.

Table 2. An example of visual data corresponding to the results in Table 1. Here, the text prompt “big lips” was used. This figure and Table 1 demonstrate that a high CLIP score may not always align with what is perceived as visually preferable by humans.

(a) 2D origin image



(b) An example of visual data






	Before manipulation	After manipulation	
		w/o parser	w/ parser
w/o texture map			-
w/ texture map			

Table 3. Comparison model parameters and manipulation time

Model	Text2Mesh	ClipFace	ITFaceEdit(Ours)
Total Parameters	151M	–	152M
Trainables Parameter	659 K	–	3.3M
Manipulation Time	12 min	6 min	8 s (w/o texture maps 1.14 s)

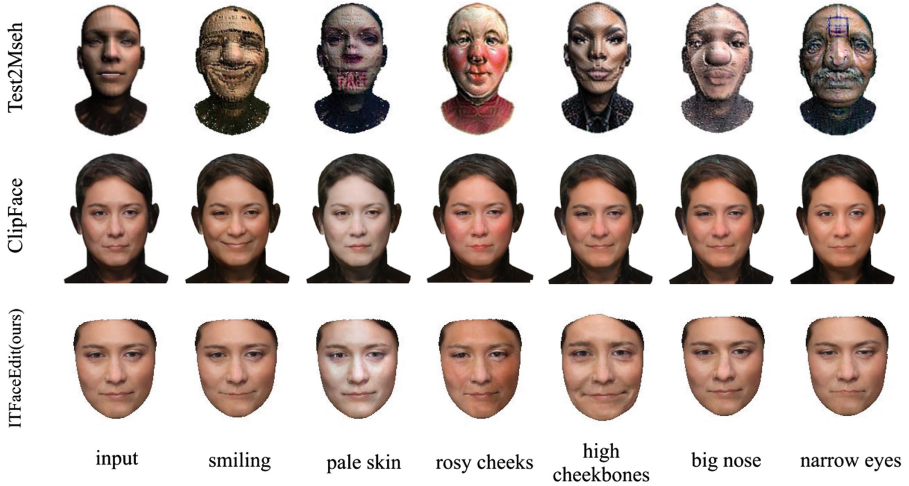


Fig. 7. Comparison of text-based 3D face manipulation results between existing methods and the ITFaceEdit model. The ITFaceEdit demonstrates better performance in accurately transforming the 3D face in the direction desired by the input text compared to Text2Mesh [24]. Additionally, it performs well in manipulating finer details such as the nose and eyes, surpassing ClipFace [2] in this aspect.

8 Limitations and Discussion

Due to the use of pre-trained models, our research was unable to adequately represent features such as hair, ears, and neck. Moreover, while we utilized text generation to accommodate a variety of texts, the model learns mappings during training, which can pose challenges in responding to unseen texts. In the future, we plan to address this issue by generating a more diverse set of text-image pairs. Additionally, because the use of CLIP scores may not be a perfect metric for evaluating text-based 3D manipulations, we anticipate that the development of new evaluation metrics could enable more accurate assessments.

9 Conclusion

We propose a model that learns the relationship between the coefficient space of 3D faces and the latent space of corresponding text, and present a method for text-based 3D face manipulation using this model. By utilizing 3DMM coefficients obtained from images in our model training, we constructed a model capable of learning the mapping between text and the latent space of 3D faces with fewer parameters. Moreover, our method enables 3D face manipulation without the need for iterative optimization processes during the inference stage, allowing for obtaining desired manipulation results within seconds. Additionally, to achieve richer textures in the results, we utilize texture maps obtained from images and employ face parsing to obtain more disentangled manipulation

results based on text inputs. Furthermore, our proposed approach allows the application of text embeddings obtained from a single text input to multiple 3D faces, and the degree of variation can be adjusted using parameters, enabling the use of our method for animation purposes.

Acknowledgment. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) [Grant No. 2023R1A2C2006201, Development of Simulation-Based Digital-Brain Editing Technology; and Grant No. 2022R1A4A1033856, Basic Research Program], and by the Institute of Information and Communication Technology Planning and Evaluation (IITP) grant funded by the MSIT [Grant No. RS-2019-II190079, Artificial Intelligence Graduate School Program, Korea University].

References

1. Abdi, H., Williams, L.J.: Principal component analysis. *Wiley Interdisc. Rev. Comput. Stat.* **2**(4), 433–459 (2010)
2. Aneja, S., Thies, J., Dai, A., Nießner, M.: ClipFace: text-guided editing of textured 3D morphable models. In: *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11 (2023)
3. Bai, J., et al.: Qwen technical report. arXiv preprint [arXiv:2309.16609](https://arxiv.org/abs/2309.16609) (2023)
4. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: *Seminal Graphics Papers: Pushing the Boundaries*, vol. 2, pp. 157–164 (2023)
5. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: FaceWarehouse: a 3D facial expression database for visual computing. *IEEE Trans. Visual Comput. Graphics* **20**(3), 413–425 (2013)
6. Chai, Z., et al.: HiFace: high-fidelity 3D face reconstruction by learning static and dynamic details. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9087–9098 (2023)
7. Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., Tong, X.: Accurate 3D face reconstruction with weakly-supervised learning: from single image to image set. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2019)
8. Feng, Y., Feng, H., Black, M.J., Bolkart, T.: Learning an animatable detailed 3D face model from in-the-wild images. *ACM Trans. Graph. (ToG), Proc. SIGGRAPH* **40**(4), 88:1–88:13 (2021)
9. Gecer, B., et al.: Synthesizing coupled 3D face modalities by trunk-branch generative adversarial networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020*. LNCS, vol. 12374, pp. 415–433. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58526-6_25
10. Gerig, T., et al.: Morphable face models—an open framework. In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 75–82. IEEE (2018)
11. Guo, Y., Cai, J., Jiang, B., Zheng, J., et al.: CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(6), 1294–1307 (2018)
12. Han, X., et al.: HeadSculpt: crafting 3D head avatars with text. In: *Advances in Neural Information Processing Systems*, vol. 36 (2024)

13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Adv. Neural. Inf. Process. Syst.* **33**, 6840–6851 (2020)
14. Hwang, S., Hyung, J., Kim, D., Kim, M.J., Choo, J.: FaceCLIPNeRF: text-driven 3D face manipulation using deformable neural radiance fields. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3469–3479 (2023)
15. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation (2018)
16. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. *Adv. Neural. Inf. Process. Syst.* **33**, 12104–12114 (2020)
17. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410 (2019)
18. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119 (2020)
19. Lee, C.H., Liu, Z., Wu, L., Luo, P.: MaskGAN: towards diverse and interactive facial image manipulation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
20. Lee, C.H., Liu, Z., Wu, L., Luo, P.: MaskGAN: towards diverse and interactive facial image manipulation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5549–5558 (2020)
21. Lei, B., Ren, J., Feng, M., Cui, M., Xie, X.: A hierarchical representation network for accurate and detailed face reconstruction from in-the-wild images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 394–403 (2023)
22. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4D scans. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* **36**(6), 194:1–194:17 (2017). <https://doi.org/10.1145/3130800.3130813>
23. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: *Proceedings of International Conference on Computer Vision (ICCV)* (2015)
24. Michel, O., Bar-On, R., Liu, R., Benaïm, S., Hanocka, R.: Text2Mesh: text-driven neural stylization for meshes. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13492–13502 (2022)
25. Mollahosseini, A., Hasani, B., Mahoor, M.H.: AffectNet: a database for facial expression, valence, and arousal computing in the wild. *IEEE Trans. Affect. Comput.* **10**(1), 18–31 (2017)
26. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D face model for pose and illumination invariant face recognition. In: *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 296–301. IEEE (2009)
27. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning*, pp. 8748–8763. PMLR (2021)
28. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695 (2022)
29. Touvron, H., et al.: LLaMA: open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023)

30. Yu, C., et al.: Towards high-fidelity text-guided 3D face generation and manipulation using only images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15326–15337 (2023)
31. Zhang, L., et al.: DreamFace: progressive generation of animatable 3D faces under text guidance. arXiv preprint [arXiv:2304.03117](https://arxiv.org/abs/2304.03117) (2023)



Unlearning Vision Transformers Without Retaining Data via Low-Rank Decompositions

Samuele Poppi^{1,2} , Sara Sarto¹ , Marcella Cornia¹ , Lorenzo Baraldi¹ ,
and Rita Cucchiara¹ 

¹ University of Modena and Reggio Emilia, Modena, Italy
{samuele.poppi,sara.sarto,marcella.cornia,
lorenzo.baraldi,rita.cucchiara}@unimore.it

² University of Pisa, Pisa, Italy
samuele.poppi@phd.unipi.it

Abstract. The implementation of data protection regulations such as the GDPR and the California Consumer Privacy Act has sparked a growing interest in removing sensitive information from pre-trained models without requiring retraining from scratch, all while maintaining predictive performance on remaining data. Recent studies on machine unlearning for deep neural networks have resulted in different attempts that put constraints on the training procedure and which are limited to small-scale architectures and with poor adaptability to real-world requirements. In this paper, we develop an approach to delete information on a class from a pre-trained model, by injecting a trainable low-rank decomposition into the network parameters, and without requiring access to the original training set. Our approach greatly reduces the number of parameters to train as well as time and memory requirements. This allows a painless application to real-life settings where the entire training set is unavailable, and compliance with the requirement of time-bound deletion. We conduct experiments on various Vision Transformer architectures for class forgetting. Extensive empirical analyses demonstrate that our proposed method is efficient, safe to apply, and effective in removing learned information while maintaining accuracy.

Keywords: Machine Unlearning · Low-Rank Adaptation · Vision Transformers · Image Classification

1 Introduction

Unlearning, the task of removing the impact of specific training data from a pre-trained model [35], is gaining attention in the Machine Learning community due to data protection laws like GDPR and the California Consumer Privacy Act [18, 43], that aim to guarantee every user with the “right to be forgotten” [14, 15] and require companies to erase personal data and their impact on trained

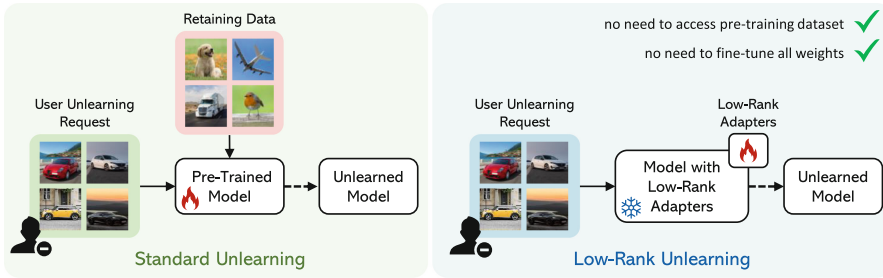


Fig. 1. Overview of our approach: we unlearn a class from a pre-trained image classification network without requiring access to the pre-training dataset and by injecting a low-rank adaptation matrix into the network weights.

models, upon request. Solutions aim to erase the influence of certain data points, essentially “untraining” the model to resemble one trained without them [4, 7, 24].

While removing one or more specific datapoints is crucial for addressing privacy concerns, the literature has also been recently investigating the removal of entire classes from a classification network [2, 33, 34, 39]. This setting, which is much more complex in both computational and algorithmic terms, is needed when the entire class is a source of privacy leak (*e.g.*, in a face recognition system), but also opens up new possibilities in terms of removing portions of the knowledge learned by the model when it is detrimental or not needed for a specific application scenario.

Previous attempts in this direction have tackled the unlearning phase as a fine-tuning step that involves all weights of a neural network [4, 20, 42], which makes untraining computationally complex and hardly feasible in practical scenarios in which a neural network, that is employed in production needs to be quickly adapted in response to a data removal request. Moreover, most approaches to unlearning require the load and access of the entire training dataset over which the network has been trained [19, 20, 42], putting another constraint on the practical applicability of previous approaches in a production environment. In these scenarios, indeed, the pre-training dataset can be significantly large or even not available if the model has been trained on private data and acquired from a third party.

To address these issues, we propose an unlearning algorithm that does not require fine-tuning the entire set of parameters of a pre-trained model and that does not need access to the dataset employed at training time (see Fig. 1). For each layer of a pre-trained neural network, we learn a low-rank matrix that is summed to the pre-trained parameters of the layer, with the aim of both forgetting unwanted data and retaining the original knowledge of the network. This is achieved by modeling a low-rank trainable decomposition and leaving the rest of the layers frozen, significantly reducing the number of trainable parameters and optimizer states needed for untraining. At the same time, this solution increases the network retaining capability and limits the loss in performance on the portion of the data to preserve, thus addressing one of the most important challenges in unlearning [12]. What is more, differently from many recent propos-

als [19, 42], this approach does not require explicit knowledge of the pre-training dataset. Indeed, the objective of retaining the original knowledge is achieved by regularizing the low-rank decomposition to increase its sparsity. Therefore, our approach can be applied without storing the pre-training dataset, thus lowering the storage requirements and allowing unlearning on backbones whose training dataset is not known or available.

Experimentally, we validate the proposed approach on modern image classification architectures based on the Vision Transformer [17] on CIFAR-10, CIFAR-20, and ImageNet-1k datasets, demonstrating its applicability and efficiency. In summary, our major contributions are as follows:

- We propose low-rank unlearning, the first approach to unlearn an entire class from a neural network by injecting a low-rank decomposition into the network parameters. Compared to existing work, low-rank unlearning greatly reduces the amount of computational resources required for unlearning.
- Our method is based on learning low-rank trainable matrices. In contrast to previous works which require complete access to retaining data, our approach allows for modeling an unsupervised retaining objective that does not require loading the pre-training dataset.
- We conduct extensive experiments to evaluate low-rank unlearning on image classification tasks. The results show that low-rank unlearning can rapidly and effectively forget undesired classes and outperform existing techniques.

2 Related Work

Machine Unlearning. Research efforts initially focused on unlearning solutions for traditional machine learning algorithms [7]. Instead, more recent approaches targeted erasing specific data points or entire classes from pre-trained deep neural networks [4, 20, 28, 29, 42]. One approach involves retraining the model from scratch on the remaining data, which is computationally demanding, especially for large-scale deep neural networks. To address these limitations, some works aim to accelerate the retraining process [4, 10, 21, 44]. In this context, Bourtole *et al.* [4] suggested partitioning the retraining dataset into shards to minimize data requirements. Another solution [21] involves storing and reusing gradient information during training. However, these methods often require modifying the original training process and are not easily applicable in real-world scenarios.

A different research line has focused on developing effective strategies to update network parameters according to the samples the model should forget, without retraining the entire model from scratch [11, 20, 42]. For example, Golatkar *et al.* [20] introduced a scrubbing procedure to remove information from parameters by adding noise. However, these methods face challenges with large datasets. To address this issue, a subsequent work [19] proposed splitting the network weights into core non-linear weights and linear user parameters, allowing for selective deletion without loss of accuracy.

Recently, other proposals for tackling the unlearning task have been presented, where unlearning is done either by retraining the model with a teacher-student paradigm with competent and incompetent teachers [42] or by shifting the decision boundary of a deep neural network to emulate the behavior of a model trained without samples of the forget class [9]. While almost all the above-mentioned works need the set of data the network should not forget, Cha *et al.* [8] proposed a novel instance-wise unlearning framework in which a set of instances is deleted from the original model by intentionally misclassifying them. Our work, in contrast, focuses on unlearning entire classes through a fine-tuning strategy that involves learning low-rank decomposition matrices. This allows for a reduction of the computational complexity of the unlearning procedure while achieving good retaining capabilities.

Some other research efforts have been dedicated to few-shot [36, 45] and zero-shot [12] machine unlearning. While the former concerns a setting in which only a few samples of the target data are available, the latter imposes the constraint that no training data are available to perform the unlearning task. While these settings are closely related to our proposal, we instead place ourselves in a more realistic scenario, in which all the unwanted samples are available, while having access to the rest of the pre-training dataset is not required.

Low-Rank Adaptation. During fine-tuning, updating all parameters of a pre-trained model is computationally expensive. Parameter-Efficient Fine-Tuning (PEFT) addresses this problem by optimizing a small portion of parameters, leaving the backbone model unchanged. Among PEFT methods, Low-Rank Adaptation (LoRA) [23] is one of the most popular approaches since it only requires tuning small low-rank matrices, achieving comparable performance compared to full fine-tuning across a wide range of tasks. For its efficiency, LoRA has been used across different research fields, like the fine-tuning of large language models to adapt them for various multimodal tasks [5, 6] and foundation models fine-tuning for improving their safety [38]. LoRA has also been used in facing the severe risk of privacy leakage in latent diffusion models [31]. Moreover, in federated learning, LoRA can be used to update local models efficiently without sharing the full model parameters. This ensures that sensitive data remains on local devices, enhancing privacy [41]. In contrast to previous research, we are the first, to the best of our knowledge, to design a LoRA-based solution to unlearn classes from pre-trained classification backbones.

3 Proposed Method

3.1 Preliminaries

Notation. Let $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N \subseteq \mathcal{X} \times \mathcal{Y}$ be the complete training set over which a deep neural network classification model has been trained, where $\mathbf{x}_i \in \mathcal{X}$ denotes an input image and $\mathbf{y}_i \in \mathcal{Y} = \{1, \dots, K\}$ its corresponding label, being K the number of classes. We denote with $\mathcal{D}_f \subset \mathcal{D}_{\text{train}}$ a set of training items whose impact needs to be removed from the model (*i.e.* the *forget set*) and $\mathcal{D}_r = \mathcal{D}_{\text{train}} \setminus \mathcal{D}_f$ the remaining set of training items over which we want

to keep prediction accuracy (more formally, the *retaining set*). In this work, we focus on the *class unlearning* scenario, in which \mathcal{D}_f will consist of all items from one class. Also, let $\mathcal{D}_{\text{test}}$ denote the test set used for evaluation.

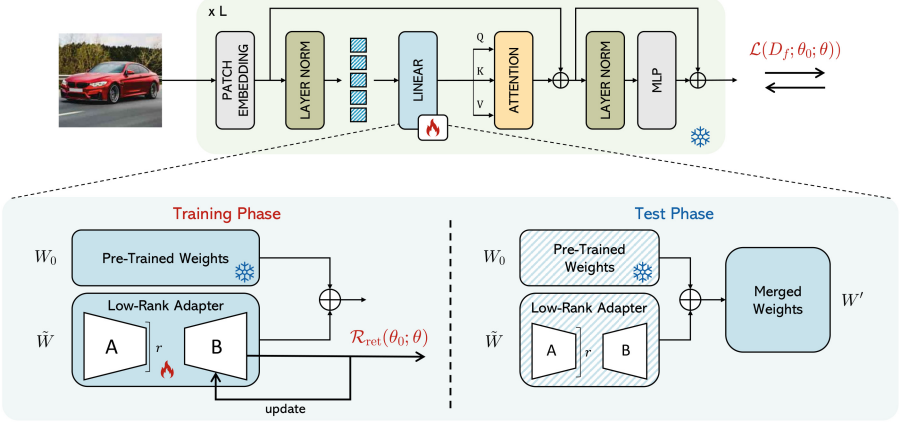


Fig. 2. Overview of the proposed low-rank unlearning solution. The pre-trained model is endowed with a trainable low-rank adapter, which is summed to existing network weights. During test, the low-rank adaptation is accumulated in the pre-trained weights to disable access to the previous state of the network.

Further, let $g_{\theta_0} : \mathcal{X} \rightarrow \mathcal{Y}$ indicate the original classification model pre-trained on $\mathcal{D}_{\text{train}}$, parametrized by a set of parameters θ_0 . The objective of the unlearning phase is to fine-tune g by moving θ towards a state of the parameters θ' where the information of \mathcal{D}_f is unlearned and the information in \mathcal{D}_r is maintained. The resulting model, therefore, should behave similarly to a model g_{θ^*} which has been trained from scratch on \mathcal{D}_r and which has never received gradient from samples in \mathcal{D}_f .

No-Retain Unlearning. Differently from previous works which require access to both \mathcal{D}_f and \mathcal{D}_r (or none of them, such as in the *zero-shot* setting defined in [12]), we suppose to have access to the pre-trained model g_{θ_0} and the forget set \mathcal{D}_f , without requiring access to the larger retain set \mathcal{D}_r . Our setting is more grounded and practical than previous ones. Indeed, it is reasonable to hypothesize that the data holder has natural access to the data that needs to be removed, *i.e.* \mathcal{D}_f . Also, the right to be forgotten gives the data holder up to a month to remove the data, which largely settles our approach within the bounds permitted by law. On the other hand, it is desirable that an unlearning method does not need to process the whole training set, *i.e.* \mathcal{D}_r . The original training set, indeed, might not be completely known or available, as in the case of models pre-trained on private data and then fine-tuned. Even when the full training set

is available, however, processing during the unlearning phase inevitably requires higher computational costs and also increases storage costs.

3.2 Class-Wise Unlearning

The objective of unlearning is that of removing the impact of the set of samples in \mathcal{D}_f . This can be achieved by either imposing a misclassification or a re-labeling of those datapoints. In the first case, the network is trained to misclassify all datapoints in \mathcal{D}_f , *i.e.* so that $g_{\theta'}(\mathbf{x}) \neq \mathbf{y}$ for $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_f$. This can be done by performing gradient ascent over a classification loss with ground-truth labels, computed over the forget set, *i.e.*

$$\mathcal{L}_{\text{unl}}(\mathcal{D}_f; \theta) = -\mathbb{E}_{\mathbf{x}, \mathbf{y} \in \mathcal{D}_f} \mathcal{L}_{\text{CE}}(g_{\theta'}(\mathbf{x}), \mathbf{y}; \theta). \quad (1)$$

In the second case, instead, a gradient descent learning can be performed over a classification loss with labels different from the ground-truth ones for all the samples in \mathcal{D}_f , *i.e.*

$$\mathcal{L}_{\text{unl}}(\mathcal{D}_f; \theta) = \mathbb{E}_{\mathbf{x}, \mathbf{y} \in \mathcal{D}_f} \mathcal{L}_{\text{CE}}(g_{\theta'}(\mathbf{x}), \mathbf{y}'; \theta), \text{ with } \mathbf{y}' \neq \mathbf{y}. \quad (2)$$

Whatever the above choice is, one of these unlearning objectives alone would induce forgetting what the network has learned on \mathcal{D}_r , therefore reducing its final performance after responding to a removal request. Under a setting in which direct access to \mathcal{D}_r is allowed, this could be avoided by performing gradient descent with a cross-entropy loss on \mathcal{D}_r , to refresh its knowledge continuously during untraining. In our scenario, in which access to \mathcal{D}_r is not considered, instead, a regularization loss can be added to keep the weights of the network close to θ_0 during the unlearning phase. The unlearning loss, therefore, becomes

$$\mathcal{L}(\mathcal{D}_f, \theta_0; \theta) = \mathcal{L}_{\text{unl}}(\mathcal{D}_f; \theta) + \mathcal{R}_{\text{ret}}(\theta_0; \theta), \quad (3)$$

where $\mathcal{R}(\cdot)$ is a regularizer that aims at overcoming forgetting of knowledge on the remaining data \mathcal{D}_r . Usually, this regularization is implemented by considering either the magnitude of weight change (*i.e.* $\theta' - \theta_0$) and its sparsity [25] or sample importance [8].

3.3 Low-Rank Unlearning

The unlearning procedure is, essentially, a fine-tuning process $\theta_0 \rightarrow \theta'$ induced by the loss \mathcal{L} . While previous literature has focused on fine-tuning the entire set of parameters θ_0 without imposing constraints on the selection of trainable weights, we instead hypothesize that this fine-tuning should happen in a low-rank space. Under this hypothesis, a complete fine-tuning of θ_0 is unnecessary and, potentially, also detrimental as it leaves the door open to overfitting on $\mathcal{D}_{\text{train}}$. In a scenario in which \mathcal{D}_r is not accessible, moreover, constraining the unlearning phase to happen in a low-rank space helps to retain the original knowledge of the model that has been learned on \mathcal{D}_r .

Without loss of generality, in the following, we describe our approach for the case of a fully-connected layer. While these are a key ingredient of many Transformer-based models as they build up the attention operator, our approach can also straightforwardly be extended to convolutional layers. Given a pre-trained layer f , with weight $W_0 \in \theta_0$, $W_0 \in \mathbb{R}^{d \times k}$ and bias $b \in \theta_0$, which applies a transformation $f(x) = xW_0^\top + b$ to its input tensor $x \in \mathbb{R}^k$, we re-parametrize its transformation during the unlearning phase by adding a low-rank trainable component \tilde{W} , initialized from zero. We then fine-tune only the low-rank decomposition, leaving the rest of the layer frozen. Formally,

$$f(x) = x \underbrace{W_0^\top}_{*} + x \overbrace{\tilde{W}^\top}^{\boxtimes} + \underbrace{b}_{*}. \text{ with } \tilde{W} = BA, \quad (4)$$

where A and B provide a bottleneck that creates a low-rank decomposition (denoted with \boxtimes , above), with $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$ and r being the rank of the decomposition. During unlearning, W_0 and b are kept frozen ($*$) and we backpropagate gradient only on A and B . These are respectively initialized with a Gaussian initialization and with zero, so that, at the beginning of the unlearning phase $\tilde{W} = BA$ is a zero matrix and f behaves exactly as in the pre-trained state.

Constraining the unlearning phase inside the low-rank decomposition \boxtimes also provides a straightforward way to overcome the forgetting of knowledge with respect to \mathcal{D}_r , as limiting the magnitude of weight change during the unlearning phase can be done by simply constraining the magnitude of \tilde{W} . In continuity with previous works that suggest that unlearning should produce a sparse update of weights, we constrain \tilde{W} to be sparse by adding an L_1 regularization on B , as follows:

$$\mathcal{R}_{\text{ret}}(\theta_0; \theta) = \lambda \|\text{vec}(B)\|_1, \quad (5)$$

where $\text{vec}(\cdot)$ is the vectorization operator and λ a scalar non-trainable constant. As it can be noticed, this induces B to be sparse, which in turn makes \tilde{W} sparse. The regularizer can then be plugged into any unlearning loss, to realize an unlearning procedure that does not require access to the retain set. In the case of unlearning via misclassification, the complete loss thus becomes

$$\mathcal{L}(\mathcal{D}_f, \theta_0; \theta) = -\mathbb{E}_{\mathbf{x}, \mathbf{y} \in \mathcal{D}_f} \mathcal{L}_{\text{CE}}(g_{\theta'}(\mathbf{x}), \mathbf{y}; \theta) + \lambda \|\text{vec}(B)\|_1. \quad (6)$$

After untraining is performed, A and B will contain the modifications applied to layer f to remove the knowledge of \mathcal{D}_f while maintaining that of \mathcal{D}_r . The original knowledge of the network, though, will still be accessible through W . During the evaluation, W can be made inaccessible by just collapsing the decomposition settled in Eq. 4 back into a single parameter matrix, as follows:

$$W' \leftarrow W_0 + BA, \quad f(x) = xW' + b. \quad (7)$$

After performing this operation, the resulting unlearned network will also have the same number of parameters as the pre-trained model. Our approach is also visually depicted in Fig. 2.

3.4 Bounded Unlearning Loss

To realize proper unlearning, \mathcal{L} should be minimized, zeroing the regularization term and increasing the forget loss as much as possible. However, this involves some drawbacks. Firstly, as we perform gradient ascent, the forget loss is not bounded, like standard loss functions. Further, as the loss approaches negative infinity, we would end up having $\|\mathcal{L}_{\text{CE}}(\cdot)\| \gg \|\mathcal{R}_{\text{ret}}(\cdot)\|$, losing any numerical guarantee that the regularizer will maintain information on the retaining classes.

To overcome these issues, we propose to minimize the following objective, in which we employ the reciprocal of the forget loss, with a positive sign:

$$\mathcal{L}(\mathcal{D}_f, \theta_0; \theta) = \frac{1}{\mathbb{E}_{\mathbf{x}, \mathbf{y} \in \mathcal{D}_f} \mathcal{L}_{\text{CE}}(g_{\theta'}(\mathbf{x}), \mathbf{y}; \theta)} + \lambda \|\text{vec}(B)\|_1. \quad (8)$$

As it can be seen, the loss defined above can be minimized towards zero, imposing the unlearning loss to be maximized, and the retain regularizer to be minimized. Through the rest of the paper, the unlearning loss above will be referred to as *bounded unlearning loss*.

4 Experimental Evaluation

4.1 Experimental Setting

We conduct a set of different experiments to validate the effectiveness of low-rank unlearning, by comparing with baselines and state-of-the-art approaches.

Backbones. While most of the recent unlearning literature has employed small-sized CNNs [9], we argue that it is crucial to test the effectiveness of unlearning methods over modern image classification architectures. This choice increases the effectiveness of the comparisons by reflecting a scenario closer to a future production-like environment and helps to guide the literature toward developing models that are more useful in real-world applications. Following this line, we employ image classification backbones based on Vision Transformers, which have proven their effectiveness on a wide range of tasks [1, 3, 13]. In particular, we employ the original ViT model [17] in its Tiny and Small versions (*i.e.* ViT-T and ViT-S respectively) and the Swin Transformer architecture [30] in its Small configuration (*i.e.* Swin-S). Following concurrent works that have employed low-rank decompositions for fine-tuning language models [23], we apply the low-rank adapters to each linear layer producing the query, key, and value vectors.

Datasets. Following [11], we perform experiments on the CIFAR-10 dataset and on a modified version of CIFAR-100 where images are grouped in 20 super-classes by considering their semantic similarity. We refer to this modified version as CIFAR-20. Both datasets [27] contain 50,000 training and 10,000 validation samples. In all experiments, we follow the standard splits. Additionally, we extend

our analysis on the ImageNet-1k dataset [16] which contains images corresponding to 1,000 different classes. For these experiments, we perform unlearning on 10 random classes¹, using the original splits.

Baselines. To test and compare the effectiveness of the proposed strategies, we employ the following baselines: the *original model*, *i.e.* trained from scratch on the corresponding datasets reported in the tables using the standard cross-entropy loss, without performing any unlearning strategy; the *retrained model*, *i.e.* a model trained from scratch on \mathcal{D}_r ; the *fine-tuned model*, *i.e.* the original model fine-tuned on \mathcal{D}_r . Additionally, we implement two other unlearning baselines typically used in previous works [8, 9], namely *random labels* [22] where we fine-tune the model using randomly assigned labels for samples from \mathcal{D}_f , and *negative gradient* [20] in which the model is fine-tuned on \mathcal{D}_f using negative gradients (*i.e.* fine-tuned in the direction of gradient ascent). To validate the effectiveness of our model, we also design three model alternatives to measure the contribution of the proposed low-rank unlearning and bounded forget loss.

Metrics. To evaluate class-wise unlearning, we measure the accuracy scores on both retaining and forget sets of the validation split of each dataset (*i.e.* Acc_r and Acc_f respectively). Ideally, the accuracy on the retaining set should be close to that of the original model, while the accuracy on the forget set should be equal to zero, thus getting close accuracy with the retrained model.

Unlearning Details. To test the true capabilities of each of the baselines, we opt for maximizing the performance of each of them by running a separate grid search over their loss weights for each backbone and dataset. This is a reasonable choice, as in practice the data holder could run a similar grid search over its own architecture and data before deploying a model in production. Also, we adopt an early stopping procedure that considers the average between the retain accuracy and the opposite of the forget accuracy (*i.e.* $100 - \text{Acc}_f$), so as to evenly balance between the capabilities of forgetting and those of retaining. This is also different from what has been done in recent works [8] in which the early stopping criterion was set exclusively on Acc_r , thus showcasing the forget capabilities of an approach to the detriment of its retaining effectiveness.

In all experiments, we employ Adam [26] as optimizer with a batch size of 256. We use a learning rate equal to 0.0001 for the baselines employing retaining data. In our setting without the retaining set, instead, we use a learning rate of 0.01 and 0.00005 respectively for the models with and without low-rank fine-tuning. In our complete model, we set the λ regularization weight to 0.001 for ViT-T and 0.0025 for ViT-S and Swin-S. The rank of the decomposition r is always set to 8, as it performed favorably in our initial experiments.

¹ The classes that we consider are as follows: kite, mud turtle, triceratops, scorpion, peacock, goose, jellyfish, snail, flamingo, beagle.

Table 1. Class-wise unlearning performance, comparing our solution with baselines with access to retaining data and different ablations. Column \mathcal{D}_r indicates whether the method needs access to the retain set. Final accuracy scores are obtained by performing an unlearning stage for each of the dataset classes and then averaging the results.

			ViT-T		ViT-S		Swin-S	
		\mathcal{D}_r	Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓
CIFAR-10	Original model	-	82.0	82.0	84.0	84.0	89.8	89.8
	Retrained model	✓	80.9	0.0	85.4	0.0	88.8	0.0
	Fine-tuned model	✓	80.2	7.9	81.3	3.0	85.0	2.3
	Random labels [22]	✓	83.0	0.0	85.1	0.0	88.9	0.0
	Negative gradient [20]	✓	84.4	0.0	85.8	0.0	88.9	0.0
	Negative gradient w/ L_1 regularization	✗	80.8	0.3	82.2	1.0	85.4	2.1
	Negative gradient w/ low-rank	✗	80.9	0.1	82.5	0.9	85.4	1.8
	Bounded loss w/ L_1 regularization	✗	81.2	0.1	82.3	0.8	85.5	1.4
	Bounded loss w/ low-rank (Ours)	✗	81.9	0.1	83.5	0.8	86.0	0.8
	Original model	-	67.0	67.0	71.9	71.9	74.4	74.4
Retrained model	✓	64.2	0.0	69.7	0.0	72.7	0.0	
Fine-tuned model	✓	64.5	8.2	67.2	8.6	68.3	4.6	
Random labels [22]	✓	66.2	0.0	70.8	0.0	73.2	0.0	
Negative gradient [20]	✓	67.6	0.0	71.4	0.0	72.2	0.0	
Negative gradient w/ L_1 regularization	✗	62.9	1.1	68.0	1.2	67.9	3.8	
Negative gradient w/ low-rank	✗	63.0	1.0	67.8	1.0	67.9	3.8	
Bounded loss w/ L_1 regularization	✗	63.1	1.2	67.9	0.8	68.0	3.7	
Bounded loss w/ low-rank (Ours)	✗	63.5	0.9	68.2	0.8	68.2	3.4	

Table 2. Single-class unlearning performance on 10 randomly selected classes from ImageNet-1k, using ViT-Small as backbone. Averaged results and standard deviations are reported in the rightmost columns.

	\mathcal{D}_r	Class 1		Class 2		Class 3		Class 4		Class 5			
		Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓		
Original model	-	86.0	92	86.0	94.0	84.5	74.0	83.5	76.0	85.0	92.0		
Random labels [22]	✓	58.9	0.0	62.7	0.0	83.8	0.0	79.3	0.0	79.3	0.0		
Negative gradient [20]	✓	77.5	2.0	74.4	3.2	65.3	0.0	70.7	0.0	62.2	0.0		
Bounded loss w/ low-rank (Ours)	✗	68.0	0.0	61.3	6.0	70.4	0.0	74.9	0.0	69.8	0.0		
	\mathcal{D}_r	Class 6		Class 7		Class 8		Class 9		Class 10		Avg (ViT-Small)	
		Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓	Acc _r ↑	Acc _f ↓
Original model	-	84.0	82.0	86.0	74.0	81.9	75.9	81.9	80.0	82.9	93.9	84.2±1.5	83.4±8.2
Random labels [22]	✓	76.0	0.0	78.7	8.0	77.8	0.0	78.0	18.0	61.3	0.0	70.6±8.5	4.6±2.7
Negative gradient [20]	✓	59.5	4.0	69.8	0.0	68.9	0.0	78.6	0.0	48.9	0.0	67.7±8.4	0.9±1.5
Bounded loss w/ low-rank (Ours)	✗	68.7	2.0	75.1	0.0	73.1	0.0	79.1	0.0	70.4	0.0	71.1±4.6	3.0±6.6

4.2 Utility Analysis

A machine unlearning method is effective when the unlearned model contains little or no information about the forget data items contained in \mathcal{D}_r . In the following, we evaluate the utility of the different baselines and that of the proposed approach, by also conducting ablation experiments. Results are reported in Table 1, over the three considered backbones and on both datasets.

We begin by considering the retrained model in comparison with the original model, which provides an upper bound in terms of accuracy on both the retaining set and the forget set and which, on the other side, needs access to the full training dataset. We then compare with three unlearning approaches which need access to the retaining data as well, and which therefore operate on a setting that is easier than the one on which we operate. Namely, we compare the model trained with random labels and one trained with negative gradient.

Firstly, we notice that both the random labels approach and the negative gradient approach are effective in forgetting the data contained in \mathcal{D}_f , as testified by their zero accuracies on the forget class. We also notice that fine-tuning the original model on \mathcal{D}_r is effective in erasing the information on \mathcal{D}_f to some degree, even though the baseline fails to reach a zero accuracy and also takes more training time, as already noted by previous literature [9]. Also, the random labels approach struggles to maintain good retain accuracy, which can be explained by the significant ground-truth noise caused by the model. The negative gradient approach, instead, is effective at both forgetting data and maintaining the accuracy on other classes and reaches an accuracy on \mathcal{D}_r which is comparable, or even superior in some cases, to that of the retrained model.

We then turn our attention to the “no retain set” scenario, in which the model has no access to \mathcal{D}_r , where we investigate the performance of the negative gradient approach, that of a model trained with the bounded unlearning loss, that of a model trained with low-rank unlearning, and that of our complete model. For the negative gradient baseline, we employ the negative gradient loss on \mathcal{D}_f , in conjunction with an L_1 regularization on weight change, without employing low-rank matrices. This baseline, while being consistent for comparing with our final model, is also in line with recent works that demonstrated the effectiveness of sparsity in unlearning [25].

Employing an effective unlearning approach such as the negative gradient one, without having access to the retain set, results in a significant lowering of the retain accuracy, of around one accuracy point on all backbones and datasets. The addition of low-rank fine-tuning and the bounded unlearning loss, instead, provide a good recovery of the retaining capabilities of the models, without compromising the forget accuracy, or even enhancing it in some cases. On CIFAR-10 and ViT-T, the combination of bounded loss and low-rank learning enhances the retain accuracy by 1.1 points, while keeping the same forget accuracy, while on ViT-S it increases the retain accuracy by 1.3 points and improves the forget accuracy by 0.2 points. The same applies to the Swin-S model, where low-rank unlearning significantly increases unlearning performance with respect to the negative gradient baseline. The same can be observed on CIFAR-20, and over all the three considered backbones. For instance, low-rank unlearning on ViT-T increases the retain accuracy by 0.6 points, while obtaining a 0.9 forget accuracy.

In Table 2, we instead report the results on ImageNet-1k, considering the ViT-S model and 10 randomly selected classes. In this setting, we compare our model with the negative gradient and random labels baselines, which both leverage the retain set during unlearning. For completeness, we also show the results of

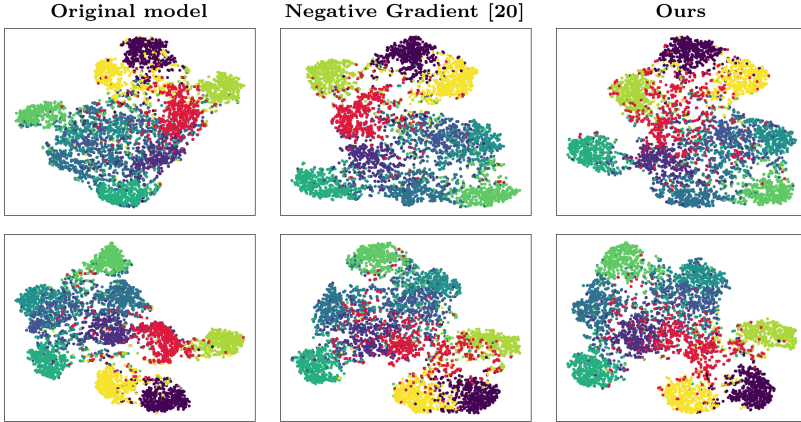


Fig. 3. Visualization of the embedding space of pre-trained models and unlearned models on CIFAR-10 using the ViT-Tiny (top) and ViT-Small (bottom) backbones. Samples from the unlearned class are represented with red markers. (Color figure online)

the original model which represents an upper bound reference. From the results, it can be noticed that performing unlearning on the ImageNet-1k dataset is in general more challenging: while all considered models can adequately unlearn the selected classes, they experience some performance drops on the retain set. It is worth noting, however, that our model can achieve competitive retain accuracy scores, performing better or on par than the two considered competitors which both have access to the retain set during unlearning. All reported results outline that our method achieves the utility guarantee effectively and low-rank decomposition is a viable solution to perform unlearning without retaining data.

4.3 Visualizations

Embedding Space Visualizations. To better visualize the effect of unlearning on the decision space of the network, we report t-SNE [32] visualizations of the embedding space produced by the classification layer of the ViT-Tiny and ViT-Small models, both unlearned on CIFAR-10. For comparison purposes, we report the visualization obtained by the pre-trained model, by the negative gradient approach (which employs retaining data), and by low-rank unlearning. As it can be seen from Fig. 3, low-rank unlearning brings the embedding of unlearned samples toward other classes, thus realizing the unlearning objective. Noticeably, unlearned samples are moved to the embedding space of multiple classes, which is a valuable effect. The opposite, indeed, could represent the Streisand effect and provide more information about the forgetting data [20]. We observe that this can happen in the case of the negative gradient baseline, especially with the ViT-Tiny backbone, despite this baseline having access to retaining data. Low-rank unlearning, instead, appears to be less prone to collapsing unwanted data in the embedding space of a single class.

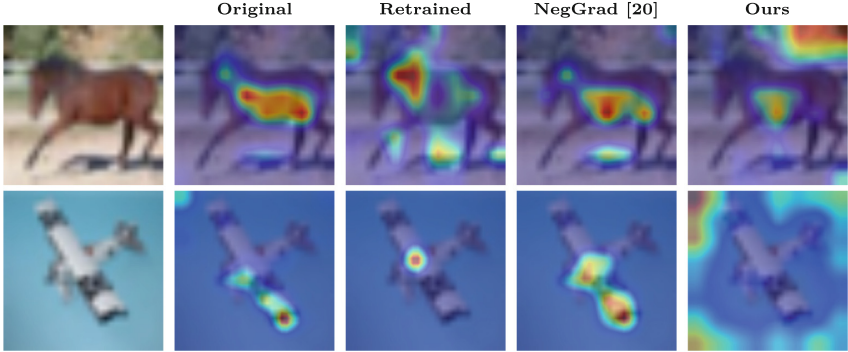


Fig. 4. Grad-CAM [40] attention visualizations of different unlearning methods.

Further, we can notice that the clusters representing the other classes have remained compact after unlearning, which testifies that their knowledge has been retained. In particular, we observe that there is no significant difference between retained clusters in negative gradient and those in low-rank unlearning. Therefore, low-rank unlearning can effectively unlearn the embedding of a class, while correctly maintaining the knowledge of retained classes.

Attention Maps. We also report the attention maps of models untrained with our approach and with other approaches from the literature. In particular, we employ Grad-CAM visualizations [40], which have been originally developed for convolutional neural networks and which can seamlessly be adapted to Vision Transformers [37]. We do this by reducing the stride of the first convolutional layer of a ViT, so as to have an attention map with higher resolution. The maps represent the areas of the input image that the network has paid more attention to when predicting the final output distribution. Results are reported in Fig. 4, where we can observe that the attention maps produced by low-rank unlearning are significantly sparser than those produced by the negative gradient baseline, and tend to shift the attention from the foreground object to the background, in a manner that closely resembles the behavior of the retrained model. These results confirm that low-rank unlearning is effective in removing knowledge of the unwanted class, and also that models fine-tuned with our approach closely resemble the models retrained without the unwanted data.

4.4 Computational Analysis

One of the most significant benefits of low-rank unlearning is that we greatly reduce memory occupation and storage requirements, and we can also reduce the computation times required to unlearn a given class. In particular, for a Vision Transformer trained with Adam, low-rank unlearning reduces the VRAM requirement by up to $2/3$ with $r = 8$. Further, we also observed a reduction in

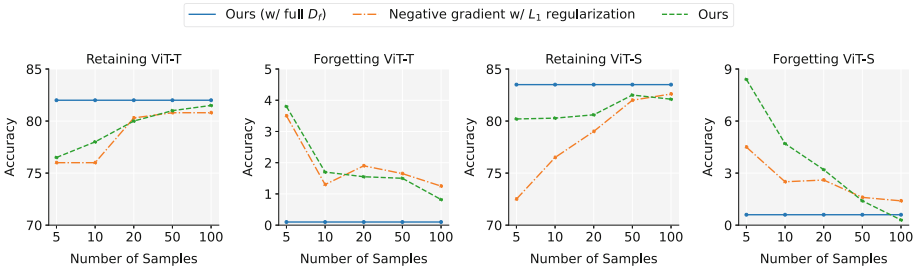
Table 3. Unlearning times measured as the average number of seconds required to unlearn a single class. Results are reported on the CIFAR-10 dataset.

	\mathcal{D}_r	Unlearning Time (s)	
		ViT-Tiny	ViT-Small
Negative gradient w/ L_1 regularization	✗	7.96	10.38
Bounded loss w/ low-rank (Ours)	✗	6.83	9.16

backward times without increasing the number of iterations needed to bring the model to early stopping. The detailed unlearning times are reported in Table 3, in which we measure the number of seconds required to unlearn a single class, averaging the results on all CIFAR-10 classes. We compare the unlearning times of our complete model with those of the baseline without low-rank decomposition and bounded unlearning loss (*i.e.* negative gradient with L_1 regularization), using a single P100 GPU to run the experiments. As it can be seen, our proposal does not negatively impact unlearning times but on the contrary, it contributes to improving the efficiency of model training when employing both ViT-Tiny and ViT-Small model versions, thus further confirming the appropriateness of our solution. It is also worth noting that at test time our model has the same number of parameters as the original model. This guarantees that we do not introduce any additional latency during inference compared to a retrained model.

4.5 Few-Shot Unlearning Analysis

Finally, we analyze the impact of using a reduced number of samples from \mathcal{D}_f to perform unlearning. The results are shown in Fig. 5 using a variable number of samples from the CIFAR-10 forget set. Also in this case, we compare our model with the negative gradient baseline with L_1 regularization and also report the accuracy upper bounds obtained by our model trained using all samples in \mathcal{D}_f . Notably, using 100 forget samples per class (*i.e.* instead of 5,000 as in the full CIFAR-10 forget set) does not significantly deteriorate the performance. Both

**Fig. 5.** Retaining and forget accuracy scores when varying the number of forget samples for each class. Results are reported on the CIFAR-10 dataset.

ViT-Tiny and ViT-Small models achieve better retaining accuracy scores when using our configuration compared to the baseline. In terms of forget accuracy, our model can effectively forget the selected class, especially with 20, 50, and 100 forget samples. When instead using a very limited number of forget samples per class, the accuracy scores are comparable or slightly worse than those obtained by the baseline, which however loses in terms of retaining capabilities.

5 Conclusion

We have presented low-rank unlearning. Our approach removes the knowledge of entire classes from a pre-trained neural network by learning a low-rank adaptation of the network weights, which is then accumulated into the original weights at test time. By leveraging a sparsity regularization, our approach does not need access to the retain dataset, making it suitable for production-like environments. Further, compared to previous approaches, it requires less computational resources, less memory allocation, and fewer storage requirements at training time. Extensive experimental results have demonstrated its performance in unlearning of modern image classification architectures. We envision our work as a step in the direction of efficient and effective unlearning.

Acknowledgments. This work has been conducted under a research grant co-funded by Leonardo S.p.A. and supported by the EU Horizon project “ELIAS - European Lighthouse of AI for Sustainability” (No. 101120237).

References

1. Barsellotti, L., Amoroso, R., Cornia, M., Baraldi, L., Cucchiara, R.: Training-free open-vocabulary segmentation with offline diffusion-augmented prototype generation. In: CVPR (2024)
2. Baumhauer, T., Schöttle, P., Zeppelzauer, M.: Machine unlearning: linear filtration for logit-based classifiers. *Mach. Learn.* **111**(9), 3203–3226 (2022)
3. Bontempo, G., Porrello, A., Bolelli, F., Calderara, S., Ficarra, E.: DAS-MIL: distilling across scales for MIL classification of histological WSIs. In: Greenspan, H., et al. (eds.) *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*. MICCAI 2023. LNCS, vol. 14220, pp. 248–258. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-43907-0_24
4. Bourtole, L., et al.: Machine unlearning. In: IEEE S&P (2021)
5. Caffagni, D., et al.: The revolution of multimodal large language models: a survey. In: ACL Findings (2024)
6. Caffagni, D., et al.: Wiki-LLaVA: hierarchical retrieval-augmented generation for multimodal LLMs. In: CVPR Workshops (2024)
7. Cao, Y., Yang, J.: Towards making systems forget with machine unlearning. In: IEEE S&P (2015)
8. Cha, S., Cho, S., Hwang, D., Lee, H., Moon, T., Lee, M.: Learning to unlearn: instance-wise unlearning for pre-trained classifiers. In: AAAI (2024)
9. Chen, M., Gao, W., Liu, G., Peng, K., Wang, C.: Boundary unlearning. In: CVPR (2023)

10. Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., Zhang, Y.: When machine unlearning jeopardizes privacy. In: ACM CCS (2021)
11. Chundawat, V.S., Tarun, A.K., Mandal, M., Kankanhalli, M.: Can bad teaching induce forgetting? Unlearning in deep networks using an incompetent teacher. In: AAAI (2023)
12. Chundawat, V.S., Tarun, A.K., Mandal, M., Kankanhalli, M.: Zero-shot machine unlearning. *IEEE Trans. IFS* **18**, 2345–2354 (2023)
13. Cornia, M., Baraldi, L., Cucchiara, R.: Explaining transformer-based image captioning models: an empirical analysis. *AI Commun.* **35**(2), 111–129 (2022)
14. Cucchiara, R., Baraldi, L., Cornia, M., Sarto, S.: Video surveillance and privacy: a solvable paradox? *Computer* **57**(3), 91–100 (2024)
15. Dang, Q.V.: Right to be forgotten in the age of machine learning. In: ICADS (2021)
16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)
17. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. In: ICLR (2021)
18. Goddard, M.: The EU General Data Protection Regulation (GDPR): European Regulation that has a global impact. *IJMR* **59**(6), 703–705 (2017)
19. Golatkar, A., Achille, A., Ravichandran, A., Polito, M., Soatto, S.: Mixed-privacy forgetting in deep networks. In: CVPR (2021)
20. Golatkar, A., Achille, A., Soatto, S.: Eternal sunshine of the spotless net: selective forgetting in deep networks. In: CVPR (2020)
21. Graves, L., Nagisetty, V., Ganesh, V.: Amnesiac machine learning. In: AAAI (2021)
22. Hayase, T., Yasutomi, S., Katoh, T.: Selective forgetting of deep networks at a finer level than samples. arXiv preprint [arXiv:2012.11849](https://arxiv.org/abs/2012.11849) (2020)
23. Hu, E.J., et al.: LoRA: low-rank adaptation of large language models. arXiv preprint [arXiv:2106.09685](https://arxiv.org/abs/2106.09685) (2021)
24. Izzo, Z., Smart, M.A., Chaudhuri, K., Zou, J.: Approximate data deletion from machine learning models. In: AISTATS (2021)
25. Jia, J., et al.: Model sparsity can simplify machine unlearning. In: NeurIPS (2023)
26. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
27. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
28. Lin, S., Zhang, X., Chen, C., Chen, X., Susilo, W.: ERM-KTP: knowledge-level machine unlearning via knowledge transfer. In: CVPR (2023)
29. Liu, J., Xue, M., Lou, J., Zhang, X., Xiong, L., Qin, Z.: Muter: machine unlearning on adversarially trained models. In: ICCV (2023)
30. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: ICCV (2021)
31. Luo, Z., Xu, X., Liu, F., Koh, Y.S., Wang, D., Zhang, J.: Privacy-preserving low-rank adaptation for latent diffusion models. arXiv preprint [arXiv:2402.11989](https://arxiv.org/abs/2402.11989) (2024)
32. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *JMLR* **9**(11), 2579–2605 (2008)
33. Neel, S., Roth, A., Sharifi-Malvajerdi, S.: Descent-to-Delete: gradient-based methods for machine unlearning. In: ALT (2021)
34. Nguyen, Q.P., Low, B.K.H., Jaillet, P.: Variational Bayesian unlearning. In: NeurIPS (2020)
35. Nguyen, T.T., Huynh, T.T., Nguyen, P.L., Liew, A.W.C., Yin, H., Nguyen, Q.V.H.: A survey of machine unlearning. arXiv preprint [arXiv:2209.02299](https://arxiv.org/abs/2209.02299) (2022)
36. Pawelczyk, M., Neel, S., Lakkaraju, H.: In-context unlearning: language models as few shot unlearners. arXiv preprint [arXiv:2310.07579](https://arxiv.org/abs/2310.07579) (2023)

37. Poppi, S., Cornia, M., Baraldi, L., Cucchiara, R.: Revisiting the evaluation of class activation mapping for explainability: a novel metric and experimental analysis. In: CVPR Workshops (2021)
38. Poppi, S., Poppi, T., Cocchi, F., Cornia, M., Baraldi, L., Cucchiara, R.: Safe-CLIP: removing NSFW concepts from vision-and-language models. In: ECCV (2024)
39. Poppi, S., Sarto, S., Cornia, M., Baraldi, L., Cucchiara, R.: Multi-class unlearning for image classification via weight filtering. *IEEE Intell. Syst.* (2024)
40. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
41. Sun, Y., Li, Z., Li, Y., Ding, B.: Improving loRA in privacy-preserving federated learning. In: ICLR (2024)
42. Tarun, A.K., Chundawat, V.S., Mandal, M., Kankanhalli, M.: Fast yet effective machine unlearning. *IEEE Trans. NNLS* (2023)
43. de la Torre, L.: A Guide to the California Consumer Privacy Act of 2018. Available at SSRN 3275571 (2018)
44. Wu, Y., Dobriban, E., Davidson, S.: DeltaGrad: rapid retraining of machine learning models. In: ICML (2020)
45. Yoon, Y., Nam, J., Yun, H., Kim, D., Ok, J.: Few-shot unlearning by model inversion. arXiv preprint [arXiv:2205.15567](https://arxiv.org/abs/2205.15567) (2022)



gWaveNet: Classification of Gravity Waves from Noisy Satellite Data Using Custom Kernel Integrated Deep Learning Method

Seraj Al Mahmud Mostafa¹(✉), Omar Faruque¹, Chenxi Wang², Jia Yue^{3,4}, Sanjay Purushotham¹, and Jianwu Wang¹

¹ Department of Information Systems, University of Maryland, Baltimore County, Baltimore, MD, USA

{serajmostafa,omarfaruque,psanjay,jianwu}@umbc.edu

² Goddard Earth Sciences Technology and Research (GESTAR) II, University of Maryland, Baltimore County, Baltimore, MD, USA

chenxi@umbc.edu

³ Department of Physics, Catholic University of America, Washington, DC, USA

⁴ NASA Goddard Space Flight Center, Greenbelt, MD, USA

jia.yue@nasa.gov

Abstract. Atmospheric gravity waves occur in the Earth's atmosphere caused by an interplay between gravity and buoyancy forces. These waves have profound impacts on various aspects of the atmosphere, including the patterns of precipitation, cloud formation, ozone distribution, aerosols, and pollutant dispersion. Therefore, understanding gravity waves is essential to comprehend and monitor changes in a wide range of atmospheric behaviors. Limited studies have been conducted to identify gravity waves from satellite data using machine learning techniques. Particularly, without applying noise removal techniques, it remains an underexplored area of research. This study presents a novel kernel design aimed at identifying gravity waves within satellite images. The proposed kernel is seamlessly integrated into a deep convolutional neural network, denoted as **gWaveNet**. Our proposed model exhibits impressive proficiency in detecting images containing gravity waves from noisy satellite data without any feature engineering. The empirical results show our model outperforms related approaches by achieving over 98% training accuracy and over 94% test accuracy which is known to be the best result for gravity waves detection up to the time of this work. We open sourced our code at <https://rb.gy/qn68ku>.

Keywords: Gravity Wave Detection · Pattern Recognition · Custom Kernel · Hybrid Deep Neural Network · Remote Sensing

1 Introduction

Gravity waves (GW) are physical perturbations caused by gravity's restoring force in a planetary environment, distinct from gravitational waves [13].

In Earth’s atmosphere, various disturbances like airflow over mountains, jet streams, and thunderstorms create atmospheric gravity waves, displacing air parcels and leading to wave patterns resembling ripples on water [20]. These waves have broad effects, including localized vertical motion, turbulence, and impacts on the middle atmosphere’s dynamics. They contribute to the transport of heat, momentum, and atmospheric composition [4], as well as influencing weather patterns, precipitation, cloud formation, tidal waves [10], and aviation safety due to clear air turbulence [23]. Due to the significant impact of gravity waves, there has been a surge of interest in their detection. AI researchers, along with domain experts, are using machine learning techniques to understand the phenomenon better and improve detection accuracy. However, the single-channel satellite dataset used in this study presents challenges. Firstly, limited ground truth availability makes data accuracy verification difficult. Secondly, the dataset contains significant noise interference such as city lights, clouds, and instrumental horizontal/vertical lines. Lastly, there’s a restricted amount of data identified by domain experts. While gravity wave data are publicly accessible [1], the ground truth is not provided. Domain experts helped select data containing gravity wave patterns. The dataset comprises night bands obtained from the VIIRS satellite’s day/night band (DNB) [7], introducing noise from city lights and clouds that may reduce classification accuracy [12]. Applying denoising methods, like Fast Fourier Transform (FFT) [11], can blend gravity wave patterns with noise, making them harder to isolate.

To enhance the classification of gravity waves in satellite images, we introduce a specialized convolutional kernel, the checkerboard kernel. This custom kernel is designed to improve pattern recognition during convolution, particularly for complex features and noisy environments. Inspired by successful applications in computer vision, such as depth completion and image classification, our approach emphasizes the importance of tailored kernels. Studies by Ku et al. [16], Pinto et al. [26], and Zhang et al. [33] underscore the effectiveness of custom kernels in diverse scenarios. In our specific application, the use of the custom kernel enhances classification accuracy for gravity waves in satellite images, capturing finer details and outperforming conventional methods. While deep neural networks (DNNs) excel in various tasks [15, 19, 24, 30], but may not always be optimal for detecting shapes in images due to the extensive training data required. Effective categorization and minimizing assumptions are crucial, as advised by [25]. In situations with limited data or subtle shapes, deep learning may struggle, suggesting the need for complementary techniques [21]. Additionally, adjusting weights to amplify signals enhances learning [35]. In our study with noisy data, we propose a hybrid method using a custom kernel to capture challenging shape information for deep neural networks. We focus on accurately identifying gravity waves within images, even in noisy conditions. We introduce ‘gWaveNet’, a hybrid deep neural network integrating the ‘checkerboard’ kernel in the first layer. Our main contributions are: 1) *We designed a unique ‘checkerboard’ kernel capable of detecting gravity wave features amidst noise.* This kernel acts as a specialized filter, highlighting gravity wave patterns for more accurate detection. 2)

We propose ‘gWaveNet’, a novel deep neural network incorporating our custom-designed checkerboard kernel to enhance gravity wave detection. This integration allows the model to effectively learn and recognize intricate wave patterns. 3) We conducted extensive ablation studies, exploring various training configurations (as discussed in Sect. 5), employing checkerboard kernels of different sizes to demonstrate our model’s performance. These experiments helped us understand the impact of different setups and modifications needed to address the challenges of detecting gravity waves in noisy datasets. Traditionally, detecting gravity waves required expert knowledge and handcrafted features tailored to wave characteristics [18]. However, our model automatically learns and extracts relevant features from the data, reducing the need for domain-specific knowledge and enhancing our approach’s generalizability.

The rest of the paper is structured as follows. Section 2 reviews relevant literature related to this research. Section 3 discusses the facts about data, its collection process, preprocessing steps, and groundtruth information. In Sect. 4, we detail the methodologies employed for both the proposed kernel and the model. Experiment details and results discussed in Sect. 5. Lastly, we conclude the paper in Sect. 6.

2 Related Works

In this section, we review related work that is particularly relevant to custom kernels and the detection of gravity waves.

Custom Kernel for Image Detection. Yousafzai et al. proposed a polynomial custom kernel based on Mercer’s theorem with the support vector machine for acoustic waveform classification with noise signals [32]. Ku et al. proposed a simple algorithm to generate the depth information of LIDAR sensor data and outperformed deep learning-based methods [16], which applied four (5×5) custom kernels of circle, cross, diamond, and full shapes to compute the missing data points in sensor data which improved the accuracy of depth information. Suresha et al. proposed the integration of a custom multiquadric kernel function ($k = \sqrt{\|x_1 - x_j\|^2 + c^2}$) with the KPCA algorithm to generate important features for image classification [29]. This custom kernel which resembles the sigmoid kernel helps to extract new features from the original feature space and also increases classification accuracy and reduces computational complexity, time complexity, and storage issues. Zhang et al. [34] performed texture and object classification using kernel-based discriminative analysis of local features to make the distinction between different classes. Also, Zhang et al. [33] proposed the integration of custom kernels with the 3D depth information to better analyze and classify 3D objects in the presence of noise and low-intensity data.

Deep Learning Approaches for GW Detection. CNN models often improve object detection performance greatly in the computer vision domain. As gravity waves can be detected by analyzing satellite images, the CNN model can be used for this task very efficiently. Thus far, some research has been conducted regarding gravity wave detection using deep learning models. There is a notable study by Lai et al. that developed a convolutional neural network-based auto-extraction program, which extracts gravity wave patterns in all-sky airglow images [17]. In this work, the process involves using cleaner images, and on top of that, there is still a step of discarding images that are not suitable for the model to learn during the training process. In our case, we fed all images that include noise. Matsuoka et al. used U-net deep neural network model to estimate the gravity wave from reanalysis data [22]. Recently, paper [28] proposed Deep Dictionary Learning algorithm to integrate deep learning and dictionary learning to detect gravity waves by learning different kernels. InceptionV3 deep learning model was used in [11] with transfer learning technique to detect gravity waves from satellite images. In addition to these advancements, attention-based techniques, particularly the Transformer, have gained popularity for image classification. Dosovitskiy et al. [8] demonstrated that Transformers can outperform CNN models. Chen et al. [6] highlighted the potential of Transformers in capturing multiscale features from images. In the context of remote sensing, Bazi et al. [5] introduced a Transformer-based classification model. In our research, we also evaluated the performance of the Vision Transformer on complex satellite dataset to assess its suitability.

In conclusion, both custom kernel-based methods and deep learning-based approaches have their strengths and weaknesses in image classification tasks. Custom kernels can capture fine-grained details in the images and are especially useful when the shape information is subtle. On the other hand, deep learning-based approaches are highly flexible and can learn complex representations of the images given enough training data. Our study seeks to integrate both techniques to perform gravity wave detection from noisy satellite data with higher accuracy.

3 Data Preprocessing

For this investigation, we used the Day/Night Band (DNB) images from the Visible Infrared Imaging Radiometer Suite (VIIRS) instrument onboard the Suomi NPP satellite [7]. VIIRS DNB observes board band upwelling radiance in the visible region. VIIRS DNB has a wide swath ($\sim 3,000$ km) and a relatively high spatial resolution at 1 km approximately. Pixels within a 6-minute granule ($\sim 4,000 \times 3,000$ pixels) are stored in one Hierarchical Data Format version-5 (HDF5) [2] file. The raw HDF5 files contain radiance measurements within the wavelength range of $0.5\mu\text{m}$ to $0.9\mu\text{m}$. To highlight the airglow from gravity wave events, nighttime images under new moon conditions are used in this study. As a result, the DNB radiance could be extremely low with a value in the order of magnitude of $-10^{-9}\text{W}/\text{cm}^{-2}\text{sr}^{-1}$. To comprehend easily, we performed preprocessing on the raw data, ensuring that the array values are within a specific range while maintaining their relative distribution. This involved subtracting the

minimum value from all array elements, scaling by the median, and normalizing to 0.5. Normalizing to this reference point enables easier visual comparison and analysis of the data. Finally, we transformed the intensity distribution from an approximate normal distribution to a uniform one, while preserving the accurate range of values. We present examples of our data processing in Fig. 1. Sub-figure 1a shows normalized data from a raw HDF5 file, while Sub-fig. 1b displays a pre-processed image derived from the same file. Sub-figure 1c illustrates an image containing gravity waves, along with various unwanted elements such as clouds, city lights, and instrumental noise. Finally, Sub-fig. 1d presents an image without gravity waves. The algorithm for raw data preprocessing is detailed in our previous work [11].

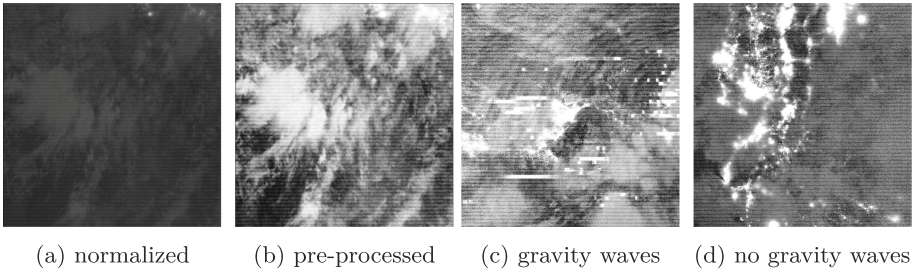


Fig. 1. Examples of Data Processing and Image Types: (a) Normalized data from raw HDF5 file (b) Pre-processed image from the same file (c) Image with gravity waves, including unwanted elements (clouds, city lights, instrumental noise) (d) Image without gravity waves.

We started by gathering raw satellite data stored in HDF5 format, focusing on 50 files chosen by domain experts containing gravity waves and noise. Using the GDAL library we normalize the HDF5 files and convert them into PNG format. We generated 200×200 grayscale image patches, overcoming the limited dataset challenge. Given the infrequent occurrence of gravity waves, we employed data augmentation, including rotation and flip, to increase the number of patches with gravity waves. We manually categorized patches into two classes: “gw” for gravity waves and “ngw” for non-gravity waves, maintaining a balanced dataset of 5,985 image patches in each class, resulting 11,970 in total.

4 Research Methodology

Checkerboard Kernel for Gravity Wave Pattern Detection. Since our dataset contains excessive noise, including city lights, clouds, and instrumental noise (horizontal/vertical lines), we designed the checkerboard kernel to capture all types of gravity wave patterns while excluding the noise. In this experiment,

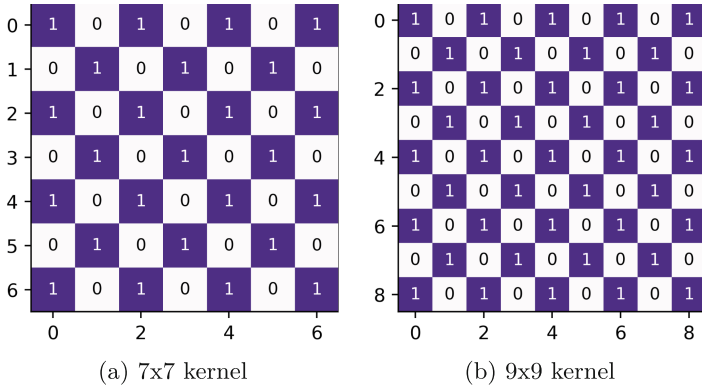


Fig. 2. Examples of ‘checkerboard’ kernels proposed in the *gWaveNet*.

we utilized kernels of different sizes, including 3×3 , 5×5 , 7×7 and 9×9 , with the same pattern as illustrated in Fig. 2. We used the kernel in the first layer of our proposed deep-learning model (discussed later in this Section) to generate low-level features from input grayscale satellite images. The proposed custom kernel is defined as: $K(x, y) = [(x + y + 1)\%2]$ $x \in (0, \dots, w)$ and $y \in (0, \dots, w)$. Here, w is the length of the square kernel (K) and (x, y) is any position in the 2-dimensional kernel of size $(w \times w)$. The visualization of the proposed kernel is provided in Fig. 3a. This kernel is capable of finding various shapes and orientations of the gravity wave traces from the noisy input dataset. We apply the checkerboard kernel on different images using the convolutional approach to observe the effects of the proposed kernel, as illustrated in Fig. 4. The figure demonstrates that the features corresponding to gravity waves are successfully extracted. To enhance visibility, we have highlighted these extracted features in yellow.

The concept behind utilizing a custom kernel within a deep learning approach is to enable the model to extract specific features relevant to the problem at hand. The conceptual purpose of the custom kernel, illustrated in Fig. 3a, is to capture intricate and nonlinear gravity wave patterns within the images that aligns the properties of gravity waves, even in the presence of noise. The alternating pattern of ‘1’s and ‘0’s within the checkerboard kernel enables the identification of lines, representing gravity waves, and gaps between them respectively. Additionally, the repeating pattern in the kernel helps identify recurring ripple-like nonlinear lines of varying shapes in the images. Our experimental results (in Sect. 5) indicate that the proposed *gWaveNet* outperformed all other approaches.

Proposed Checkerboard Kernel-Based Hybrid Neural Network. Our proposed *gWaveNet* in Fig. 5 is a deep convolutional network with 15 layers. It features a hybrid model with the proposed checkerboard kernel integrated at

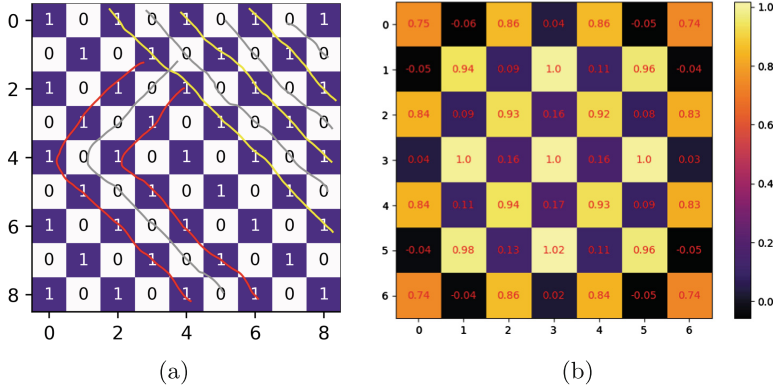


Fig. 3. Checkerboard kernel concept and application. (a) Proposed kernel capturing gravity wave patterns: yellow lines - potential linear waves, red lines - potential non-linear waves, gray lines - wave gaps. (b) Learned kernel from trained *gWaveNet* (Color figure online).

the beginning, comprising 6 convolutional layers, followed by ReLU activations, 6 max-pooling layers, 2 dense layers, and 1 dropout layer, and a sigmoid activation at the end. Our custom kernel is placed in the first convolutional layer along with ReLU activation in the network. The hybrid architecture is designed for binary classification tasks on grayscale satellite images. The rationale behind incorporating the custom kernel in the initial layer is to specifically extract features aligned with those associated with gravity waves, allowing subsequent layers to identify similar intricate patterns. Focusing on the complexity of the problem, a large number of kernels is used in the earlier layers of the model which helps to learn more low-level features from the data. Gradually the number of kernels is reduced in the later layers to combine earlier features into more problem-specific high-level features. To avoid overfitting from the custom kernel, we apply L2 regularization in the second convolutional layer.

We conducted experiments by configuring the layer in the proposed network to be either trainable or non-trainable, incorporating the checkerboard kernel (details are in Sect. 5). The checkerboard kernel’s original weights of ‘0’ or ‘1’ can be updated during the model training process based on its configuration. Weights in the trainable layer are adjustable, directly influencing the model’s predictions by learning from input data during training if the method is configured as trainable with the checkerboard kernel integrated. On the other hand, in the non-trainable configuration, although the weights are fixed, certain layers can still update their statistical entities such as mean and variance, indirectly impacting the model’s output, as noted in [3]. This indicates that non-trainable layers continue to impact the model’s performance. We further explain, how the settings of the kernels with trainable/non-trainable would affect the overall model performance in Sect. 5.

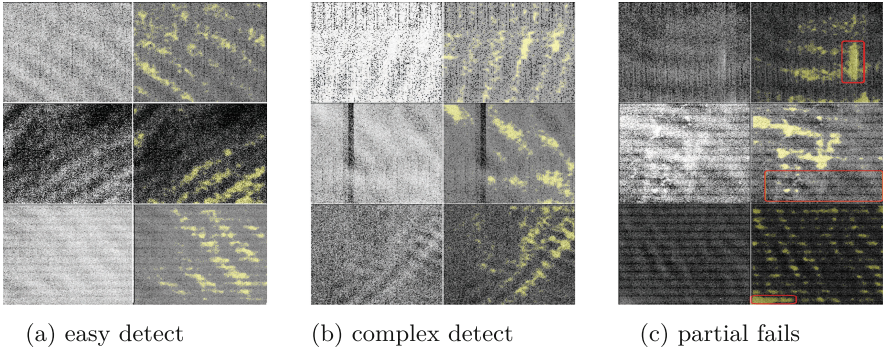


Fig. 4. The proposed kernel highlights extracted gravity wave features in sub-figures (a), (b), and (c). The left column of each sub-figure displays the actual PNG files, while the right column highlights the extracted gravity wave features. In sub-figure (a), simple patterns of gravity waves are detected. Sub-figure (b) shows detection of complex patterns even in the presence of noise, such as vertical lines in the top image and a black bar in the second image. However, sub-figure (c) exhibits partial failure in detection, as indicated by the red rectangular box. (Color figure online)

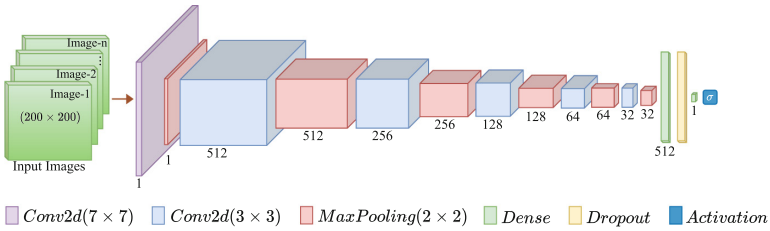


Fig. 5. Architecture of the proposed *gWaveNet* network.

Experiment and Evaluation Setup. For our experiment, all the implementation is carried out using Keras 2.11 and TensorFlow 2. The training and testing processes for both the proposed and baseline models were conducted on a GPU machine equipped with 20 gigabytes of memory.

We considered four distinct configurations to train all the models. In the “trainable” configuration, we integrated the proposed checkerboard kernel into the first layer, with this layer set as trainable (*trainable = true*). Conversely, the “non-trainable” configuration retained the same concept, but with the first layer designated as non-trainable (*trainable = false*). The “no-custom-kernel-layer” (denoted as ‘nckl’ in Tables) configuration indicated the absence of a custom kernel in the first layer. Lastly, the “kernel-applied-prior-training” (denoted as ‘kapt’ in Tables) configuration was similar to “no-custom-kernel-layer,” except the kernel was applied to the images as a filter before initiating the model training process.

All models are trained for 2,000 epochs with a batch size of 128 on the training dataset. The training and validation dataset is split into a 65:35 ratio, and we reserved 240 image patches for testing, which were never used in training. The hyperparameters for the remaining methods are kept intact for experimentation with our data. We employ binary cross-entropy loss function during training to calculate the detection loss and The stochastic gradient descent (SGD) method, coupled with the binary cross-entropy loss objective function, is chosen for model optimization. In terms of evaluations, we used overall detection accuracy and the $F1$ score as evaluation metrics.

5 Experiment Results

In this section, we present our findings from various aspects such as, kernel size, trainable or non-trainable, kernel applied prior training (kapt) and no custom kernel layer (nckl) which is discussed in Ablation Study. We conducted extensive experiments to answer the following questions. Q1: *How does the performance of our hybrid deep learning model with the integrated checkerboard kernel compare with state-of-the-art (SOTA) approaches?* Q2: *How can the capability of the proposed model be inferred from the ablation studies?* Q3: *How can the model learn without denoising the data?* Q4: *Is the kernel integration approach generalizable?* Q5: *How does the model perform when trained with a reduced amount of data?*

Comparing with State-of-the-Art Techniques. We evaluated our model against five advanced State-of-the-Art (SOTA) techniques in computer vision research. This includes the Vision Transformer (ViT) [8], an attention-based model, and VGG16 [27], chosen for its similar architecture to our proposed model. Additionally, we assessed three influential convolutional filters: Gabor [9], Sobel [14], and Laplacian [31]. Furthermore, our evaluation included an FFT denoising-based approach using Transfer Learning Mechanism [11].

We compared our model with State-of-the-Art techniques, summarized in Table 1 where the best results in each category are in bold and the overall best results are emboldened and underlined. First, we incorporated Gabor filters of size 7×7 with various orientations (0° , 30° , 60° , 120° , 150°) into our deep learning model considering the gravity wave patterns in the image. This process yielded high accuracy with an $F1$ score of 88.35%. Secondly, we applied Sobel filters (using 3×3 kernels) as a preprocessing step before training the model. Additionally, we integrated the Sobel filters into our proposed method during training, labeled as ‘Sobel.3×3.t’ with a trainable kernel. The result indicates that the model trained with Sobel filters integrated into the first layer achieved higher accuracy and $F1$ score compared to the model trained on images pre-processed by Sobel filters (denoted as, Sobel.3×3_pt). However, our detailed optimization plot reports overfitting as the training keeps progressing. We followed the same process for the Laplacian filter, using a 7×7 kernel. The results showed a similar pattern, with better accuracy obtained when the Laplacian filter was integrated into the model (Laplacian.7×7_t) as opposed to utilizing it

Table 1. Performance comparison of the proposed model against state-of-the-art techniques: ViT [8], VGG16 [27], Gabor [9], Sobel [14], Laplacian [31], and Transfer Learning approach using FFT denoised data [11].

Methods	Architecture	Accuracy			F1 Score	Train config.
		Train	Validation	Test		
ViT	Transformer	89.12	86.08	83.74	79.71	nckl
VGG16	VGG	100.00	81.52	61.91	60.15	nckl
VGG16.3×3_t	VGG	87.86	80.19	68.74	66.29	trainable
Gabor_7×7_t	gWaveNet	95.24	93.46	89.17	88.35	trainable
Sobel.3×3_kapt	gWaveNet	88.76	85.35	82.91	80.89	kapt
Sobel.3×3_t	gWaveNet	91.23	87.17	83.74	82.24	trainable
Laplacian_7×7_kapt	gWaveNet	86.44	83.50	76.66	74.38	kapt
Laplacian_7×7_t	gWaveNet	90.91	87.19	79.16	80.00	trainable
FFT	gWaveNet	76.59	75.04	70.66	69.50	nckl
FFT_7×7_nt	gWaveNet	92.68	91.64	84.47	82.19	non-trainable
FFT_7×7_t	gWaveNet	93.78	92.50	90.78	90.07	trainable
gWaveNet_5×5_nt	gWaveNet	94.40	93.16	93.75	91.89	non-trainable
gWaveNet_7×7_t	gWaveNet	98.10	96.53	94.21	93.69	trainable

as a data preprocessing step (Laplacian_7×7_pt). However, the model also shows overfitting similar to Sobel filter. During our testing of the ViT model, the results indicated 89.12% accuracy in training with 79.71% F1 score which is quite comparable to the performance of Sobel and Laplacian approaches. However, the overfitting is significant.

Comparing the VGG16 method with its base architecture and its modified architecture with our trainable approach (denoted as VGG16 and VGG16.3×3_t, respectively) reveals a notable difference. The base model is highly overfitted, showcasing high accuracy during training but experiencing a considerable drop in validation accuracy, nearly 20%. On the contrary, VGG16 with a 3×3 kernel integrated trainable layer improved across all metrics though there are inconsistencies. While it still falls short of outperforming other models, the enhancement from the base model is notable. This suggests the potential generalizability of our proposed kernel with other methods addressing Q4. Furthermore, performance comparisons between VGG16.3×3_t and our gWaveNet_noK model (Table 1 and Table 2, respectively) show that VGG16.3×3_t trails behind gWaveNet_noK in achieving competitive scores. Our experiments suggest that the gWaveNet_noK model, specifically designed for our noisy dataset performs better in extracting relevant features from noisy data while the base VGG16 falls short due to its architectural configuration.

We further compared models trained with denoised data using FFT techniques based on three training configurations which are, trainable, non-trainable and no-custom-kernel-layer. The experiments revealed that the highest train-

ing accuracy achieved with FFT-denoised images reached 93.78%. Notably, the model with a trainable layer outperformed the two other models with different training configurations for the same dataset. The results indicate a gradual improvement in model performance from the no-custom-kernel-layer to the non-trainable layer and finally to the trainable layer. However, with a 7×7 trainable layer the performance improved with consistent training. It is noteworthy that despite the high accuracy achieved by the model trained on denoised data, it still falls short compared to our proposed model trained using noisy data. This discrepancy could be attributed to the FFT transformation, which removes lower amplitude signals. There is a possibility that this process eliminated some gravity wave patterns that matched specific frequencies, leading to a lower overall performance score.

Referring to ‘gWaveNet_7x7.t’, our approach outperforms all the aforementioned SOTA techniques. The proposed model, featuring a trainable 7×7 custom kernel, achieved the best results in terms of different accuracies and F1 score. Notably, these accuracies demonstrate improved optimization without overfitting. These findings address Q1 at the beginning of this section, highlighting the performance of our proposed hybrid deep learning model (in Table 1) with the integrated checkerboard kernel.

Ablation Study. We thoroughly examined variety of configurations using the *gWaveNet* core architecture in our ablation study, as shown in Table 2. In all training configurations, except for the ‘non-trainable’ layers, we used various kernel sizes of 3×3 , 5×5 , 7×7 , and 9×9 , for all the *gWaveNet* models. We discuss training configuration wise performances as follows.

Model with Trainable or Non-trainable Layer. The model with a trainable layer with integrated kernels refers to learning following kernel patterns, while the non-trainable layer refers to the opposite. With the trainable layer with the checkerboard kernel integrated, training, validation and testing accuracies are achieved as high as 98.43%, 97.22%, and 94.21%, respectively, along with 93.69% F1 score across models for various kernel sizes. The non-trainable layer also demonstrated significant performance with accuracies of 97.53% in training, 95.60% in validation, and 93.75% in testing, along with an F1 score of 91.89%.

Models with Kernel-Applied-Prior-Training. We also evaluated the proposed *gWaveNet* model by applying the kernels to the images prior to training the models that also show significant performance. With this training approach, we achieved the best training accuracy of 97.77% with a competitive F1 score of 93.07%.

Model with No-Custom-Kernel-Layer. As our next evaluation, we experimented with no-custom-kernel-layer (denoted as gWaveNet_noK in Table 2). Without the custom kernel, the model exhibited relatively poor performance compared to other approaches in the table. The low F1 score indicates a large number of

Table 2. Ablation Studies.

Methods	Accuracy			F1 Score	Train config.
	Train	Validation	Test		
gWaveNet_3×3_t	97.17	95.23	92.43	92.00	trainable
gWaveNet_5×5_t	98.43	97.22	93.75	92.11	trainable
gWaveNet_7×7_t	98.10	96.53	94.21	93.69	trainable
gWaveNet_9×9_t	97.22	95.01	91.06	91.55	trainable
gWaveNet_3×3_nt	91.38	90.25	87.50	86.63	non-trainable
gWaveNet_5×5_nt	94.40	93.16	93.75	91.89	non-trainable
gWaveNet_7×7_nt	97.53	95.60	92.50	91.61	non-trainable
gWaveNet_9×9_nt	93.94	92.80	91.66	90.88	non-trainable
gWaveNet_3×3_pt	97.21	95.69	90.83	89.47	kapt
gWaveNet_5×5_pt	96.64	94.94	91.75	90.29	kapt
gWaveNet_7×7_pt	97.72	96.03	93.24	93.07	kapt
gWaveNet_9×9_pt	97.77	95.94	93.58	92.46	kapt
gWaveNet_noK	82.49	85.20	76.77	75.22	nckl
gWaveNet_16K_7×7	93.49	91.69	92.50	91.53	trainable-16
gWaveNet_64K_7×7	93.33	91.61	90.13	89.07	trainable-64

false positives and negatives in the confusion matrix possibly due to not learning the patterns of gravity waves, but the noise. This finding emphasizes the importance of using the proposed model architecture by integrating the checkerboard kernel for datasets dominated by noise. Though this approach performed poorly compared to other approaches that we proposed, it still performed better than VGG16 approaches in terms of learning and achieving better F1 scores.

Models with Multiple Kernels. We further expanded our experiment to evaluate the proposed models with stacked identical 7×7 kernels, either 64 or 16 times. We observed that both approaches performed almost similarly, with a slight improvement using the kernel 16 times. The training accuracies were over 93%, with F1 scores exceeding 91%. From the experiment results, we notice that *gWaveNet* model with the trainable custom kernel provided better performance than the models with the non-trainable custom kernel. This left us curious to investigate what changes are made to the custom kernel by the training process. For clarity, we derived the learned kernel, shown in Fig. 3b, at the first layer of the proposed trained model. If we compare this learned kernel with the proposed kernels (Fig. 2), the updated weights follow the same pattern as the initial kernel. This justifies that the custom kernel at the first layer has an impact on gaining better performance by the proposed model with trainable configuration, ensuring the continued relevance and applicability of the custom kernel.

In summary, the ablation studies showed significant improvements in detecting intricate patterns with both trainable and non-trainable configurations. The

method without a custom kernel (“gWaveNet_noK”) achieved an F1 score of 75.22, while integrating the custom kernel increased the F1 score to 91.89 (non-trainable) and 93.69 (trainable). Notably, the non-trainable configurations, apart from the trainable ones, outperformed all other state-of-the-art approaches in Table 2, demonstrating the custom kernel’s effectiveness and improved generalizability over a standard convolutional kernel. Our findings from the ablation studies led us to conclude that the proposed model is highly effective in detecting gravity waves amidst noise, achieving higher accuracies. These conclusions address Q2 and Q3 that are related to the model’s overall capability and ability to learn without denoising data.

Table 3. Comparison of models trained on 60% and 40% reduced datasets.

Methods	Accuracy			F1 Score
	Train	Validation	Test	
gWaveNet_9×9_60%	89.74	87.85	86.66	86.39
gWaveNet_7×7_60%	94.18	93.24	89.99	88.67
gWaveNet_5×5_60%	93.99	92.46	86.66	84.39
gWaveNet_3×3_60%	94.02	92.27	93.75	90.76
gabor_7×7_60%	99.86	93.44	91.04	93.50
sobel_3×3_60%	89.70	82.66	68.15	69.37
vgg16_3×3_60%	99.37	80.86	64.17	63.17
gWaveNet_9×9_40%	99.50	85.14	63.74	61.35
gWaveNet_7×7_40%	98.10	91.04	64.10	64.30
gWaveNet_5×5_40%	99.29	83.33	69.08	70.36
gWaveNet_3×3_40%	99.59	82.76	73.33	75.19
gabor_7×7_40%	99.93	92.66	90.88	92.05
sobel_3×3_40%	88.40	80.76	66.28	67.71
vgg16_3×3_40%	100.00	77.99	74.76	—

Model Evaluation with Reduced Amount of Data. We further evaluated model’s performance using a reduced dataset, which is 60% and 40% of the total data and the comparisons are depicted in Table 3. The results in the Table highlights that *gWaveNet* models trained with 60% of the data exhibit minimal fluctuations in both training and validation accuracy, as well as in the F1 score. Conversely, other models with the same data proportion show significant overfitting, indicating suboptimal performance. When assessing models with 40% of the data, all models, including *gWaveNet*, exhibited overfitting issues. This portion of ablation study shows, training the model with more than 50% of the data resulted in improved scores. In particular, VGG16 with a 3×3 trainable kernel failed to produce a meaningful F1 score due to its poor performance with the

limited dataset. These findings addresses the Q5 regarding the reduced amount of data.

Table 4. Comparison of mean and standard deviation between state-of-the-art and our proposed techniques.

Methods	Train_acc	Validation_acc	F1_Score
VGG16_3×3.t	86.98 ± 2.88	80.05 ± 1.59	67.66 ± 1.07
Gabor_7×7.t	91.83 ± 2.57	88.57 ± 3.50	76.29 ± 4.61
FFT_7×7.t	92.78 ± 0.66	90.74 ± 0.61	89.56 ± 0.88
<i>gWaveNet</i> _3×3.t	98.25 ± 0.54	96.64 ± 0.71	91.58 ± 0.81
<i>gWaveNet</i> _5×5.t	98.07 ± 0.18	96.31 ± 0.46	91.83 ± 0.99
<i>gWaveNet</i> _7×7.t	97.71 ± 0.26	95.85 ± 0.54	92.95 ± 0.48
<i>gWaveNet</i> _9×9.t	97.10 ± 0.12	94.99 ± 0.09	90.80 ± 0.52
<i>gWaveNet</i> _64k_7×7.t	92.61 ± 0.58	90.80 ± 0.76	88.89 ± 0.21
<i>gWaveNet</i> _16k_7×7.t	92.84 ± 0.46	90.53 ± 0.81	89.19 ± 0.60
<i>gWaveNet</i> _60%_3×3.t	93.89 ± 0.08	93.30 ± 1.61	88.11 ± 2.41
<i>gWaveNet</i> _60%_5×5.t	93.82 ± 0.33	92.41 ± 0.37	87.74 ± 2.55
<i>gWaveNet</i> _60%_7×7.t	95.57 ± 1.04	93.78 ± 0.42	89.62 ± 2.09
<i>gWaveNet</i> _60%_9×9.t	91.53 ± 1.35	89.30 ± 1.16	87.61 ± 1.03

Model Robustness Comparison. As the final step, we compare the mean and standard deviation of selected methods in Table 4. Our comparison includes VGG16 with a 3×3 trainable kernel (denoted as VGG16_3×3.t), ViT method, Gabor approach and FFT-based methods along with all *gWaveNet* methods with trainable layers. To ensure a thorough analysis of average performance and variability, each model was run five times. As we see from the table, the deviations for VGG16_3×3.t, are not much, however, the model shows an inadequate performance compared to others in terms of F1 score. The Gabor approach performs well in both accuracies and F1 scores, but the standard deviation is higher across all cases. Comparing the model with FFT denoised data with a 7×7 trainable layer(denoted as FFT_7×7.t) exhibits higher accuracies with lower deviations. However, the F1 score does not achieve as high as our proposed model, even when using the custom kernel layer. Despite this, the deviation in all categories is better in FFT_7×7.t compared to the above models. When comparing the performance of our models, *gWaveNet* with different kernel sizes, we observe minimal deviations, except few cases trained with 60% data, indicating the robustness of our proposed hybrid deep learning model with the checkerboard kernel integrated.

Limitations. From our experiments, we observed a performance drop when *gWaveNet* models are trained with less than 50% of the data, emphasizing the common requirement of ample data in deep learning model training. Additionally, in some instances, applying the convolutional kernel detected partial patterns or missed certain patterns (Sub-fig. 4c). We attribute this to the image preprocessing steps, indicating an area for improvement in future.

Discussions. Our proposed 7×7 and 5×5 kernels with trainable layers in *gWaveNet* demonstrated significant performance improvements over baseline methods, including Gabor, Sobel, or Laplacian filter-based kernels, as well as advanced models like ViT and Vgg16. Despite achieving high training accuracy by fewer approaches like, the Gabor, VGG16 was hampered with the overfitting issue or low F1 scores, indicating its limited effectiveness. However, *gWaveNet* performed well in those cases, highlighting the critical role of both kernel shape and coefficients in enhancing model performance. Notably, our model showed superiority in non-trainable configurations, consistently achieving higher F1 scores compared to state-of-the-art techniques. These results highlight the effectiveness and improved generalizability of our custom kernel over standard convolutional kernels, establishing our approach as a new benchmark in gravity wave detection.

6 Conclusions

Overall, our propose *gWaveNet* model demonstrated the ability to learn without data denoising, with higher accuracies and the versatility of the proposed kernel allows for integration into other approaches towards its generalizability. However, in future, we would like to experiment with various satellite data of similar patterns collected from multi-angular view and also would like to address the underlying physics behind the patterns of the gravity waves and the proposed kernel. Additionally, we are interested in addressing the challenges associated with the localization of gravity waves using bounding boxes, particularly with a diverse set of images capturing similar patterns in presence of noise and unwanted objects.

Acknowledgements. This work is supported by the NASA grant “Machine Learning based Automatic Detection of Upper Atmosphere Gravity Waves from NASA Satellite Images” (80NSSC22K0641).

References

1. Gravity wave data from VIIRS. <https://noaa-nesdis-n20-pds.s3.amazonaws.com/index.html#reprocessed/>
2. Hierarchical data format (HDF5). <https://www.neonscience.org/resources/learning-hub/tutorials/about-hdf5>. Accessed 30 June 2022

3. Trainable vs non-trainable. <https://copyprogramming.com/howto/what-is-the-definition-of-a-non-trainable-parameter>
4. Alexander, M., Holton, J.R.: A model study of zonal forcing in the equatorial stratosphere by convectively induced gravity waves. *J. Atmos. Sci.* **54**(3), 408–419 (1997)
5. Bazi, Y., Bashmal, L., Rahhal, M.M.A., Dayil, R.A., Ajlan, N.A.: Vision transformers for remote sensing image classification. *Remote Sens.* **13**(3), 516 (2021)
6. Chen, C.F.R., Fan, Q., Panda, R.: CrossViT: cross-attention multi-scale vision transformer for image classification. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 357–366 (2021)
7. Seaman, C.: Beginning to See the Light: an Introduction to VIIRS DNB and NCC. <https://rammb.cira.colostate.edu/projects/alaska/blog/index.php/unategorized/beginning-to-see-the-light-an-introduction-to-viirs-dnb-and-ncc/>. Accessed 2 Aug 2022
8. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
9. Fogel, I., Sagi, D.: Gabor filters as texture discriminator. *Biol. Cybern.* **61**(2), 103–113 (1989)
10. Fritts, D.C., Alexander, M.J.: Gravity wave dynamics and effects in the middle atmosphere. *Rev. Geophys.* **41**(1) (2003). <https://doi.org/10.1029/2001RG000106>, <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2001RG000106>
11. González, J.L., et al.: Atmospheric gravity wave detection using transfer learning techniques. In: *2022 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, pp. 128–137. IEEE (2022)
12. Hasan, R., Chu, C.: Noise in datasets: what are the impacts on classification performance? In: *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods* (2022)
13. Jovanovic, G.: About the nature of gravitational and gravity waves. *Phys. Astron. Int. J.* **2**(2), 75–77 (2018)
14. Kanopoulos, N., Vasanthavada, N., Baker, R.L.: Design of an image edge detection filter using the Sobel operator. *IEEE J. Solid-State Circuits* **23**(2), 358–367 (1988)
15. Khan, M.A., et al.: Flood-ResNet50: optimized deep learning model for efficient flood detection on edge device. In: *2023 International Conference on Machine Learning and Applications (ICMLA)*, pp. 512–519. IEEE (2023)
16. Ku, J., Harakeh, A., Waslander, S.L.: In defense of classical image processing: fast depth completion on the CPU. *CoRR* abs/1802.00036 (2018). <http://arxiv.org/abs/1802.00036>
17. Lai, C., et al.: Automatic extraction of gravity waves from all-sky airglow image based on machine learning. *Remote Sens.* **11**(13), 1516 (2019)
18. Lee, S.H., Chan, C.S., Mayo, S.J., Remagnino, P.: How deep learning extracts and learns leaf features for plant classification. *Pattern Recogn.* **71**, 1–13 (2017)
19. Li, X., Li, J., Williams, Z., Huang, X., Carroll, M., Wang, J.: Enhanced deep learning super-resolution for bathymetry data. In: *2022 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, pp. 49–57. IEEE (2022)
20. Mann, A.: To improve weather and climate models, researchers are chasing atmospheric gravity waves. *Earth Atmos. Planet. Sci.* **116**(39), 19218–19221 (2019)
21. Marcus, G.: Deep learning: a critical appraisal. arXiv preprint [arXiv:1801.00631](https://arxiv.org/abs/1801.00631) (2018)

22. Matsuoka, D., Watanabe, S., Sato, K., Kawazoe, S., Yu, W., Easterbrook, S.: Application of deep learning to estimate atmospheric gravity wave parameters in reanalysis data sets. *Geophys. Res. Lett.* **47**(19), e2020GL089436 (2020)
23. Moran, D.: What are gravity waves? https://www.weather.gov/source/zhu/ZHU_Training_Page/Miscellaneous/gravity_wave/gravity_wave.html. Accessed 17 June 2022
24. Mostafa, S.A.M., Wang, J., Holt, B., Wang, J.: YOLO based ocean Eddy localization with AWS SageMaker (2024). <https://arxiv.org/abs/2404.06744>
25. O'Mahony, N., et al.: Deep learning vs. traditional computer vision. In: *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC)*, vol. 11, pp. 128–144. Springer, Cham (2020)
26. Pinto, N., Barhomi, Y., Cox, D.D., DiCarlo, J.J.: Comparing state-of-the-art visual features on invariant object recognition tasks. In: *2011 IEEE workshop on Applications of computer vision (WACV)*, pp. 463–470. IEEE (2011)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
28. Sreekanth, V.S., Raghunath, K., Mishra, D.: Deep kernel dictionary learning for detection of wave breaking features in atmospheric gravity waves. *Comput. Geosci.* **176**, 105361 (2023)
29. Suresha, M., Raghukumar, D., Kuppa, S., Raghavendra, R.: MQ-KPCA: custom kernel PCA for classification of microscopic images. *J. Inst. Eng. (India): Ser. B* **103**(6), 2025–2033 (2022)
30. Tushar, Z.H., Ademakinwa, A., Wang, J., Zhang, Z., Purushotham, S.: CloudUNet: adapting UNet for retrieving cloud properties. In: *2024 IEEE International Geoscience and Remote Sensing Symposium (2024)*
31. Wang, X.: Laplacian operator-based edge detectors. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(5), 886–890 (2007)
32. Yousafzai, J., Cvetković, Z., Sollich, P.: Custom-designed SVM kernels for improved robustness of phoneme classification. In: *2009 17th European Signal Processing Conference*, pp. 1765–1769. IEEE (2009)
33. Zhang, H., Su, H.: 3D object recognition using kernel construction of phase wrapped images. In: *Third International Conference on Digital Image Processing (ICDIP 2011)*, vol. 8009, pp. 119–123. SPIE (2011)
34. Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. *Int. J. Comput. Vision* **73**, 213–238 (2007)
35. Zohuri, B., Moghaddam, M.: Deep learning limitations and flaws. *Mod. Approaches Mater. Sci* **2**, 241–250 (2020)



Neural-Code PIFu: High-Fidelity Single Image 3D Human Reconstruction via Neural Code Integration

Ruizhi Liu, Paolo Remagnino^(✉), and Hubert P. H. Shum

Department of Computer Science, Durham University, Durham, UK
{ruizhi.liu,paolo.remagnino,hubert.shum}@durham.ac.uk

Abstract. We introduce neural-code PIFu, a novel implicit function for 3D human reconstruction, leveraging neural codebooks, our approach learns recurrent patterns in the feature space and reuses them to improve current features. Many existing methods predict normal maps from image feature space which easily overlook non-trivial features. Moreover, neglecting global geometric correlations restricted the use of repetitive features to improve the expressive power of current features. In this work, we propose neural-code PIFu, a novel framework that enhances initial features by fusing them with neural codes that are learned from the input features and geometric prior. It also models the global geometric correlation to facilitate the use of neural codes. Extensive experiments demonstrate that our method outperforms state-of-the-art (SoTA) PIFu-based approaches by a large margin, and achieves comparable results to parametric-models-based methods without the use of auxiliary data.

Keywords: 3D Human Reconstruction · Deep Learning · Neural Code Integration

1 Introduction

The growing demand for realistic 3D human reconstruction has driven the development of diverse methodologies, serving as a crucial foundation for the meta-verse, and AR/VR industries. The main objective of 3D human reconstruction is to transform 2D features onto 3D surfaces that accurately represent the human in the RGB images. Early techniques [1] relied on dense view reconstruction to model intricate 3D human surfaces, but their reliance on sophisticated camera arrays made large-scale applications impractical. Recently, deep learning has revolutionized the field. Explicit representation is commonly used with deep learning to model human surfaces, early methods [1, 2] based on explicit surfaces cannot generate details for human surfaces. To address the issue, [3, 4] predicts 3D geometric offsets as clothing details. Despite promising results, explicit surface representations suffer from the inflexibility of modeling shape and struggle to replicate intricate garments such as dresses due to the significant divergence in shape from the human body.

Unlike explicit surfaces such as meshes, implicit surfaces can model arbitrary shapes and are not limited by the resolution of input data. Pixel-aligned implicit function first proposed in [5] has emerged as a promising approach in the field. PIFu [5] and PIFuHD [6] represent pioneering methods that employ implicit functions to reconstruct a human surface from a single RGB image directly. To reconstruct more detailed human surfaces, some methods [6, 8–10] attempt to predict normal maps from the image feature and use them as additional inputs to inform the models.

The main problems of many PIFu-based methods are twofold: (1) Predicting normal maps from image feature space has limited improvements on non-trivial details. Many current methods easily overlook subtle details in image features which are also underrepresented in the predicted normal maps. This limits the improvements provided by normal maps. (2) Neglecting global geometric correlations among query points hinders the exploitation of repetitive patterns. In this work, our proposed alternative method addresses these issues without relying on complex architectures or additional data assistance.

To address these challenges, we propose Neural-Code PIFu, an end-to-end trainable approach for 3D human reconstruction from a single image. Inspired by [11] which learns quality-dependent features using vector quantization. Our method effectively learns repetitive patterns via neural codebook learning modules and models the overall global geometric correlations via self-attention with positional encoding to facilitate the use of neural codes. We improve pixel-aligned features by fusing them with relevant neural codes locally and globally via context-aware latent fusion. Finally, We fully integrate local and global features by facilitating query points to sufficiently interact via neural code integration.

Our method outperforms SoTA quantitatively and qualitatively. We evaluate neural-code PIFu on Thuman2.0 [12] and BUFF dataset [7] as well as out-of-distribution images to show the generalization of the proposed method. Our method demonstrates promising results, outperforming PIFu-based SoTA by a noticeable margin, and achieving comparable results with parametric-model-informed methods (i.e. ICON [13] and ECON [14]). The out-of-distribution evaluation demonstrates that our method generalises well to unseen garments and poses with minimum artifacts.

Our main contributions are summarised as follows:

- We propose an end-to-end trainable approach named **Neural-Code PIFu** for 3D human reconstruction from a single image, which learns reoccurring patterns and stores them as neural codes. It also models the global geometric correlation among query points.
- We propose **Context-Aware Latent Fusion** to reuse learned neural codes to improve the expressiveness of the feature. This allows more geometric details even if they are blurry in the given latent space.
- We propose **Neural Code Integration** to facilitate the interaction between query points, and also encourage local and global features to be adequately integrated.

2 Related Works

In this section, we briefly review the development and the relevant domains of single-view 3D human reconstruction.

Explicit Reconstruction. An explicit surface can be described as a prescription of the precise location of the surface. The early methods represent a surface via voxels which discrete a 3D surface into a grid. This allows the explicit surface reconstruction to align with modern learning-based image processing methods [21–23], which can properly transfer 2D features to 3D surfaces without sacrificing a massive amount of consistency between 2D and 3D feature space. However, the aforementioned methods are highly sensitive to the resolution of input data, the computational consumption non-linearly increases with resolutions, which makes large-scale applications unfeasible. The point clouds, on the other hand, are computationally friendly in comparison to voxel representations [25, 26, 29]. Taking advantage of the properties of point clouds, recently proposed learning-based methods [13, 14, 26, 27] can encode a sophisticated surface into a compact and sparse latent space with the cost of a small amount of computational resources, but loss of information is inevitable when mapping from a dense to a sparse latent space, point clouds normally lack abundant geometric information. This results in a loss of details and over-smoothed surfaces. Our method proposes to reuse repetitive patterns in the learned image feature space to enhance the surface details without additional inputs.

Implicit Human Surface Reconstruction. Implicit representation could be considered as a function of the level set of the function [5]. This representation can be implemented as a multi-layer perceptron predicting occupancy field or SDF values, which indicate the probability of whether query points lie within the surface [5, 6]. To convey more useful information from 2D input data to 3D surface, recent methods predict occupancy field conditioned on pixel-aligned features [5, 6, 16]. These methods have achieved promising results. However, most of the methods suffer from over-smoothed reconstructed surfaces.

To address this challenge, recent methods either introduce auxiliary data as prior or strong constraints, such as normal maps and parametric models (e.g. SMPL [2] and SMPL-X [24]) or add more 3D supervisions to the models. However, these methods fail to fully explore the valuable 2D space, and useful information such as detailed features lost during the transition from 2D feature space to 3D space.

3 Methodology

Our objective is to extract a highly detailed 3D human surface from a single-view image using a novel implicit function. This function employs neural codebooks to capture repetitive patterns and preserve them as neural codes, leveraging them

to enhance the expressiveness of features. We argue that when image features are blurred or over-smoothed, normal maps struggle to capture details, consequently restricting the improvement offered by normal maps. Additionally, these methods fail to emphasize global geometric correlation. Some methods like [7] incorporate a global feature map derived from image feature space. However, it has limited improvement in global awareness of models, as features of each query point are still isolated. To alleviate these challenges, we propose a novel framework to improve initial features by fusing them with neural codes that are learned from the input features and geometric prior. As shown in Fig. 1, we propose a selective learning neural codebook that specifically preserves representative and reoccurring features as neural codes. We purposefully utilize these neural codes to enhance the expressiveness of the current features, achieving the addition of human surface details without the need for additional data assistance. Moreover, we introduce an extra branch for modeling global geometric correlations which facilitates the use of neural codes.

Preliminary. We start by detailing the background of the implicit function representation. An implicit function parameterizes a 3D surface as a level set of functions. Given a query point in the 3D space, an implicit function classifies the point as either inside or outside the surface. This is denoted as:

$$f(X) = \begin{cases} 1, & \text{if } X \text{ is inside the surface,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Pixel-aligned implicit function captures detailed features from RGB images. It predicts the occupancy field which represents the probability distribution of whether a query point is inside or outside the surface. The pixel-aligned implicit function is mathematically defined as:

$$f(F_c(x), \phi(X)) = s : s \in R, \quad (2)$$

where $F_c(x)$ is 2D image feature at position x which is the 2D projection of query point X , and $\phi(\cdot)$ is the depth value of point X in the relative camera coordinates. For more details, we refer readers to [5].

3.1 Neural-Code PIFu Representation

The inferior performance of current methods [6, 8–10] is attributed to the prediction of normal maps from image features and the absence of global geometric correlation. These methods reconstruct detailed human surfaces dependent on normal maps derived from image features. Although introducing normal maps has been proven useful in adding details, it does not address the core issues. Firstly, the improvement provided by normal maps is limited if the initial features are non-trivial in the image feature space. Secondly, the majority of PIFu-based methods [5, 6, 8, 9] fail to consider the global geometric correlation within query

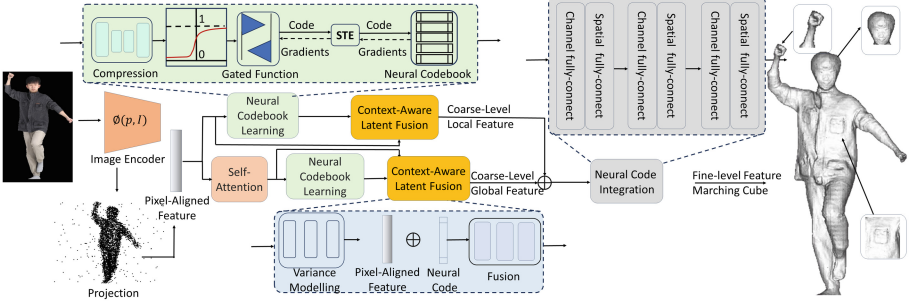


Fig. 1. The overview of our proposed method. The single view RGB image is fed into the image encoder, and the query points are projected to the image plane in order to obtain the pixel-aligned feature which is then used to obtain coarse-level local features and global features. The fine-level features are produced via neural codebook integration, which is used to predict the occupancy field for query points.

points. The global geometric correlation within the query points is essential for artifact reduction and completed human mesh.

To address the aforementioned challenges, we propose Neural-Code PIFu representation for human reconstruction, which possesses the ability to improve details based on input features and model global geometric correlations between query points. We adapt neural codebooks to learn representative and recurring features within the given latent space, selectively preserving them as neural codes. These neural codes are used as a complement for feature improvement, which allows the model not only to rely on image feature space but also on neural codebooks to acquire details. Moreover, modeling global geometric correlation informs the model with a global context, this allows a noticeable reduction of artifacts and efficient use of neural codes.

Our proposed model is mathematically represented as:

$$F_Q(x_c, F_g(f_l, f_g), \phi(X)) = s : s \in R, \quad (3)$$

where x_c is the input feature, F_g is the neural code integration which takes global and local features, denoted as f_g and f_l , as inputs. This module allows local and global features to be fully combined. Additionally, coarse global feature f_g and coarse local feature f_l are generated via the context-aware latent fusion described in Sect. 2.3. We apply self-attention with positional encoding to model the overall global correlation within all query points. This is denoted as follows:

$$SA(x_c) = softmax\left(\frac{Pos(Q)Pos(K)^T}{\sqrt{d_k}}\right) \cdot Pos(V). \quad (4)$$

Each query point not only contains its features but is also weighted based on all the other query points after this operation.

Neural Codebook Learning. We use neural codebooks to effectively capture and reuse representative features for reconstructing detailed human surfaces. The goal of the neural codebook learning module is to learn the latent distribution representing a shared collection of appearance and geometry within the given features. Given a pixel-aligned feature x_c of query points, we first extract the n most representative features in x_c via a softmax relaxation of nearest-neighbor:

$$z_i \leftarrow \frac{e^{-\|z_i - x_c\|_2}}{\sum_{n=1}^k e^{-\|z_n - x_c\|_2}}. \quad (5)$$

We adapt a straight-through-estimator (STE) to enable backpropagation through the neural codebook, which is vital for a learnable codebook. All neural codes are initialized with a standard Gaussian distribution $\mathcal{N} \sim (\mu, \sigma)$.

Gated Function. This function selectively preserves the neural codes of interest while discarding less relevant ones, ensuring the retention of the most distinct features captured in the input latent space. This step is crucial for reconstructing intricate surface details without introducing artifacts. The gated function is denoted as:

$$z_i \leftarrow \omega(\varphi(v_s - i^2/\lambda), T) \cdot z_i. \quad (6)$$

The gated function provides a hard decision boundary for neural code selection. The ω is a binarization function. T is a manually defined threshold, v_s is a scoring function that weights the inputs, and λ is a scaling hyperparameter based on the size of the neural codebook.

Discussions. Our method possesses better generalisation and flexibility in selecting features in comparison to existing methods like SuRS [15]. SuRS learns a prior difference between high- and low- resolution surfaces. This benefits reconstruction when the details in the image are non-trivial. Nevertheless, it is significantly constrained by the limitations imposed by the training distribution, demanding additional data and supervision. Additionally, it lacks the flexibility of applying learned prior knowledge to inform the model, which introduces a lot of artifacts. In contrast, our approach can selectively employ neural codes to enhance those blurred features. This contributes to artifact reduction and better generalisation.

3.2 Context-Aware Latent Fusion

Intuitively, details of the clothed human body, such as clothing wrinkles and facial contours, exhibit significant similarities. Existing works [6, 13–16] fail to take advantage of these similarities and reuse them to enhance non-trivial details.

Therefore, we propose context-aware latent fusion leveraging neural codebook learning modules for improvements of both local and global features. This module generates coarse-level local and global features by combining the input features with learned neural codes. This allows a better representative power to

improve the non-trivial details in the initial image space. This also enables the model to process out-of-distribution images. The module has two primary steps, variance modeling and latent fusion between neural codes and input features.

Variance Modelling. To ensure the neural codes are distinctive within the codebook, we follow [17] to further maximize the distance between learned neural codes by modeling the intra-variance between each code. The intra-variance is modeled using a convolutional neural network V which takes both neural code z_i and input feature x_c and outputs the variance perturbation:

$$z_i = z_i + \epsilon \cdot \frac{V(z_i, x_c)}{\|V(z_i, x_c)\|_2}. \quad (7)$$

It draws a clearer boundary within different neural codes and benefits the reduction of artifacts on the reconstructed surface, as ambiguity within the features deteriorates the uncertainty of points near the surface [18]. Introducing variance perturbation to neural codes eases the uncertainty.

Latent Fusion. It aims to generate local and global coarse-level features by merging the input latent with its relevant neural codes. There are two branches to separately process local and global features. We concatenate the input feature and its neural code and feed it into the local fusion module which is modeled as a residual MLP to obtain both coarse-level features.

3.3 Neural Code Integration

Deficiency in communication between query points is one of the weaknesses of previous PIFu-based models. Existing approaches [5, 6, 8] fail to facilitate sufficient interaction among the query points, and the local and global features are not adequately integrated.

Hence, we propose neural code integration to integrate coarse-level local and global features into fine-level features. The purpose of this module is to enable spatial-wise and channel-wise communication between both features.

We adapt MLP-mixer architecture [19] over the commonly used vision transformer for not only its simplicity but also for its comparable performance with a lighter computational burden. We modify the original architecture and directly apply it to the latent space. There are two steps within the neural code integration module: channel-wise mixing and location-wise mixing. In our case, the former enables communication within each feature of query points, the latter allows interaction within different query points. The neural code integration module is defined as:

$$f_{channel} = x_{g,l} + MLP_{channel}(x_{g,l}), \quad (8)$$

$$f_{spatial} = f_{channel} + MLP_{spatial}(f_{channel}), \quad (9)$$

where $MLP_{channel}$ and $MLP_{spatial}$ are responsible for channel-wise mixing and spatial mixing respectively, the $x_{g,l}$ is the concatenation of coarse-level local feature and global feature.

We use fine-level features to predict the occupancy field with an MLP surface classifier, and the reconstructed mesh is extracted following [20].

4 Experiments

To evaluate the performance of the proposed method, we conducted extensive experiments on two publicly accessible datasets that are widely accepted by the community, including Thuman2.0 [12], BUFF [7].

Datasets. Thuman2.0 [12] constitutes 524 high-resolution human meshes with rich details on the surface. We follow the split ratio mentioned in [15] to split the dataset into training and testing sets, which contain 402 and 122 meshes respectively. To evaluate the generalization of our proposed model, we conduct further experimentation on 143 human scans of both BUFF which no methods use for training.

Evaluation Metric. In our experiments, we leverage Chamfer Distance (CF) to measure the distance between the reconstructed surfaces and the ground truth surfaces. Average point-to-surface Euclidean distance (P2S) is applied to measure the distance from the vertices of the reconstructed surfaces to the ground truth surfaces. Lastly, we harness normal reprojection error to evaluate the projection consistency from input image. All metrics are measured in centimeters (cm).

Implementation Details. Our proposed model is trained with RGB image with the size of $(N_I \times N_I, N_I = 512)$. We follow the same rendering process used in PIFU [5] to generate images at every degree along the yaw axis for each human scan. The ground truth 3D points are sampled following the spatial sampling procedure mentioned in PIFU [5]. The input image is first encoded via a 2D convolutional neural network containing a stacked hourglass network which has been proven to possess better generalization for human-related estimation. The encoded continuous image features, which have the shape of $(W, H, C, W = 128, H = 128, C = 321)$. Pixel-aligned features then are obtained by projecting the query points to the image feature space. Pixel-aligned features are then passed to the neural codebook learning module to be decomposed and extract the most representative neural code. The coarse-level features are learnt through context-aware latent fusion which are learned via a 4-layer Multi-Layer Perceptron (MLP). A fine-level feature is produced via neural code integration which takes both global and local coarse-level features as input. Regarding the final occupancy prediction, we adapt a surface classifier formulated as a residual

MLP to classify the fine-level features. Once the occupancy values are obtained, we visualize it using [20].

Our model is trained on TITAN X GPU with 8 batches, and a learning rate of 0.0001 with decay. The model is optimized via Adam.

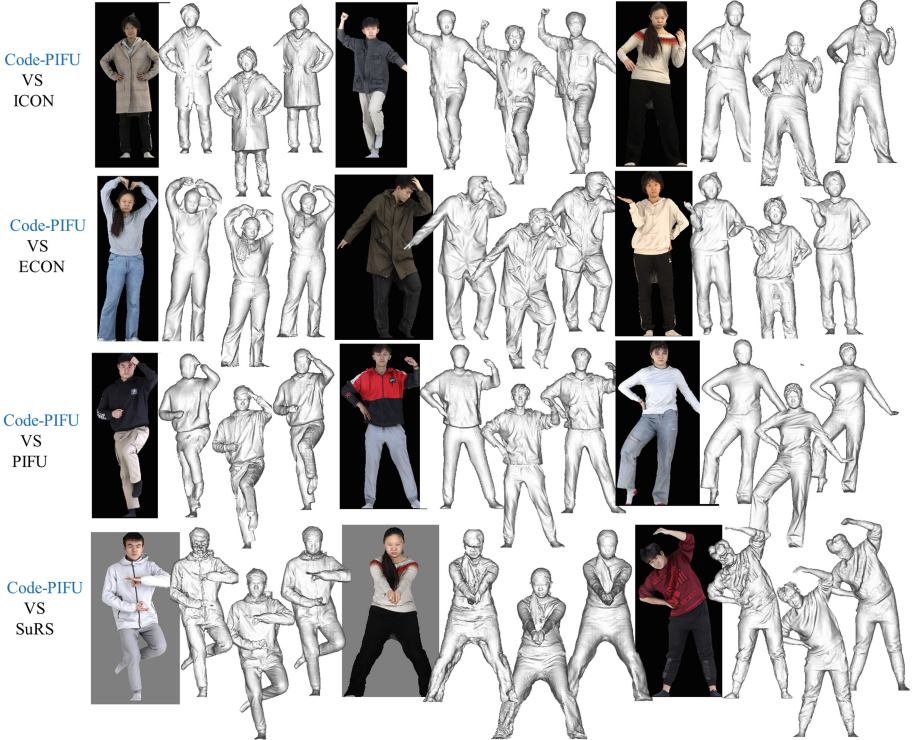


Fig. 2. The comparisons between our method and SoTA. From left to right are the results of the selected SoTA, the ground truth and our results.

4.1 Comparisons on SoTA

We compare our methods with state-of-the-arts methods: PIFu [5], PIFuHD [6], PaMIR [16], SuRS [15], ICON [13], ECON [14], GTA [28], D-IF

Table 1 shows quantitative results on Thuman2.0 [12], BUFF [7] dataset. Our method outperforms all PIFu-based SoTA with noticeable margins on the Thuman2.0 test dataset and BUFF dataset. As shown in Fig. 2, our proposed method produces more plausible meshes with minimum artifacts.

Our method outperforms parametric model-based methods ICON [13] and ECON [14] on Thuman2.0 dataset [12], and achieves comparable results on BUFF dataset [7]. This is largely attributed to the utilization of parametric

Table 1. The Quantitative results on Thuman 2.0 and BUFF dataset. The percentage shows the improvements of the proposed method in comparison to SoTA. Chamfer, P2S and Normal consistency evaluation: the smaller the better.

Models	THuman 2.0 Dataset			BUFF Dataset		
	Chamfer(↓)	P2S(↓)	Normal(↓)	Chamfer(↓)	P2S(↓)	Normal(↓)
PIFu	1.501 (↓50%)	1.523 (↓51%)	0.122 (↓37%)	1.781 (↓57%)	1.754 (↓55%)	0.142 (↓39%)
PIFuHD	1.372 (↓46%)	1.432 (↓49%)	0.124 (↓38%)	1.634 (↓54%)	1.671 (↓53%)	0.133 (↓35%)
PaMIR	1.713 (↓56%)	1.818 (↓60%)	0.134 (↓43%)	1.752 (↓57%)	1.872 (↓58%)	0.148 (↓42%)
SuRS	0.931 (↓20%)	1.151 (↓36%)	0.107 (↓28%)	1.532 (↓50%)	1.622 (↓51%)	0.136 (↓37%)
ICON	0.747 (↓0.5%)	0.735 (↓0.5%)	0.086 (↓10%)	0.832 (↓9%)	0.854 (↓9%)	0.087 (-)
ECON	0.748 (↓0.5%)	0.737 (↓0.5%)	0.079 (↓3%)	0.762 (↓0.5%)	0.732 (↓16%)	0.082 (↓5%)
GTA	0.755(↓0.6%)	0.742(↓0.6%)	0.082(↓12%)	0.822(↓12%)	0.841(↓15%)	0.085(↓0.5%)
D-IF	0.743(↓0.1%)	0.766(↓1.2%)	0.091(↓20%)	0.843(↓20%)	0.824(↓20%)	0.083(↓0.5%)
Ours	0.745	0.733	0.077	0.759	0.781	0.086

Table 2. The quantitative results on CAPE-NPE and CAPE-FP datasets

Models	CAPE-NFP			CAPE-FP		
	Chamfer	P2S	Normal	Chamfer	P2S	Normal
PIFu	2.559	2.340	0.093	1.756	1.625	0.077
PIFuHD	3.372	3.445	3.445	2.439	2.363	0.877
PaMIR	1.422	1.409	0.733	1.198	1.259	0.709
ICON	1.343	1.462	0.092	1.357	1.453	0.918
ECON	1.772	1.730	0.789	1.785	1.743	0.810
GTA	<u>1.021</u>	<u>0.937</u>	0.053	<u>0.786</u>	0.752	0.043
D-IF	1.123	1.087	0.068	0.996	0.805	0.060
Ours	0.893	0.812	0.072	0.901	<u>0.801</u>	<u>0.055</u>

Table 3. The ablation results on Thuman 2.0 and BUFF dataset. The percentage shows the performance improvement with or without key components. Chamfer, P2S, and Normal consistency evaluation: the smaller the better.

Modules	Thuman 2.0 Dataset			BUFF Dataset		
	Chamfer	P2S	Normal	Chamfer	P2S	Normal
w/ codebooks w/o fusion	0.774(↓4%)	0.764(↓4%)	0.082(↓6%)	0.787(↓4%)	0.791(↓1%)	0.090(↓4%)
w/o global codebook	0.762(↓2%)	0.754(↓3%)	0.081(↓5%)	0.797(↓5%)	0.798(↓2%)	0.089(↓3%)
w/o local codebook	0.780(↓5%)	0.773(↓5%)	0.084(↓8%)	0.815(↓7%)	0.824(↓5%)	0.092(↓7%)
w/o fusion	0.767(↓3%)	0.769(↓5%)	0.086(↓10%)	0.782(↓3%)	0.813(↓4%)	0.090(↓4%)
w/o Integration	0.782(↓5%)	0.791(↓7%)	0.088(↓13%)	0.812(↓7%)	0.833(↓6%)	0.092(↓7%)
w/ all modules	0.745	0.733	0.077	0.759	0.781	0.086

models in these methods for rendering human normal vector maps, which are subsequently employed to predict normal vector maps with clothing. The normal vectors obtained through parametric model rendering demonstrate greater

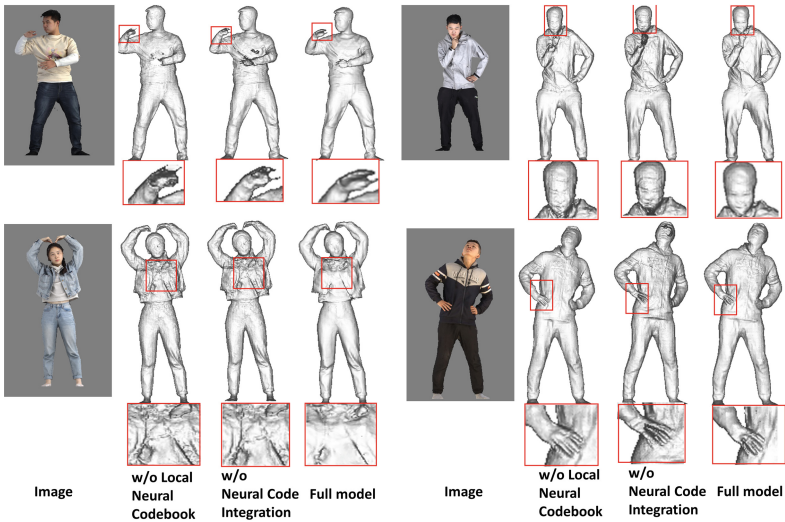
Table 4. The ablation study on parameter size and inference time on Nvidia TITAN X GPU.

Method	Parameter Size	Inference Time
w/ codebooks w/o fusion	17.6M	35.5 s
w/o global codebook	18.7M	36.3 s
w/o local codebook	18.9M	37.5 s
w/o fusion	18.5M	35.6 s
w/o Integration	17.5M	33.7 s
Full Model	20.6M	41.1 s

stability and accuracy compared to those predicted directly from image features. The discrepancy in performance is particularly noticeable on the BUFF dataset (Table 2).

4.2 Ablation Study

We evaluate our methods with a series of ablation studies to assess the key components contributing to the overall performance. Table 3 illustrates the performance with and without some significant modules of the proposed method. First, we evaluate the importance of two neural codebook learning modules. It is obvious that the performance dramatically deteriorates without the two neural codebook learning modules. Moreover, deployment of either neural codebook learning module boosts the performance, but the local codebook learning mod-

**Fig. 3.** The qualitative results of proposed model

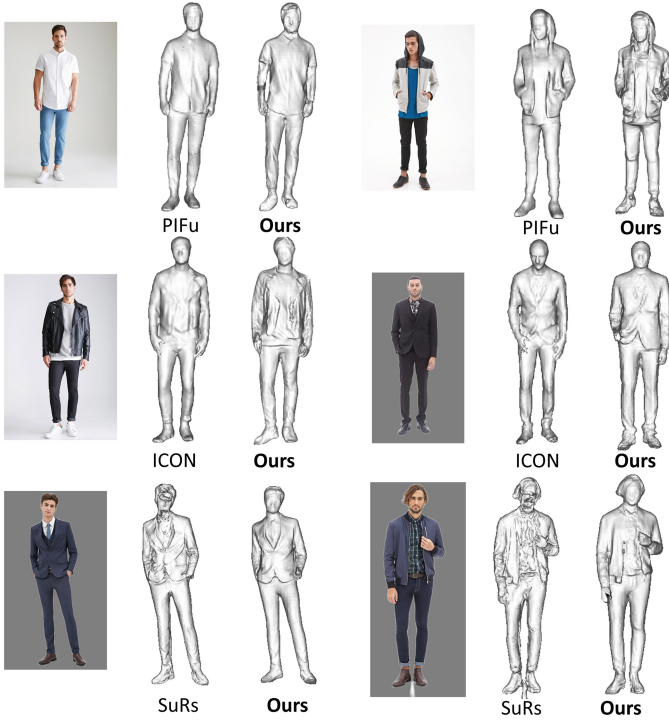


Fig. 4. The evaluation on out-of-distribution images

ule has a large impact on overall performance in comparison to its counterpart. Lastly, it is noticeable that without neural code integration results in the worst performance. Figure 3 shows the results of the qualitative ablation study. The local neural codebook contributes to the diversity of surface details such as fingers and facial details. It is noticeable that surfaces tend to suffer from local surface detail insufficiency without the local neural codebook. Similarly, neural code integration encourages local and global neural codes to be fully integrated, this is beneficial for surface details preservation.

We also conduct an ablation study on parameter size and inference time in Table 4 to further demonstrate the efficiency of the proposed model.

4.3 Out-of-Distribution Image Evaluation

We involve out-of-distribution image evaluation to further demonstrate the generality of our proposed models. As shown in Fig. 4, our model generalizes well on unseen images that are beyond the distribution of the training dataset. Our learned neural code book can generalize well to various unseen garment details and fashion poses without further training. Unlike SuRS [15] which are highly constrained by the distribution of training data, our method captures the most

frequently appeared patterns in the training data, and utilizes them to improve the expressiveness of input features beyond training distribution. We also capture more details than ICON which also predicts normal maps from image feature space.

5 Conclusion and Discussions

In conclusion, we propose a novel framework for 3D human reconstruction from a single image named neural-code PIFu which bridges the pixel-aligned features and its neural codes for better expressiveness. Our method predicts the coarse-level feature for both local and global contexts and applies two neural code books to learn the distinctive neural codes. The fine-level feature is produced via a neural code integration which considers the global geometric correlation of each feature, resulting in much detailed human surfaces.

Although our method surpasses SoTA in terms of generalisation, details capturing, and preservation for unseen clothing, our method shows weaknesses in reconstructing unseen poses which may result in broken meshes. Additionally, our method tends to recognize hair as details of garments, this frequently occurs when reconstructing females in fashion poses. Despite the promising performance, our model is trained on a fully synthetic dataset, and there is still a domain gap between the training data and the real-world data.

In future research, we will investigate combining uncertainty modeling, domain adaption, and diffusion models to alleviate the mentioned challenges, and also make effects into producing a real-world dataset for 3D human reconstruction.

References

1. Joo, H., Simon, T., Sheikh, Y.: Total capture: a 3D deformation model for tracking faces, hands, and bodies. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
2. Loper, M., et al.: SMPL: a skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries (2023)
3. Alldieck, T., et al.: Learning to reconstruct people in clothing from a single RGB camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
4. Lazova, V., Insafutdinov, E., Pons-Moll, G.: 360-degree textures of people in clothing from a single image. In: 2019 International Conference on 3D Vision (3DV), pp. 643–653. IEEE (2019)
5. Saito, S., et al.: PIFu: pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2304–2314 (2019)
6. Saito, S., Simon, T., Saragih, J., Joo, H.: PIFuHD: multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 84–93 (2020)

7. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
8. Chen, L., Jianghu, S., Luo, S.: TransPIFu: combining transformer and pixel-aligned implicit function for single-view clothed human reconstruction. *Comput. Graph.* **111**, 1–13 (2023)
9. Chan, K.Y., Lin, G., Zhao, H., Lin, W.: IntegratedPIFu: integrated pixel aligned implicit function for single-view human reconstruction. In: Avidan, S., Brostow, G., Cisse, M., Farinella, G.M., Hassner, T. (eds.) *Computer Vision – ECCV 2022*. ECCV 2022, vol. 13662, pp. 328–344. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20086-1_19
10. Chan, K., Lin, G., Zhao, H., Lin, W.: S-PIFu: integrating parametric human models with PIFu for single-view clothed human reconstruction. *Adv. Neural. Inf. Process. Syst.* **35**, 17373–17385 (2022)
11. Yang, Z., Dong, W., Li, X., Huang, M., Sun, Y., Shi, G.: Vector quantization with self-attention for quality independent representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24438–24448 (2023)
12. Yu, T., Zheng, Z., Guo, K., Liu, P., Dai, Q., Liu, Y.: Function4D: real-time human volumetric capture from very sparse consumer RGBD sensors. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, June 2021
13. Xiu, Y., Yang, J., Tzionas, D., Black, M.J.: ICON: implicit clothed humans obtained from normals. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13286–13296. IEEE (2022)
14. Xiu, Y., Yang, J., Cao, X., Tzionas, D., Black, M.J.: ECON: explicit clothed humans optimized via normal integration. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 512–523, June 2023
15. Pesavento, M., Volino, M., Hilton, A.: Super-resolution 3D human shape from a single low-resolution image. In: Avidan, S., Brostow, G., Cisse, M., Farinella, G.M., Hassner, T. (eds.) *Computer Vision – ECCV 2022*. ECCV 2022. LNCS, vol. 13662, pp. 447–4644. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20086-1_26
16. Zheng, Z., Yu, T., Liu, Y., Dai, Q.: PaMIR: parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(6), 3170–3184 (2021)
17. Wallingford, M., et al.: Neural radiance field codebooks. *arXiv preprint arXiv:2301.04101* (2023)
18. Yang, X.: D-IF: uncertainty-aware human digitization via implicit distribution field. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023)
19. Tolstikhin, I.O., et al.: MLP-Mixer: an all-MLP architecture for vision. *Adv. Neural. Inf. Process. Syst.* **34**, 24261–24272 (2021)
20. Lorensen, W.E., Cline, H.E.: Marching cubes: a high-resolution 3D surface construction algorithm. In: *Seminal Graphics: Pioneering Efforts that Shaped the Field*, pp. 347–353 (1998)
21. Ji, M., Gall, J., Zheng, H., Liu, Y., Fang, L.: SurfaceNet: an end-to-end 3D neural network for multiview stereopsis. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017)
22. Jimenez Rezende, D., Eslami, S.M., Mohamed, S., Battaglia, P., Jaderberg, M., Heess, N.: Unsupervised learning of 3D structure from images. In: *Advances in Neural Information Processing Systems*, vol. 29 (2016)

23. Kar, A., Häne, C., Malik, J.: Learning a multi-view stereo machine. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
24. Pavlakos, G., et al.: Expressive body capture: 3D hands, face, and body from a single image. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10975–10985 (2019)
25. Jiang, H., Cai, J., Zheng, J.: Skeleton-aware 3D human shape reconstruction from point clouds. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5431–5441 (2019)
26. Wang, J., et al.: Complete 3D human reconstruction from a single incomplete image. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023)
27. Lu, Y., et al.: 3D real-time human reconstruction with a single RGBD camera. *Appl. Intell.* **53**(8), 8735–8745 (2023)
28. Zhang, Z., et al.: Global-correlated 3D-decoupling transformer for clothed avatar reconstruction. In: *Advances in Neural Information Processing Systems*, vol. 36 (2024)
29. Zhao, X., et al.: Occupancy planes for single-view RGB-D human reconstruction. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3 (2023)



Sea-ShipNet: Detect Any Ship in SAR Images

Qinglin Zhang, Donghai Guan^(✉), Weiwei Yuan, and Mingqiang Wei

College of Computer Science and Technology, Nanjing University of Aeronautics and
Astronautics, Nanjing, China
{qinglinz, dhguan, yuanweiwei, mqwei}@nuaa.edu.cn

Abstract. In SAR image detection, small target ships are susceptible to interference from clutter and noise, making accurate classification and detection challenging. Despite significant progress in this field, there has been a lack of methods specifically adapting to the characteristics of small target ships dynamically. This limitation causes the existing dynamic detectors to equally allocate attention to small objects in simple and complex backgrounds, resulting in poor detection of small objects in complex backgrounds. To address this issue, we propose the SAR Dynamic Feature Adaptive Network (Sea-ShipNet). Firstly, we aggregate semantic information at the shallow feature level, significantly enhancing the feature contrast between small targets and the maritime background. Secondly, we propose a dynamic feature adaptive vector to guide image features to the detection head, paying more attention to small targets within complex backgrounds. We conduct comparative experiments with common methods on two SAR ship datasets, further demonstrating the superiority of our approach in detecting small target ships.

Keywords: SAR · Object detection · Small targets · Three-level feature fusion · Dynamic feature adaptation

1 Introduction

Synthetic Aperture Radar (SAR) has the ability of all-weather and all-day observation, which is of great significance in maritime surveillance, maritime security, national defense security and other fields. In SAR applications, ship detection especially small target ship detection plays a crucial role in maritime management and surveillance. Small target ships are easily disturbed by clutter and other noises due to their small size, which brings challenges to traditional detectors to achieve accurate detection.

The purpose of small target ship detection (STSD) in SAR images is to separate small target ships from complex background. The task faces two main challenges: **1) Small size:** Small target ships typically have very small dimensions, typically less than 25×25 pixels, and different small targets may exhibit different sizes. **2) Complex background:** Small target ships are susceptible

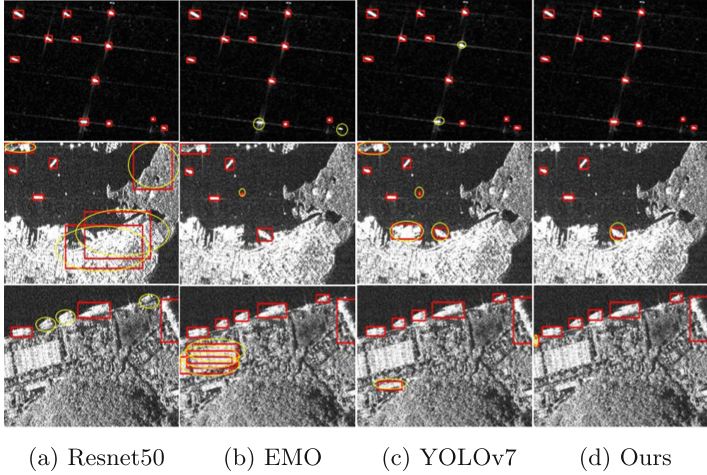


Fig. 1. The detection results of Sea-ShipNet on the SSDD dataset compared to SOTAs are shown in the detection images. The red boxes represent the predicted bounding boxes, and the yellow circles represent ships that are incorrectly classified. From the images, it can be observed that our Sea-ShipNet outperforms other methods in terms of detection accuracy. (Color figure online)

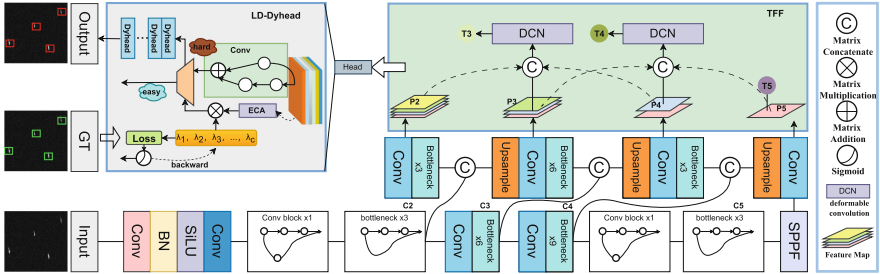


Fig. 2. Architecture of Sea-ShipNet. After obtaining feature maps $\{T3, T4, T5\}$ through TFF, they are input into LD-DyHead for dynamic detection. Here, $\{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_c\}$ is the dynamic feature adaptation vector, and its parameters are updated during the backpropagation stage.

to the interference of complex backgrounds, resulting in the blurring and loss of semantic information related to these small targets. As a result, state of the art (SOTA) detectors are susceptible to interference from complex backgrounds, leading to errors and missed detections in the results.

The traditional STSD method try to solve the feature ambiguity problem of small objects by using deeper network structures, multi-scale feature fusion, and designing detection heads to enhance the features of small objects. For example, some methods utilize pyramid feature [4, 6, 7, 13] fusion to enhance the representation ability of small objects by fusing multi-scale feature information. However, the original intention of these methods is to extract the semantic and spatial information of the object from the feature maps at different scales. In some cases,

more complex feature fusion operations in pursuit of richer feature information may lead to the loss of small target features. Designing a simple and effective feature fusion method that highlights the characteristics of small targets is an important research issue.

Additionally, numerous researchers have sought improvements from the detection head. Many works have proposed novel detection head [1, 8, 10–12] structures to better meet the detection requirements of small targets. Dai et al. [2] propose the dynamic detection head (Dyhead), which achieved significant performance in multi-scale object detection tasks through cascading multiple Dyhead blocks. However, it does not differentiate between complex and simple backgrounds; instead, it applies attention globally and does not prioritize small targets with complex backgrounds.

To address these issues, we propose a Dynamic Feature Adaptation Network specifically designed for SAR ship images. It consists of two main components: **1) Three-level Feature Fusion (TFF)**: We devise a feature fusion method tailored for small target ships, focusing on the fusion of shallow features. This method selectively aggregates semantic information for small target ships using deformable convolutions. **2) Lambda Dynamic detection head (LD-Dyhead)**: We propose a crucial learnable vector λ in the DyHead. It collaborates with the Efficient Channel Attention (ECA) [15] to guide different detection head algorithms for ship images with varying difficulty levels. This approach allows the model to truly “adapt” and pays more attention towards small targets with complex backgrounds.

Figure 1 shows the detection results of different network models in three distinct scenarios. We select several baseline models for comparison with our approach, and mark the correctly detected and falsely detected ships. Clearly, our method outperforms other baseline models by detecting more ships, while maintaining the lowest false detection rate. This indicates that our approach is indeed more effective at extracting small target ships from various complex backgrounds. Further experimental results will be presented in the experimental section.

In summary, our contributions mainly consist of the following three points:

- We propose the Three-level Feature Fusion, to enhance semantic information for small objects. By directly fusing shallow, intermediate, and deep feature maps, we obtain small object features rich in semantic information.
- Addressing the issue of conventional dynamic detection heads not adequately focusing on complex images, we introduce the dynamic feature adaptation vector λ . Additionally, we design a loss function tailored to λ to enable end-to-end training. The λ guides the dynamic detection head to pay more attention to complex image features during training.
- We conduct extensive ablation and comparative experiments to validate the effectiveness of our proposed method. Experimental results on two datasets demonstrate Sea-ShipNet’s performance in SAR ship detection, particularly its superiority in detecting small targets, outperforming current methods.

2 Methodology

2.1 Overview

Imagine how you would locate small target ships in a SAR image. You would compare the target with the background, focusing on the region with the highest contrast, and then assess whether it is indeed a ship. In the initial search, whether the background is simple or complex, you would need to expend attention to identify small target ships. As you become more adept, you might subconsciously employ a form of “muscle memory” in searching simpler images without expending additional attention on meticulous annotations. We propose Dynamic Feature Adaptation Network to imitate this human behavior as shown in Fig. 2. Firstly, we establish the TFF feature fusion network, enhancing the contrast between the target and the background. Subsequently, we propose a dynamic feature adaptation vector, emulating human “muscle memory,” allowing straightforward predictions for simple images and employing various self-attention functions for predicting complex image features.

2.2 Three-Level Feature Fusion

The structure of the three-level feature fusion module is shown in Fig. 2. C3, C4, C5 are obtained through FPN feature fusion, resulting in P3, P4, P5. P5 and P4 undergo max pooling and upsampling operations, respectively, to align spatially with P4. They are then concatenated along the channel dimension with the P4 feature map. Finally, the deformable convolution is applied to obtain the output feature map T. The above process can be summarized as:

$$P_i = \begin{cases} Conv(C_i), & i = 5 \\ Conv(Concat(P_{i+1}, C_i)), & i = 2, 3, 4 \end{cases} \quad (1)$$

$$T_i = DCN([Upsample(P_{i-1}), P_i, MaxPool(P_{i+1})]), i = 3, 4 \quad (2)$$

where DCN is deformable convolutional and $[\cdot]$ is matrix concatenation. T_i is output of the TFF.

To compare the effectiveness of FPN [6], PAN [7], BiFPN [13], and the proposed TFF in feature fusion for small ship detection, we select small target SAR ship images as inputs and visualized the fused feature maps using these four fusion methods, as shown in Fig. 3. FPN does not further perform three-level feature fusion on multi-scale feature maps, which is difficult to highlight the feature information of small targets. PAN and BiFPN use multi-step cascaded feature fusion, resulting in the feature information of small targets being submerged in massive features. In contrast, TFF uses a three-level feature fusion method without too many cascaded feature fusion operations, which effectively fuses multi-scale feature maps and significantly highlights the feature information of small objects. As can be seen from Fig. 3 (e), TFF clearly separates ship features from ocean features, demonstrating the significant improvement of TFF in detecting small ship targets.

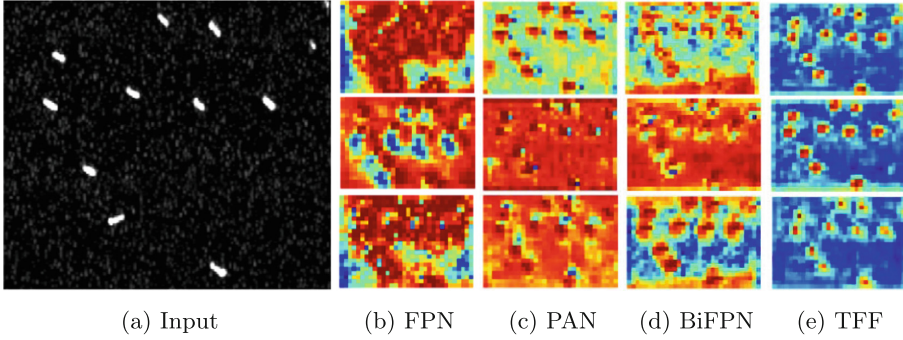


Fig. 3. Comparison between TFF and different feature fusion methods. (a) is a SAR ship image of SSDD dataset. (b), (c), (d) and (e) are the visualized feature maps of a obtained by the current feature fusion method, and we take the visualized feature maps of the first three channels (top-down) for display.

2.3 Lambda Dynamic Detection Head

A regular dynamic detection head [2] utilizes three attention functions: π_L , π_S , π_C , which operate on the dimensions of the feature map, namely L, S and C. Assuming the feature map $F \in R^{L \times S \times C}$, the output feature map of Dyhead is denoted as $W(F)$:

$$W(F) = \pi_C (\pi_S (\pi_L(F) \cdot F) \cdot F) \cdot F, \quad (3)$$

The attention function π_L is scale-aware attention:

$$\pi_L(F) \cdot F = \sigma(f(\frac{1}{SC} \sum_{S,C} F)) \cdot F, \quad (4)$$

where $f(\cdot)$ is a convolution operation with a 1×1 kernel and $\sigma(x) = \max(0, \min(1, \frac{x+1}{2}))$ is a hard-sigmoid function.

The attention function π_S is spatial-aware attention:

$$\pi_S(F) \cdot F = \frac{1}{L} \sum_{l=1}^L \sum_{k=1}^K w_{l,k} \cdot F(l; p_k + \Delta p_k; c) \cdot \Delta m_k, \quad (5)$$

where K is the number of sparsely sampled locations. Both Δp_k and Δm_k are learned from the intermediate layer features of the input feature map F .

The attention function π_C is task-aware attention:

$$\pi_C(F) \cdot F = \max(\alpha^1(F) \cdot F_c + \beta^1(F), \alpha^2(F) \cdot F_c + \beta^2(F)), \quad (6)$$

where F_c is the feature map on the c -th channel, and $[\alpha^1, \beta^1, \alpha^2, \beta^2]$ is the adaptive control threshold hyperparameters.

To dynamically adapt to features in both simple and complex images, we propose the $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_c\}$. It governs the selective entry of image features

into the Dyhead. Assuming the feature map $F \in R^{L \times S \times C}$, the output feature map is denoted as LD(F):

$$\theta = \frac{\lambda^T \cdot \Lambda(F)}{\|\lambda\|_2 \cdot \|\Lambda(F)\|_2}, \quad (7)$$

$$LD(F) = \theta \times W(F) + F. \quad (8)$$

In this context, Λ is ECA attention [15], which can map F into a c-dimensional vector used for matrix multiplication with λ , where \cdot is matrix multiplication. In our experiments, we find that the confidence loss for complex images tends to be larger than that for simple images. To address this characteristic, we define the loss function:

$$Loss = \alpha_{box} \times L_{box} + \alpha_{obj} \times \left(1 - \frac{1}{1 + e^{-\|\lambda\|_2}}\right) \times L_{obj}, \quad (9)$$

where $\alpha_{box}, \alpha_{obj}$ are balancing coefficients and $\|\cdot\|_2$ is L2 norm. L_{box} uses the CIoU [19] loss function.

The λ is a set of one-dimensional vectors, multi-dimensional image features are mapped to one-dimensional vectors by ECA attention formula, and matrix multiplication with λ is performed to obtain a numerical parameter θ . This parameter θ controls whether the image feature is detected by Dyhead or not. As θ tends to 1, LD(F) is approximately equal to $W(F) + F$, where $W(F)$ is the result of F passing through Dyhead to detect the head. LD(F) is approximately equal to F as θ tends to 0, indicating that F is not detected by the Dyhead head but by using a simple convolution.

The trend of the parameter θ is controlled by the loss function. According to the loss function, when the input image features are extremely complex, the confidence loss is usually large. In order to balance L_{box} and L_{obj} , the λ needs to take a large value to reduce the confidence loss. When the input image features are too simple, the confidence loss is usually small, and the λ needs to take a small value to balance L_{box} with L_{obj} .

In this way, λ can dynamically control the detection path through which the image passes, so as to adapt to image features of different complexity and achieve more flexible and effective detection.

3 Experiment

All experiments are conducted on Python version 3.7, PyTorch version 1.13 and NVIDIA GeForce RTX 4080 with a memory capacity of 16 GB.

3.1 Evaluation Metrics

Model performance is evaluated by precision, recall, and average precision (AP), which calculate how well the predictions overlap with the ground truth based on IoU. If the IoU exceeds the threshold, it is a true positive (TP), otherwise it is

Table 1. The results of comparative experiments between popular object detection models and Sea-ShipNet on the SSDD and SAR-ship datasets are as follows. The number of Dyhead blocks denoted by BN.

Model	Param	FLOPs	P	R	AP_{50}	AP_{75}	mAP	AP_S
SSDD Dataset								
Resnet50-l	33.8M	63.8G	84.4	77.0	81.9	52.5	47.8	47.3
YOLOv5-l	46.1M	107.6G	89.8	81.4	87.7	57.6	50.8	49.7
UniRepLKNet-n	82.3M	193.5G	90.2	78.3	86.6	48.4	47.4	46.1
EfficientViT-b5	60.1M	102.2G	88.2	81.4	85.9	55.4	50.1	47.5
FocalNet-s	70.9M	180.3G	86.3	81.1	85.6	51.4	47.6	46.3
YOLOv7-l	37.1M	104.5G	83.3	83.4	86.0	57.3	51.7	50.4
VanillaNet-s	55.1M	193.6G	87.0	81.2	86.1	51.1	48.4	46.4
EMO-5M	26.4M	153.6G	86.3	82.4	86.6	55.7	50.4	49.3
Sea-ShipNet-l(BN = 2)	54.7M	191.5G	91.8	82.0	88.3	57.3	52.1	50.9
Sea-ShipNet-l(BN = 3)	56.3M	192.8G	91.2	82.5	88.6	58.4	53.7	52.9
Sea-ShipNet-l(BN = 4)	58.1M	194.0G	92.6	84.7	89.4	62.7	54.2	53.6
SAR-Ship Dataset								
Resnet50-n	1.9M	3.6G	89.9	87.6	93.4	55.9	53.6	50.3
YOLOv5-n	1.8M	4.2G	89.7	90.0	94.3	59.0	55.3	52.0
UniRepLKNet-t	5.8M	11.0G	90.7	87.5	93.8	57.5	54.7	51.2
EfficientViT-b0	20.7M	39.0G	90.9	88.5	93.6	60.3	55.7	53.7
FocalNet-t	29.1M	74.3G	89.4	86.2	92.3	53.8	52.5	49.8
YOLOv7-n	2.3M	6.7G	90.6	90.2	94.6	63.7	57.7	55.1
VanillaNet-n	3.2M	11.0G	90.6	90.1	94.8	61.5	56.4	52.7
EMO-1M	2.0M	25.6G	90.0	89.6	94.7	63.5	57.5	54.2
Sea-ShipNet-n(BN = 2)	2.6M	10.2G	90.7	90.7	94.2	63.3	56.7	54.9
Sea-ShipNet-n(BN = 3)	2.8M	10.6G	90.6	91.1	95.2	64.4	58.4	56.7
Sea-ShipNet-n(BN = 4)	3.0M	10.9G	89.9	90.1	95.0	64.3	58.1	56.4

a false positive (FP). A labeled box with no corresponding prediction is a false negative (FN). Their calculation formula is as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (10)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (11)$$

$$\text{mAP} = \int_0^1 P(R) dR. \quad (12)$$

The AP value is positively correlated with the model performance. In this paper, the AP values for IoU = 0.5 and 0.75 are used, which are AP_{50} and AP_{75} ,

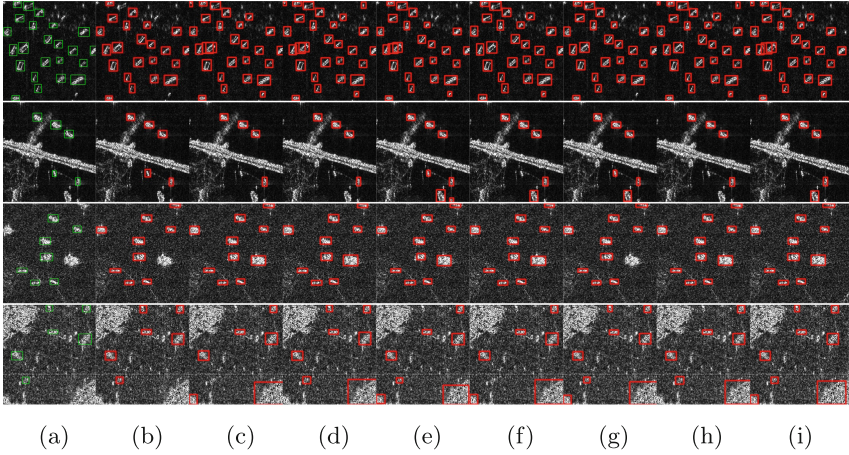


Fig. 4. The detection results of Sea-ShipNet on the SAR-Ship dataset compared to SOTAs are shown in the detection images. Sea-ShipNet has the lowest false and missed detection rates among the compared SOTAs. (a) is ground truth, (b) is the result of Sea-ShipNet, (c)–(i) respectively show the detection results of ResNet, YOLOv5, EfficientViT, FocalNet, YOLOv7, VanillaNet, and EMO.

respectively. mAP is the average AP value for IoU ranging from 0.5 to 0.95, and AP for small ship targets is denoted as AP_S .

3.2 Datasets

In the experiments, we use SSDD [5] and SAR-ship [16] datasets. The SSDD dataset contains 1,160 images of ships with 512×512 pixels, which are divided into training and validation sets at a 7:3 ratio. The SAR-ship dataset contains 39,729 images of 256×256 pixels, again divided into training and validation sets in a 7:3 ratio.

3.3 Comparison with the State of the Art

Table 1 is the comparative experimental results between Sea-ShipNet and current popular methods. We compared CNN-based methods including ResNet, YOLOv7 [14], UniRepLKNet [3], FocalNet [17], VanillaNet [17] and DETR-based methods including EfficientViT [9], EMO [18]. Sea-ShipNet outperforms the current SOTA detectors across various metrics, particularly excelling in the AP_S metric, achieving the highest values of 53.6 and 56.7 on the SSDD and SAR-Ship datasets, respectively. This strongly validates the outstanding performance of our method in detecting small target ships. Additionally, Sea-ShipNet achieves the best results in AP_{50} , indicating its superiority not only in detecting small targets but also in overall ship detection performance across different sizes. Figure 4 shows the comparison between our method and other methods.

Table 2. Compare the effect of TFF and LD-Dyhead respectively on SSDD dataset.

TFF	LD-Dyhead	AP_{50}	AP_{75}	mAP	AP_S
×	×	87.8	57.6	51.6	50.4
×	✓($BN = 2$)	86.8	62.2	53.4	51.7
×	✓($BN = 3$)	87.8	59.3	52.3	50.9
×	✓($BN = 4$)	88.3	62.0	54.0	52.2
✓	×	88.3	60.4	53.6	52.0

Table 3. Compared the proposed TFF method with the SOTA feature fusion methods on both datasets.

Model	AP_{50}	AP_{75}	mAP	AP_S
SSDD				
FPN	87.8	57.6	51.6	50.4
PAN	87.6	57.8	50.8	49.7
BiFPN	87.4	56.7	51.8	50.4
TFF(ours)	88.3	60.4	53.6	52.0
SAR-Ship				
FPN	94.3	58.7	55.1	52.3
PAN	94.3	59.9	55.7	52.0
BiFPN	94.4	60.2	55.6	52.1
TFF(ours)	94.7	60.9	56.1	52.8

3.4 Ablation Experiment

We conduct ablation experiments to validate the roles of the TFF and LD-Dyhead components in the model. The experimental results are shown in Table 2. Regarding the impact of TFF: when only the TFF component is added, there is a significant improvement in the AP_S small target evaluation metric. This improvement is attributed to TFF significantly enhancing the semantic features of ship targets during the feature fusion stage.

Effectiveness of TFF. To verify the superiority of our proposed TFF in small target detection, we compare it with mainstream feature fusion methods in current use, and the results are shown in Table 3. Among the three compared methods, FPN and PAN do not perform feature fusion on shallow feature maps, while BiFPN, due to its complex feature fusion operations, disrupts the semantic information of small targets. TFF, without complex feature fusion operations and specifically fusing shallow semantic features, exhibits superior performance in small target detection compared to these methods.

Effectiveness of λ (LD). To verify the effectiveness of the dynamic feature adaptation module, we conduct experiments on the YOLOv7 network, and the

Table 4. The results of the experiments are conducted on the SSDD dataset using the YOLOv7 network model are as follows. The BN is the number of Dyhead blocks. The bold numbers are the experimental results of YOLOv7 combined with LD-Dyhead, while the non-bold numbers represent the experimental results of YOLOv7 combined with Dyhead.

BN	AP_{50}	AP_{75}	mAP	AP_S
0(baseline)	86.1	57.6	51.5	50.5
2	87.7	56.4	51.6	49.7
	88.0	57.7	52.1	50.8
3	88.1	56.6	52.2	49.3
	87.9	58.1	52.9	51.2
4	87.2	57.9	52.3	49.8
	89.0	59.3	53.8	52.0

Table 5. Experiments with multi-scale feature maps

Layers	Param	SSDD					SAR-Ship				
		P	R	AP_{50}	mAP	AP_S	P	R	AP_{50}	mAP	AP_S
P3	37.6M	79.7	75.2	75.4	48.9	47.1	77.1	76.4	80.2	49.6	48.3
P3,P4	41.4M	81.6	79.3	79.4	51.8	51.5	86.1	84.7	88.5	52.7	51.9
P3,P4,P5	45.7M	87.9	83.9	88.3	53.6	52.0	89.1	88.6	94.7	56.1	52.8
P3,P4,P5,P6	47.9M	87.9	83.2	88.1	53.4	51.3	87.5	86.2	93.1	54.6	50.9
P3,P4,P5,P6,P7	49.4M	86.4	82.7	87.1	51.6	50.8	86.2	85.6	91.4	53.1	50.5

results are shown in Table 4. Before the addition of the LD module, Dyhead is unable to selectively process features from simple and complex images, resulting in poorer detection performance. After integrating the LD module, the model exhibit increased attention towards complex images, leading to an average improvement of 1.5 in various mAP metrics compared to Dyhead alone.

3.5 Comparative Experiments Within the Module

We conduct comparative experiments inside the TFF and LD-Dyhead modules. In the TFF module, we gradually add multi-scale feature maps starting from P3 feature map to verify the detection performance after using fewer or more multi-scale feature maps for three-level feature fusion. The experimental results are shown in Table 5. Within the LD-Dyhead module, we verified the ability of lambda to regulate L_{obj} by changing the balance coefficient α_{obj} of L_{obj} , and further verified the adaptability of lambda to simple and complex image features by comparing with the model without λ adjustment. The experimental results are shown in Table 6.

Table 6. Experiments on different a_{obj}

Parameter	Backbone	SSDD					SAR-Ship				
		P	R	AP ₅₀	mAP	AP _S	P	R	AP ₅₀	mAP	AP _S
$\alpha_{obj} = 0.1$	DarkNet53	86.2	82.1	86.7	51.2	51.1	84.8	83.7	89.2	53.5	50.8
$\alpha_{obj} = 0.3$	DarkNet53	87.9	84.1	87.5	53.3	52.0	87.2	85.4	91.5	54.1	51.6
$\alpha_{obj} = 0.5$	DarkNet53	88.1	83.5	86.9	51.7	51.9	88.6	87.9	94.1	55.6	52.4
$\alpha_{obj} = 0.7$	DarkNet53	88.4	84.6	88.3	54.0	52.2	87.7	87.4	92.9	55.2	52.0
$\alpha_{obj} = 1.0$	DarkNet53	87.4	84.1	87.7	52.9	51.7	88.1	87.6	93.4	54.8	51.1
$\alpha_{obj} = 10.0$	DarkNet53	86.8	81.4	86.2	51.5	50.9	85.4	83.2	90.6	52.7	50.2
$\alpha_{obj} = 10.0(\text{no } \lambda)$	DarkNet53	84.7	12.9	82.1	35.3	24.1	79.5	20.3	83.4	26.8	17.2

Table 7. Experiments of Sea-ShipNet with the generic model ResNet on detection and classification tasks.

Model	SSDD			ImageNet	
	P	R	AP ₅₀	Top.1	Top.5
ResNet50	84.4	77.0	81.9	78.1	93.3
Sea-ShipNet	92.6	84.7	89.4	81.2	94.7

Effectiveness of Using Fewer or More Layers. The experimental results are shown in Table 5. Starting from using only P3, all the metrics improve significantly as the number of layers increases, indicating that multi-scale feature maps can significantly improve detection performance. P3, P4 and P5 achieve the best detection results. However, when P6 is added, the indicators begin to decrease, and the decrease in AP_S is the most obvious by 0.7. This verifies that too many feature fusion operations mentioned in this paper will lead to the loss of semantic information of small objects, and then affect the detection performance. When P7 is added, all the indicators decrease significantly. In summary, P3, P4 and P5 achieve the best results, so we choose P3, P4 and P5 as the feature maps to be detected.

Effectiveness of Adaptive Capabilities of λ . The experimental results are shown in Table 6. On the SSDD dataset, the best results are achieved when $a_{obj} = 0.7$, indicating that λ can adapt well to the features of simple and complex images, and thus better balance L_{box} and L_{obj} . Similarly, on the SAR-Ship dataset, the best results are achieved when $a_{obj} = 0.5$. The difference in the optimal a_{obj} value between the two datasets is mainly due to the large difference in the image feature distribution between the two datasets. In order to verify the adaptability of λ to image features and the balance ability of L_{box} and L_{obj} , we remove λ when $a_{obj} = 10$ for experiments. The experimental results show that the model performs poorly when the λ is removed. This is because the balance coefficient a_{obj} of L_{obj} is too large, which makes the detector unable to balance the fluctuation of loss values caused by complex and simple features, and thus

Table 8. Experiments with different proportions of datasets.

Proportion	SSDD			SAR-Ship		
	P	R	AP ₅₀	P	R	AP ₅₀
10%	73.7	6.9	77.1	87.8	24.1	85.1
20%	83.2	27.6	86.9	87.4	62.9	87.0
50%	87.6	79.5	88.6	89.4	86.3	91.5
100%	92.6	84.7	89.4	90.7	91.1	95.2

unable to detect more objects. Although the precision P is at a reasonable value, this is because the value of a_{box} is reasonable, so that L_{box} can converge properly, so that the model can detect the object with a high degree of accuracy.

3.6 Comparative Experiments on Multi-task Learning

We compare the performance of Sea-ShipNet with the general model ResNet on multi-task learning. The experimental results are shown in Table 7. From the table, it can be seen that Sea-ShipNet performs significantly better than ResNet50 on SAR dataset (SSDD) in terms of precision (P), recall (R), and average precision (AP₅₀). Meanwhile, the Top-1 and Top-5 accuracy of Sea-ShipNet on ImageNet dataset is also slightly higher than ResNet50. These results show that the generic model can effectively utilize the custom information of SAR data through multi-task learning techniques, thereby improving its performance on multiple tasks. Specifically, Sea-ShipNet shows better adaptability and detection performance when dealing with both complex SAR data and standard image classification tasks. In summary, our experiments verify the effectiveness of the multi-task learning technique in combining SAR data custom information and demonstrate the comprehensive performance advantage of Sea-ShipNet over ResNet50 on multiple datasets.

3.7 Experiments with Different Proportions of Datasets

We conduct experiments with different proportions of datasets to show the advantage of the custom solution when there is less data and the performance improvement of large models when the amount of data increases. The experimental results are shown in Table 8. The experimental results show that the performance of the custom solution improves significantly on SSDD and SAR-Ship datasets when the proportion of data is low, such as 10% and 20%. This indicates that the custom model can better capture the features and provide higher detection precision and recall when there is less data. As the proportion of datasets increases 50% and 100%, the performance on both datasets improves, and the best performance is achieved especially at 100% data size. This shows that large models can gradually catch up with or even surpass the performance of custom models when the amount of data is sufficient.

4 Conclusion

SAR small target ship detection is an extremely challenging task. Due to the small size of the target ships and their susceptibility to background clutter, existing methods often struggle to effectively address both issues. To tackle these challenges, we propose the Dynamic Feature Adaptation Network, Sea-ShipNet: 1) enhancing the contrast between small target ships and background clutter to significantly highlight the features of the target ships; 2) employing multiple attention mechanisms to iteratively search for small target ships. Additionally, the proposed dynamic feature adaptation vector imitates a “muscle memory” filter, inspired by human filtering of simple background images, directing attention more towards complex background images. Our experimental results on two SAR ship datasets demonstrate that Sea-ShipNet effectively addresses the aforementioned challenges, achieving competitive performance across various metrics.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No. 62472220).

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: optimal speed and accuracy of object detection. *arXiv: abs/2004.10934* (2020). <https://api.semanticscholar.org/CorpusID:216080778>
2. Dai, X., et al.: Dynamic head: unifying object detection heads with attentions. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7369–7378 (2021). <https://api.semanticscholar.org/CorpusID:235436118>
3. Ding, X., et al.: UniRepLKNet: a universal perception large-kernel convnet for audio, video, point cloud, time-series and image recognition. *arXiv: abs/2311.15599* (2023). <https://api.semanticscholar.org/CorpusID:265456035>
4. Kong, T., Yao, A., Chen, Y., Sun, F.: HyperNet: towards accurate region proposal generation and joint object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 845–853 (2016). <https://api.semanticscholar.org/CorpusID:7087523>
5. Li, J., Qu, C., Shao, J.: Ship detection in SAR images based on an improved faster R-CNN. In: 2017 SAR in Big Data Era: Models, Methods and Applications (BIGSAR DATA), pp. 1–6 (2017). <https://api.semanticscholar.org/CorpusID:45651180>
6. Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936–944 (2016). <https://api.semanticscholar.org/CorpusID:10716717>
7. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018). <https://api.semanticscholar.org/CorpusID:3698141>

8. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *Computer Vision – ECCV 2016*. ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2, <https://api.semanticscholar.org/CorpusID:2141740>
9. Liu, X., Peng, H., Zheng, N., Yang, Y., Hu, H., Yuan, Y.: EfficientViT: memory efficient vision transformer with cascaded group attention. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14420–14430 (2023). <https://api.semanticscholar.org/CorpusID:258615318>
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788 (2016). <https://doi.org/10.1109/CVPR.2016.91>
11. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525 (2016). <https://api.semanticscholar.org/CorpusID:786357>
12. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. [arXiv: abs/1804.02767](https://arxiv.org/abs/1804.02767) (2018). <https://api.semanticscholar.org/CorpusID:4714433>
13. Tan, M., Pang, R., Le, Q.V.: EfficientDet: scalable and efficient object detection. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787 (2019). <https://api.semanticscholar.org/CorpusID:208175544>
14. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7464–7475 (2022). <https://api.semanticscholar.org/CorpusID:250311206>
15. Wang, Q., Wu, B., Zhu, P.F., Li, P., Zuo, W., Hu, Q.: ECA-Net: efficient channel attention for deep convolutional neural networks. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11531–11539 (2019). <https://api.semanticscholar.org/CorpusID:203902337>
16. Wang, Y., Wang, C., Zhang, H., Dong, Y., Wei, S.: A SAR dataset of ship detection for deep learning under complex backgrounds. *Remote Sens.* **11**, 765 (2019). <https://api.semanticscholar.org/CorpusID:92987019>
17. Yang, J., Li, C., Gao, J.: Focal modulation networks. [arXiv: abs/2203.11926](https://arxiv.org/abs/2203.11926) (2022). <https://api.semanticscholar.org/CorpusID:247596882>
18. Zhang, J., et al.: Rethinking mobile block for efficient attention-based models (2023). <https://api.semanticscholar.org/CorpusID:257921138>
19. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-IoU loss: faster and better learning for bounding box regression. In: *AAAI Conference on Artificial Intelligence* (2019). <https://api.semanticscholar.org/CorpusID:208158250>



Semantic Correlation Adaptation for Union-Set Multi-label Image Recognition

Xinyu Wang, Tao Pu, Dongyu Zhang^(✉), and Liang Lin

Sun Yat-Sen University, Guangzhou, China

wangxy675@mail2.sysu.edu.cn , zhangdy27@mail.sysu.edu.cn, linliang@ieee.org

Abstract. Existing multi-label classification works are confined to fixed target categories, requiring lots of effort in collecting complete labels. However, annotating all relevant labels for novel categories is impracticable. To cope with this challenge, we investigate a new task, union-set multi-label image recognition (US-MLR), which allows a varying label space for each image rather than a fixed one (see Fig. 1). Beyond complementing missing labels, it further requires aligning semantic correlations among different splits. In this work, we propose a novel semantic correlation adaptation (SCA) framework, which firstly explores semantic correlations within each domain and across different domains to complement missing labels and then performs semantic correlation co-adaptation to alleviate the correlation inconsistency due to the domain gap. Comprehensive experiments on a new US-MLR benchmark and multiple MLR benchmarks demonstrate the effectiveness of the proposed SCA framework.

Keywords: Union-Set Multi-label Image Recognition · Semantic Correlations · Co-Adaptation

1 Introduction

Multi-label image recognition (MLR), which aims to identify all semantic objects in the given scene, is a fundamental but practical task since daily images inherently contain multiple objects. In the last decade, lots of efforts [3, 4] were dedicated to facilitating this task as it supports plenty of downstream applications of image content understanding. However, earlier works predominantly confine a fixed target space, which requires complete labels for all relevant categories. When models are required to identify new categories, this issue compels the collection of an entirely new dataset, which is extremely labor-intensive [7, 11, 13], instead of reusing existing datasets. To alleviate this dilemma, we delve into a

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78122-3_14.

novel task, union-set multi-label image recognition (US-MLR), where each image possesses a distinct set of labels derived from potentially different category splits, as shown in Fig. 1.

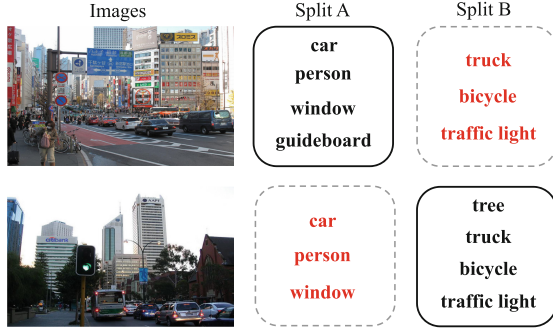


Fig. 1. Two examples in the US-MLR setting (the missing labels are highlighted in red). Due to the varying label space, each image merely contains annotations in the original category split while missing all labels in other splits. (Color figure online)

Due to the inconsistency of label space among different images, there are lots of missing labels in the US-MLR setting, leading traditional MLR models to poor performance. Fortunately, daily multi-label images contain rich and strong semantic correlations among different object categories, e.g., cars are likely to co-exist with roads. Hence, we propose exploring these semantic correlations to complement unknown labels. It is worth noting that current works in MLR-PL tasks [17, 19], where merely some labels are known while others are missing per image, are also focusing on complementing unknown labels in multi-label images. However, there is not only the absence of labels but also gaps between split domains in the US-MLR task. The latter prevents training MLR-PL models to solve this task because simply utilizing these correlations from different domains may introduce noise. Moreover, these semantic correlations obtained from distinct domains are incomplete, even biased, which can easily lead to model prediction bias. To address this challenge, we propose to align semantic correlations between different split domains, as shown in Fig. 2.

Based on these insights, we propose the semantic correlation adaptation framework (SCA) to achieve co-adaptation of semantic correlations in case the label space is variable. First, we use label co-occurrence and category similarity to mine potential semantic correlations within each domain and across different domains. However, it is not guaranteed that we can acquire all accurate correlations due to the existence of domain gaps, e.g., missing correlations among categories or all correlations related to one category, as shown in Fig. 2. Therefore, a co-adaptation module is designed to regularize the consistency between these semantic correlations.

Our contributions are summarized as follows: 1) We propose a new task, union-set multi-label image recognition (US-MLR), which allows a varying label

space for each image rather than a fixed one. 2) We design a novel semantic correlation adaptation framework (SCA) consisting of two modules: semantic correlation learning and semantic correlation co-adaptation. 3) We construct a new US-MLR benchmark for fair and comprehensive evaluations. To the best of our knowledge, this is the first attempt to construct such a unified evaluation benchmark.

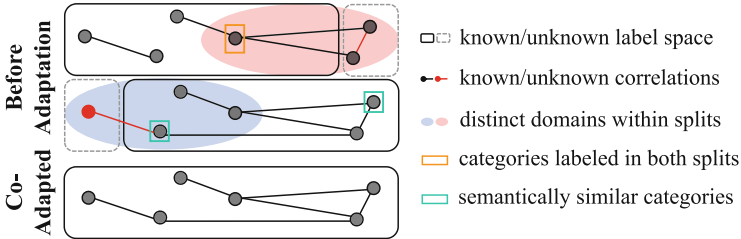


Fig. 2. Categories annotated in both splits or semantically similar ones allow correlating between the original split and the other split to learn new correlations (black dots and lines on the pink or blue background). However, gained correlations are incomplete, even biased (red dots and lines in gray dashed frames). After co-adapting, all semantic correlations are aligned to be consistent (all dots and lines are black). (Color figure online)

2 Related Work

With the rapid growth of search engines, recognizing multi-label images has received considerable attention in the computer vision community [20,21]. However, annotating a complete list of labels for every image is time-consuming and labor-intensive, making collecting large-scale and complete multi-label datasets less practical and scalable. To reduce the annotation cost, lots of efforts [2,15,17,19] are dedicated to training MLR models with partial labels, in which merely a few positive and negative labels are provided while others are unknown. Previous approaches [17,19] treat missing labels as negative or ignore them and consider the MLR-PL task as the multiple binary classification problem. However, these methods lose some data and even introduce noisy labels, resulting in poor performance. To overcome this problem, current works propose generating pseudo-labels for missing labels. Chen et al. [2] proposes to explore within- and cross-image semantic correlations to transfer knowledge of known labels to generate pseudo labels for the unknown. Pu et al. [15] proposes to exploit instance- and prototype-level semantic representation to complement unknown labels. However, these methods implicitly require consistent semantic correlations among categories and thus easily generate noisy labels when target categories are sampled from different domains.

Furthermore, there are various methodologies among current works. 1) Leveraging Prior: Ding et al. [5] propose exploring the structured semantic prior via a semantic prior prompter; 2) Estimating Noise: Li et al. [12] design a novel noise estimator that exploits label correlations without neither anchor points nor accurate fitting of noisy class posterior; 3) Tuning Loss: Kim et al. [10] provide an observation about memorization effect in the noisy multi-classsetting; 4) Exploring Correlation: Xia et al. [22] provide a high-level understanding of why label dependence helps distinguish the examples with clean/noisy multiple labels.

Distinct from these prior works, we further explore the semantic correlation inconsistency resulting from the varying label space of each image. Specifically, we propose extracting regularization patterns from common categories to align semantic correlations across distinct split domains.

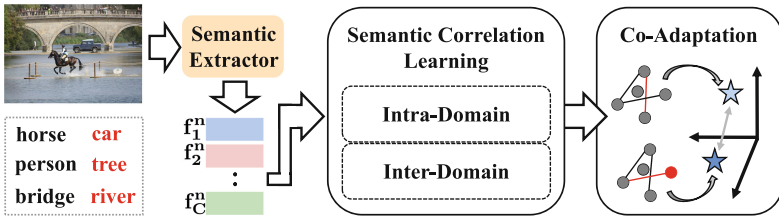


Fig. 3. An overall illustration of the proposed SCA framework that consists of a semantic correlation learning module and a semantic correlation co-adaptation module.

3 Semantic Correlation Adaptation

In this section, we introduce the proposed semantic correlation adaptation framework, as shown in Fig. 3.

3.1 Semantic Correlation Learning

As semantic correlations can effectively reduce vague predictions, recent works [8,18] use transformer-based prompt learning to learn label dependencies among categories. However, these methods require extensive multi-label samples and are limited to implicit correlation learning. To explicitly model correlations and facilitate performance comparisons with earlier works, we exploit graph convolutional networks (GCNs).

Initialization-Step. Here, we show two ways for initializing adjacency matrices, which are crucial for GCNs and enable propagated messages across the graph to explore semantic correlations.

Intra-domain. Previous existing multi-label recognition methods [3, 20, 21] have shown that co-occurrence in the semantic space coming from labels contain a wealth of prior knowledge to regularize model learning better visual representations. Therefore, we follow previous works to initialize the adjacency matrix based on labels coming from each split. Firstly, we use the matrix $G \in R^{C \times C}$ to record label co-occurrences, where C is the number of categories, G_{ij} is the number of co-occurrences of L_i and L_j , and G_{ii} is the number of appearances of L_i in the current split. Then, based on the matrix G , we get the adjacency matrix A as follows

$$A_{ij} = P(L_j|L_i) = G_{ij}/G_{ii}, \quad (1)$$

where $P(L_j|L_i)$ denotes the probability that when label L_i appears, label L_j also appears, so $P(L_j|L_i) \neq P(L_i|L_j)$.

Inter-domain. Although label co-occurrence can give accurate and complete semantic correlations within each domain, most label intersections between splits are too weak to uncover all complete semantic correlations across different splits. Also, gaps between splits make methods [2] of generating pseudo-labels easily introduce noise, resulting in poor performance. Fortunately, numerous web texts contain a lot of semantic knowledge that can give guidance on the more common distribution, so it is proposed to use category similarity to initialize the adjacency matrix. Specifically, for each category c , we firstly use the pre-trained GloVe model [14] to extract the category semantic embedding vector w_c . Then, the similarity of category semantic embedding vectors in the adjacency matrix A' can be calculated using cosine distance with the following formulation

$$A'_{ij} = \text{cosine}(w_i, w_j) = \frac{w_i \cdot w_j}{\|w_i\| \cdot \|w_j\|}, \quad (2)$$

where i and j are two categories belonging to different or same splits.

Learning-Step. With the help of the above two ways, we can use label co-occurrence and category similarity to correlate with the original split and obtain more semantic correlations for categories between different splits, which is beneficial for recognizing objects in the US-MLR task.

Specifically, we first construct corresponding GCNs based on adjacency matrices, which contain semantic correlation prior within and across splits by the following equation

$$f_{\text{Intra-SCL}}(X, A) = \left(A\sigma \left(AXW^{(0)} \right) W^{(1)} \right), \quad (3)$$

where $\sigma(\cdot)$ is a logistic sigmoid function. If replacing A with A' , we can get $f_{\text{Inter-SCL}}$. Secondly, we obtain the feature embedding \hat{o}_c^n and \bar{o}_c^n for each category c by following

$$\hat{o}_c^n = f_o \left(f_{\text{Intra-SCL}}(\mathbf{f}_c^n, A), \mathbf{f}_c^n \right), \quad (4)$$

where \mathbf{f}_c^n denotes the c -th category’s semantic representation vector described in detail in the supplementary material, $f_o(\cdot)$ is an output function that maps the concatenation into an output vector. If replacing $f_{Intra-SCL}$ with $f_{Inter-SCL}$, we can get \bar{o}_c^n . Finally, we can gain category prediction scores $\hat{S}^n = \{\hat{s}_1^n, \hat{s}_2^n, \dots, \hat{s}_C^n\}$ and $\bar{S}^n = \{\bar{s}_1^n, \bar{s}_2^n, \dots, \bar{s}_C^n\}$ for a training image I^n by the following formulation

$$\hat{s}_c^n = f_c(\tanh(\hat{o}_c^n)), \quad (5)$$

where $\tanh(\cdot)$ is a hyperbolic tangent function, and $f_c(\cdot)$ is a classifier that takes \hat{o}_c^n as input to predict the probability of objects belonging to category c . If replacing \hat{o}_c^n with \bar{o}_c^n , we can get \bar{S}^n .

3.2 Semantic Correlation Co-adaptation

As mentioned above, the proposed framework uses common labels or category semantic similarity to mine semantic correlations, which are implicitly existent within and across domains, accomplishing semantic correlation learning. However, there are two reasons for the inconsistency between learned semantic correlations and those originally contained within split domains: 1) When we try to correlate between the original split and the other split, we can’t make sure to learn all semantic correlations in the other split, so they are incomplete. Specifically, “car” and “person” will appear in the original split at the same time, and by the common label of “person”, we can get the co-occurrence of “person” and “bag”, as well as “person” and “bike”, which appear in the other split. However, it is not possible to obtain the co-occurrence of “bag” and “bike”, which is indicated by a red line in the first row of Fig. 2. 2) It is not sure that we can learn all correlations of categories belonging to the other split, but category correlations belonging to the original split are complete, which causes biased judgment. Specifically, “bag” and “bike” co-occur in a split, and by the fact that “motorbike” and “bike” have similar semantics, we can obtain the co-occurrence across splits for “bag” and “motorbike”. However, we cannot obtain correlations with “helmet” which is indicated by red dots and lines in the second row of Fig. 2.

To address this, we design a module to regularize consistency to realize effective semantic correlation co-adaptation. We use Kullback-Leibler (KL) Divergence to measure the difference between semantic correlations as follows

$$l(\hat{S}^n || \bar{S}^n) = \sum_{c=1}^C \hat{s}_c^n(I^n) \log \frac{\hat{s}_c^n(I^n)}{\bar{s}_c^n(I^n)}. \quad (6)$$

Compared to original correlations within each split, we need to not only check the completeness of learned semantic correlations by correlates but also prevent bias. Since KL Divergence is asymmetric, we measure the distance between \bar{S}^n and \hat{S}^n equally to get a total adaptation equation as follows

$$\mathcal{L}_{co} = \sum_{n=1}^N \left(l(\hat{S}^n || \bar{S}^n) + l(\bar{S}^n || \hat{S}^n) \right). \quad (7)$$

In this optimization, learned semantic correlations and correlations originally contained within splits are pulled closer in the feature space to realize effective alignment across split domains.

4 Optimization

Due to varying label spaces for different images in the novel US-MLR task, some labels contained in the original category split are completely absent in other splits. Inspired by [2], we transfer known labels by category-level feature similarity to supplement missing labels. For each category c , we use cosine distance to calculate the similarity of category features belonging to different images with the following formulation

$$s_c^{n,m} = \text{cosine}(\mathbf{f}_c^n, \mathbf{f}_c^m) = \frac{\mathbf{f}_c^n \cdot \mathbf{f}_c^m}{\|\mathbf{f}_c^n\| \cdot \|\mathbf{f}_c^m\|}. \quad (8)$$

We assume that the label of category c in the picture I^n is unknown and $\mathcal{D}_c = \{m | y_c^m = 1\}$ is the set of pictures that have labels on category c . In order to obtain a pseudo label representing the presence or absence of objects belonging to category c in the picture I^n , we firstly calculate the average similarity between the feature vector \mathbf{f}_c^n of the picture I^n and feature vectors of pictures in \mathcal{D}_c and then utilize the formulation as follows

$$\tilde{y}_c^n = \mathbf{1} \left[\left(\frac{1}{|\mathcal{D}_c|} \sum_{\{m \in \mathcal{D}_c\}} s_c^{n,m} \cdot y_c^m \right) \geq \theta \right]. \quad (9)$$

Here, $\mathbf{1}[\cdot]$ is the indicator function and θ is a threshold value. After combining known labels and pseudo labels, we can get $\tilde{Y}^n = \{\tilde{y}_1^n, \tilde{y}_2^n, \dots, \tilde{y}_C^n\}$.

To improve the feature similarity between images with the same positive labels, we use a pair loss for ranking tasks to supervise our network with the following formulations:

$$\mathcal{L}_{con} = \sum_{n=1}^N \sum_{m=1}^M \sum_{c=1}^C l_c^{n,m}, \quad (10)$$

$$l_c^{n,m} = \begin{cases} 1 - s_c^{n,m}, & y_c^n = 1, y_c^m = 1; \\ 1 + s_c^{n,m}, & \text{otherwise.} \end{cases} \quad (11)$$

Here, $s_c^{n,m}$ is the cosine similarity between feature vectors \mathbf{f}_c^n and \mathbf{f}_c^m . If both I_c^n and I_c^m have the same positive label c (i.e., $y_c^n = 1$ and $y_c^m = 1$), we aim to minimize the gap between \mathbf{f}_c^n and \mathbf{f}_c^m in the feature space by setting the loss to $1 - s_c^{n,m}$. Otherwise, we aim to maximize the distance between them in the feature space by setting the loss to $1 + s_c^{n,m}$.

We get the prediction scores \hat{S} by the Intra-SCL module and \bar{S} by the Inter-SCL module. Previous works used partial binary cross-entropy loss functions to evaluate the performance of classification modules, and we also use this function.

To calculate margins between prediction scores and the corresponding pseudo label (i.e., \hat{S} and \tilde{Y} , \bar{S} and \tilde{Y}), we use the objective functions defined by follows

$$l(\hat{S}^n, \tilde{Y}^n) = \frac{1}{\sum_{c=1}^C |\tilde{y}_c^n|} \sum_{c=1}^C [\mathbf{1}(\tilde{y}_c^n = 1) \log(\hat{s}_c^n) + \mathbf{1}(\tilde{y}_c^n = -1) \log(1 - \hat{s}_c^n)], \quad (12)$$

$$l(\bar{S}^n, \tilde{Y}^n) = \frac{1}{\sum_{c=1}^C |\tilde{y}_c^n|} \sum_{c=1}^C [\mathbf{1}(\tilde{y}_c^n = 1) \log(\bar{s}_c^n) + \mathbf{1}(\tilde{y}_c^n = -1) \log(1 - \bar{s}_c^n)]. \quad (13)$$

We define similar objective functions for prediction scores and the ground truth (i.e., \hat{S} and Y , \bar{S} and Y). The total classification loss is the sum of these losses over all images, formulated as following

$$\mathcal{L}_{cls} = \sum_{n=1}^N \left(l(\hat{S}^n, \tilde{Y}^n) + l(\bar{S}^n, \tilde{Y}^n) + l(\hat{S}^n, Y^n) + l(\bar{S}^n, Y^n) \right). \quad (14)$$

Finally, we sum the classification, co-adaptation, and contrastive losses of all images to obtain the final loss, formulated as

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{co} + \lambda \mathcal{L}_{con}. \quad (15)$$

Here, λ is a balance parameter that ensures the classification, co-adaptation, and contrastive losses have a comparable magnitude. In our experiments, we set λ to 0.05.

5 Experiments

5.1 Experimental Settings

Some fundamental settings are introduced in this subsection.

Dataset. Following earlier multi-label image recognition works, we conduct experiments on multiple datasets for fair comparisons, including MS-COCO [13], VG-200 [11], and Pascal VOC 2007 [7]. They consist of 122, 218, 108, 249, and 9,963 images, from 80, 200, and 20 classes, respectively.

Benchmark. To delve into potential challenges in reality, we combine MS-COCO and VG-200 to construct a large-scale US-MLR benchmark COCO&VG, which contains 231,536 images. Due to the lack of label information for unique categories, we select 38 categories that overlap among datasets as all categories for this benchmark. We set the intersecting proportion of splits to be between 20% and 30%, which corresponds to the median of intersection. Thus, each split

contains information on 23 categories, and we adopt a random strategy to allocate categories to two splits. For brevity, we name two splits COCO' and VG-200'. In addition, we randomly partition target categories and training images from traditional MLR benchmarks into two distinct splits to simulate the setting of the US-MLR task. Given that there may be intersections between splits, we set the proportion to 0%, 10%, 20%, ..., and 50%. Specifically, each split only contains information that belongs to itself and the intersecting part while missing all information belonging to others.

Evaluation Metric. For fair comparisons, we adopt the mean average precision (mAP) across all categories at varying proportions and calculate the average mAP across these proportions. Additionally, we utilize precision, recall, and F1 measures for more comprehensive comparisons, which are presented in the supplementary material.

5.2 Comparison with State-of-the-Art Algorithms

To evaluate the effectiveness of our SCA framework, we compare it with the following algorithms that are classified into two folds: 1) MLR algorithms: SSGRL [3], ML-GCN [4], KGGR [1]. These methods have achieved excellent results by exploring label correlations and semantic information on traditional MLR tasks, which are fully annotated. In our experiment, we use these methods by adopting partial BCE loss instead of BCE loss and keeping others unchanged. 2) MLR-PL algorithms: CL [6], ILRB [15], CST [2], IPRB [16]. Compared to MLR algorithms, these algorithms perform better on traditional MLR tasks and have gained impressive results on tasks that require more robustness for missing labels, e.g., the MLR-PL task. For a fair comparison, we use ResNet-101 [9] as a backbone to extract global feature maps for given images.

Table 1. On the new US-MLR benchmark, the mAP of categories only in the COCO' split (COCO') and only in the VG-200' split (VG-200'), as well as intersecting categories between two splits (Both). The best results are in bold.

Methods	Publication	COCO&VG		
		COCO'	VG-200'	Both
SSGRL	ICCV'19	77.8	69.3	85.7
ML-GCN	CVPR'19	78.3	70.0	85.7
KGGR	TPAMI'22	78.3	70.4	85.6
CL	CVPR'19	75.9	63.0	83.5
CST	AAAI'22	78.9	71.3	86.0
ILRB	AAAI'22	78.6	71.0	85.7
IPRB	ESWA'24	78.9	71.2	85.9
Ours	–	79.2	71.6	86.4

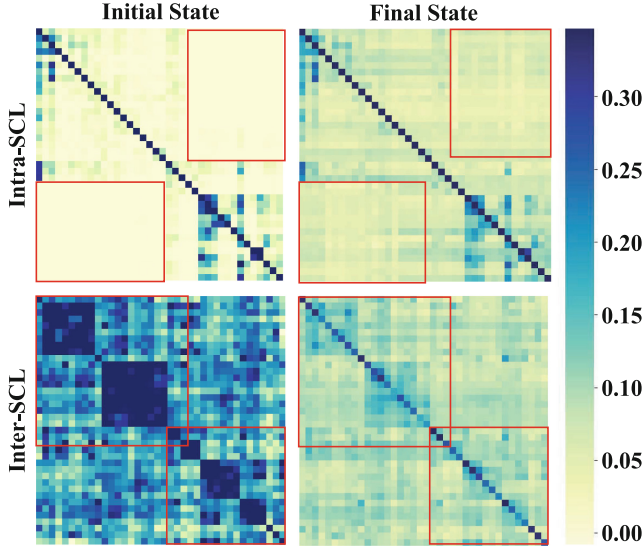


Fig. 4. Visualizations of the correlation matrices of the Intra-SCL (top) and Inter-SCL (bottom) modules in the initial (left) and final (right) states on the new US-MLR benchmark COCO&VG. Best viewed in color.

Performance on the New US-MLR Benchmark. We combine two existing datasets to construct a new benchmark named “COCO&VG” to mirror the settings of the union-set multi-label image recognition task. Different from previous benchmarks, it evaluates algorithms on unknown categories of each split to provide more comprehensive comparisons under the US-MLR setting. In experiments, we choose SSGRL as the baseline and compare our proposed method with existing excellent algorithms, as shown in Table 1. For the unknown categories of COCO’, previous algorithms gain the mAPs of 77.8%, 78.3%, 78.3%, 75.9%, 78.9%, 78.6%, and 78.9%. For the unknown categories of VG-200’, previous algorithms obtain the mAPs of 69.3%, 70.0%, 70.4%, 63.0%, 71.3%, 71.0%, and 71.2%. For the intersecting categories between two splits, previous algorithms gain the mAPs of 85.7%, 85.7%, 85.6%, 83.5%, 86.0%, 85.7%, and 85.9%. Our approach achieves the mAPs of 79.2%, 71.6%, and 86.4%, respectively. To further illustrate the effectiveness of our approach, we visualize the correlation matrices of Intra-SCL and Inter-SCL modules, as presented in Fig. 4. In the beginning, the matrix initialized by the intra-domain label co-occurrence has large blanks in the lower-left and upper-right corners; the matrix initialized by the inter-domain category similarity has inaccurate correlations in the upper-left and lower-right corners. At the end of the training, correlations across domains are mined, as guided by label co-occurrence and category similarity (the colors are darker and lighter, no longer white in the first row), and learned correlations are aligned with these originally contained within split domains (the color distributions in the second row tend to be consistent with the first line).

Table 2. Performance of current state-of-the-art MLR algorithms, MLR-PL algorithms, and our SCA framework under the setting of US-MLR on MS-COCO, VG-200, and Pascal VOC 2007 datasets. The best results are highlighted in bold.

Datasets	Methods	Publication	Intersecting Proportions						Avg.
			0%	10%	20%	30%	40%	50%	
COCO	SSGRL	ICCV'19	75.4	75.9	76.0	76.6	77.3	77.4	76.4
	ML-GCN	CVPR'19	75.2	76.1	76.8	77.3	77.8	78.0	76.9
	KGGR	TPAMI'22	77.7	78.1	78.3	78.7	78.9	79.0	78.5
	CL	CVPR'19	65.6	69.6	70.1	71.9	73.1	74.8	70.8
	CST	AAAI'22	77.6	78.2	78.4	78.6	79.1	79.2	78.5
	ILRB	AAAI'22	77.7	78.1	78.5	78.9	79.3	79.4	78.7
	IPRB	ESWA'24	77.8	78.2	78.4	79.0	79.2	79.3	78.7
	Ours	–	78.7	79.6	79.9	80.2	80.4	80.5	79.9
VG-200	SSGRL	ICCV'19	39.8	40.7	40.9	41.0	41.5	42.4	41.1
	ML-GCN	CVPR'19	39.9	40.7	41.5	41.6	41.9	42.0	41.3
	KGGR	TPAMI'22	43.3	43.4	43.6	43.7	43.9	44.1	43.7
	CL	CVPR'19	31.1	33.2	33.8	35.6	37.3	38.5	34.9
	CST	AAAI'22	43.4	43.5	43.6	43.8	44.1	44.3	43.8
	ILRB	AAAI'22	43.5	43.7	43.6	43.7	43.9	44.3	43.8
	IPRB	ESWA'24	42.9	43.2	43.3	43.4	43.5	44.1	43.4
	Ours	–	44.1	44.6	44.9	45.2	45.5	45.7	45.0
VOC	SSGRL	ICCV'19	91.4	91.7	91.9	92.1	92.3	92.5	92.0
	ML-GCN	CVPR'19	88.5	89.2	89.3	89.5	89.8	90.3	89.4
	KGGR	TPAMI'22	90.9	91.3	91.5	91.8	91.8	91.9	91.5
	CL	CVPR'19	89.0	91.0	91.5	91.6	92.2	92.2	91.3
	CST	AAAI'22	91.2	92.0	92.3	92.4	92.4	92.7	92.1
	ILRB	AAAI'22	91.9	92.1	92.2	92.4	92.3	92.7	92.3
	IPRB	ESWA'24	92.0	92.2	92.5	92.6	92.7	92.7	92.4
	Ours	–	92.4	92.6	92.9	93.0	93.0	93.1	92.8

Performance on the Traditional MLR Benchmarks. We experiment with the setting of the US-MLR task on traditional MLR benchmarks and discuss the impact of the intersection proportion between two splits, as shown in Table 2. On MS-COCO, our method improves the average mAP from 78.7% to 79.9% compared to the second-best. Notably, our method still gains obvious improvements in the case of extremely low intersecting proportions, where categories and images across splits are wildly different. For instance, the mAP improvements over the previous ILRB algorithm are 1.0% and 1.5% when the intersecting proportion is 0% and 10%, respectively. Compared with MS-COCO, VG-200 is a more challenging benchmark because of more categories. Our framework obtains the best performance for all different intersecting proportions. Specifically, it

obtains the mAPs of 44.1%, 44.6%, 44.9%, 45.2%, 45.5%, 45.7% on the settings of 0%-50% intersecting proportions, outperforming the second-best CST algorithm by 0.7%, 1.1%, 1.3%, 1.4%, 1.4%, 1.4%, respectively. Compared with the above two, Pascal VOC 2007 is more simple since it only covers 20 categories. Although previous algorithms achieve noticeable results, our SCA framework beats these methods for all intersecting proportions, as well as outperforms the excellent CST algorithm by 0.7% and the superior IPRB algorithm by 0.4% on the average mAP.

5.3 Ablation Study

Performance on the New US-MLR Benchmark. In this part, we conduct ablation experiments on MS-COCO and VG-200, which are traditional MLR benchmarks, as well as on the new US-MLR benchmark, which is constructed by MS-COCO and VG-200, to analyze the contribution of each module in our proposed SCA framework. More detailed results are provided in the supplementary materials.

Table 3. Comparisons of the average mAPs of the baseline SSGRL, our framework doing semantic correlation learning only from intra-domain (Intra-SCL) and only from inter-domain (Inter-SCL), our framework without the co-adaptation module (Ours w/o \mathcal{L}_{co}) and our framework (Ours) on COCO&VG, MS-COCO and VG-200.

Methods	Datasets		
	COCO&VG	MS-COCO	VG-200
SSGRL	76.8	76.4	41.1
Intra-SCL	77.5	78.2	42.6
Inter-SCL	77.7	77.7	43.0
Ours w/o \mathcal{L}_{co}	78.1	79.3	43.4
Ours	78.7	79.9	45.0

Analysis of the Intra- and Inter-domain SCL. To analyze the actual contribution of semantic correlation learning from intra- and inter-domain, we first compared the results of two SCL modules with the baseline on multiple benchmarks. As shown in Table 3, two SCL modules obtain the average mAPs of 77.5%, 77.7% on COCO&VG, 78.2%, 77.7% on MS-COCO, and 42.6%, 43.0% on VG-200. Learning semantic correlations only from intra- or inter-domain improves compared to the baseline. Furthermore, “Ours w/o \mathcal{L}_{co} ”, which has both Intra-SCL and Inter-SCL modules, improves the average mAPs of 0.6%, 1.1%, 0.8% than “Intra-SCL”, and 0.4%, 1.6%, 0.4% than “Inter-SCL”, respectively. It has been proved that the Inter-SCL module helps the other find missing correlations across different splits with category similarities, while the Intra-SCL module guides the other in learning the distributions of object co-occurrence within each split.

Analysis of the Co-adaptation Module. We propose a co-adaptation module to align the learned semantic correlations with the original semantic correlations contained within each split domain to regularize the consistency between two SCL modules. We design experiments with and without this co-adaptation module (namely, “Our” and “Ours w/o \mathcal{L}_{co} ”) to analyze the effectiveness of this module. In Table 3, “Ours w/o \mathcal{L}_{co} ”, which does semantic correlation learning from both intra- and inter-domain, already performs well, achieving the average mAPs of 78.1% on COCO&VG, 79.3% on MS-COCO, 43.4% on VG-200, and the average mAPs increases by another 0.6%, 0.6%, 1.6%, after adopting this module. It is evident that, with the help of this co-adaptation module, two SCL modules establish more precise category relationships after regularizing and aligning the consistency of semantic correlations between different split domains.

Performance on the Traditional MLR Benchmarks. In this part, we conduct ablation experiments on settings with intersecting proportions ranging from 0% to 50% to analyze the contribution of each module in our proposed SCA framework.

Table 4. Comparisons of mAP of the baseline SSGRL, our framework merely using the intra-domain semantic correlation learning module (Intra-SCL), our framework merely using the inter-domain semantic correlation learning module (Inter-SCL), our framework without the loss \mathcal{L}_{co} (Ours w/o \mathcal{L}_{co}) and our framework (Ours).

Dataset	Methods	Intersecting Proportions						Avg.
		0%	10%	20%	30%	40%	50%	
COCO	SSGRL	75.4	75.9	76.0	76.6	77.3	77.4	76.4
	Intra-SCL	77.3	77.9	78.0	78.4	78.7	78.9	78.2
	Inter-SCL	77.2	77.4	77.5	77.7	78.0	78.2	77.7
	Ours w/o \mathcal{L}_{co}	78.5	78.9	79.1	79.6	79.8	79.9	79.3
	Ours	78.7	79.6	79.9	80.2	80.4	80.5	79.9
VG-200	SSGRL	39.8	40.7	40.9	41.0	41.5	42.4	41.1
	Intra-SCL	42.1	42.4	42.5	42.7	42.8	42.9	42.6
	Inter-SCL	42.6	42.8	42.9	43.1	43.2	43.3	43.0
	Ours w/o \mathcal{L}_{co}	42.7	43.0	43.3	43.3	43.6	43.7	43.3
	Ours	44.1	44.6	44.9	45.2	45.5	45.7	45.0
VOC	SSGRL	91.4	91.7	91.9	92.1	92.3	92.5	92.0
	Intra-SCL	91.6	92.2	92.5	92.6	92.7	92.8	92.4
	Inter-SCL	91.7	91.9	92.0	92.4	92.5	92.5	92.2
	Ours w/o \mathcal{L}_{co}	92.3	92.5	92.6	92.7	92.9	92.9	92.7
	Ours	92.4	92.6	92.9	93.0	93.0	93.1	92.8

Analysis of the Intra-SCL Module. To analyze the actual contribution of the Intra-SCL module, we conduct experiments that merely use it (namely, Intra-SCL) compared with the baseline on multiple benchmarks. As shown in Table 4, it obtains the average mAPs of 78.2% on COCO, and 42.6% on VG-200, with an improvement of 1.8%, and 1.5%. Specifically, it obtains the mAPs of 77.3%, 77.9%, 78.0%, 78.4%, 78.7%, 78.9% on the settings of 0%–50% intersecting proportions, outperforming the baseline by 1.9%, 2.0%, 2.0%, 1.8%, 1.4%, 1.5%. Similar trends are also observed on VG-200 and Pascal VOC 2007.

Analysis of the Inter-SCL Module. To analyze the actual contribution of the Inter-SCL module, we conduct experiments that merely use it (namely, Inter-SCL) compared with the baseline on multiple benchmarks. As shown in Table 4, it obtains the average mAPs of 77.7% on COCO, and 43.0% on VG-200, with an improvement of 1.3%, and 1.9%. Specifically, it obtains the mAPs of 77.2%, 77.4%, 77.5%, 77.7%, 78.0%, 78.2% on the settings of 0%–50% intersecting proportions, outperforming the baseline by 1.8%, 1.5%, 1.5%, 1.1%, 0.7%, 0.8%. Similar trends are also observed on VG-200 and Pascal VOC 2007. “Ours w/o \mathcal{L}_{co} ”, which has two SCL modules, improves the average APs of 1.1% than “Intra-SCL” and 1.6% than “Inter-SCL”, respectively. It is proved that this Inter-SCL module helps the other find the missing correlations across different domains with category semantic similarities, while this Intra-SCL module guides the other to learn the distributions of object co-occurrence in this task.

Analysis of the Co-adaptation Module. We propose a co-adaptation module to align correlations to regularize the consistency between two SCL modules. We design experiments with and without this module (namely, Our and Ours w/o \mathcal{L}_{co}) to analyze the effectiveness of it. In Table 4, “Ours w/o \mathcal{L}_{co} ” already performs well with two SCL modules, and the average mAP increases by another 0.6% after adopting this module on COCO. A similar trend is also observed on VG-200 and Pascal VOC 2007. It is evident that, with the help of this module, two SCL modules construct more correct category relationships after organizing and aligning semantic correlations.

6 Conclusion

In this work, we introduce a challenging task, union-set multi-label image recognition, which allows a varying label space for different images rather than a fixed one. To solve this task, we propose a semantic correlation adaptation framework that explores intra- and inter-domain semantic correlations by label co-occurrence and category similarity. Besides, we design a co-adaptation module to resolve the inconsistency between semantic correlations, which is caused by gaps between split domains. To prove the effectiveness of our framework, we conduct extensive experiments on the new US-MLR benchmark and traditional MLR benchmarks.

Acknowledgements. This work is supported in part by Natural Science Foundation of Guangdong Province of China Under Grant No. 2024A1515011741, and partly supported by National Natural Science Foundation of China under Grant No. 62376292.

References

1. Chen, T., Lin, L., Chen, R., Hui, X., Wu, H.: Knowledge-guided multi-label few-shot learning for general image recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(3), 1371–1384 (2020)
2. Chen, T., Pu, T., Wu, H., Xie, Y., Lin, L.: Structured semantic transfer for multi-label recognition with partial labels. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 339–346 (2022)
3. Chen, T., Xu, M., Hui, X., Wu, H., Lin, L.: Learning semantic-specific graph representation for multi-label image recognition. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 522–531 (2019)
4. Chen, Z.M., Wei, X.S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5177–5186 (2019)
5. Ding, Z., et al.: Exploring structured semantic prior for multi label recognition with incomplete labels. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3398–3407 (2023)
6. Durand, T., Mehrasa, N., Mori, G.: Learning a deep convnet for multi-label classification with partial labels. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 647–657 (2019)
7. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. *Int. J. Comput. Vision* **88**, 303–338 (2010)
8. Guo, Z., Dong, B., Ji, Z., Bai, J., Guo, Y., Zuo, W.: Texts as images in prompt tuning for multi-label image recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2808–2817 (2023)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
10. Kim, Y., Kim, J.M., Akata, Z., Lee, J.: Large loss matters in weakly supervised multi-label classification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14156–14165 (2022)
11. Krishna, R., et al.: Visual genome: connecting language and vision using crowd-sourced dense image annotations. *Int. J. Comput. Vision* **123**(1), 32–73 (2017)
12. Li, S., Xia, X., Zhang, H., Zhan, Y., Ge, S., Liu, T.: Estimating noise transition matrix with label correlations for noisy multi-label learning. *Adv. Neural. Inf. Process. Syst.* **35**, 24184–24198 (2022)
13. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014*. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
14. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (2014)
15. Pu, T., Chen, T., Wu, H., Lin, L.: Semantic-aware representation blending for multi-label image recognition with partial labels. In: *Proceedings of the AAAI Conference on Artificial Intelligence* (2022)

16. Pu, T., Chen, T., Wu, H., Shi, Y., Yang, Z., Lin, L.: Dual-perspective semantic-aware representation blending for multi-label image recognition with partial labels. *Expert Syst. Appl.* **249**, 123526 (2024)
17. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 843–852 (2017)
18. Sun, X., Hu, P., Saenko, K.: DualCoOP: fast adaptation to multi-label recognition with limited annotations. *Adv. Neural. Inf. Process. Syst.* **35**, 30569–30582 (2022)
19. Wang, Q., Shen, B., Wang, S., Li, L., Si, L.: Binary codes embedding for fast image tagging with incomplete labels. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014*. LNCS, vol. 8690, pp. 425–439. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_28
20. Wang, Z., Chen, T., Li, G., Xu, R., Lin, L.: Multi-label image recognition by recurrently discovering attentional regions. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 464–472 (2017)
21. Wei, Y., et al.: HCP: a flexible CNN framework for multi-label image classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(9), 1901–1907 (2015)
22. Xia, X., et al.: Holistic label correction for noisy multi-label classification. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1483–1493 (2023)



FedSC: Federated Generalized Face Anti-Spoofing via Shuffled Codebook

Shuai Yang¹ , Mei Wang² , Weihong Deng¹ , and Jiani Hu¹ 

¹ Beijing University of Posts and Telecommunications, Beijing, China
jnhu@bupt.edu.cn

² Beijing Normal University, Beijing, China
wangmei1@bnu.edu.cn

Abstract. Domain generalization methods for Face Anti-Spoofing (FAS) have drawn increasing research attention. However, existing domain generalization (DG) methods usually require sharing data from varying source distributions, without considering privacy concerns. In this work, we propose the Federated Shuffle Codebook (FedSC), a federated FAS domain generalization method. Instead of sharing raw data, FedSC facilitates access to multi-source distributions by exchanging information within codebooks, ensuring privacy. Specifically, we first separate the images into style and content features. Style information is embedded into the style codebook through vector quantization during the training stage. Then the style codebooks are uploaded, shuffled and downloaded to transmit style information across domains. Each domain's source training data is diversified by the shuffled style codebook to achieve generalization. As the codebook represents the overall distribution rather than any specific image, FedSC offers both efficiency and privacy preservation. We have also devised a contrastive learning strategy to suppress the adverse effects of distribution differences on the liveness classification task. Theoretically, the established error boundaries of domain generalization provide robust support for our approach. Extensive experiments show that our proposed approach is effective and outperforms previous methods.

Keywords: Face Anti-Spoofing · Federated Learning · Domain Generalization

1 Introduction

With the rise of face presentation attacks (PAs), such as photo, video replay, or 3D facial masks, numerous Face Anti-Spoofing (FAS) methods have been proposed to tackle these challenges [34, 38, 39]. While most of these FAS methods demonstrate commendable performance in their specific domains, but may suffer dramatic degradation in cross-domain settings, primarily due to dataset biases [29], leading to diminished generalization capabilities in unseen domains.

Consequently, domain generalization methods (DG) are garnering increasing attention from researchers. Aimed at extracting representations that remain robust to distribution shifts, existing DG approaches typically necessitate access to multi-source distributions during the learning phase. For example, adversarial feature alignment methods [14] mandate training the domain discriminator using samples from various source datasets. Meta-learning based techniques [24] [26] leverage multi-source data from disparate distributions to formulate virtual training and testing domains within each minibatch. Methods based on style transfer [35] necessitate the use of style information from specific images to augment data by transitioning from one source domain to another. Without exception, these methods invariably require direct access to multi-source data, overlooking potential privacy concerns.

Federated learning (FL) [15] is a distributed and privacy-preserving machine learning technique, allowing for the training of models on distributed datasets while ensuring data remains localized. However, traditional federated learning algorithms, akin to FedAvg [22], primarily concentrate on enhancing model performance for internal clients, while neglecting model’s generalizability to unseen domains beyond the federation. In [19], the problem setting of FedDG was introduced, emphasizing that the challenge of the issue lies in enabling each client to access multi-source data distributions without compromising privacy. FedGPAD [27] is the first framework designed specifically to tackle the FedDG issue in the FAS area, employing a federated domain disentanglement strategy to extract domain-invariant features. Nevertheless, we contend that in FedGPAD, the approach of averaging updates for model parameters of domain-invariant part might lead to a loss of distribution information, thereby adversely affecting the generalization capability of the federated model.

Based on the aforementioned analysis, our motivation is to employ a method that transfers data distributions across domains in a privacy-preserving manner, replacing the averaging approach of FedAvg to address the FedDG issue in the FAS area. Inspired by the generative model VQ-VAE [30], we opt to introduce the concept of codebook. A codebook is composed of a fixed number of embedding vectors, forming a latent embedding space. We employ the codebook to convert features with continuous representations into discrete representations using vector quantization. Concurrently, embedding vectors in the codebook undergo updates, thereby storing the information about local data distribution. Since the embedding vectors in the codebook represent overall distribution information from various domains rather than any specific sample, transferring the codebook as a carrier for distribution information, as opposed to raw data, achieves both efficiency and privacy protection.

In light of the discussed concepts, we introduce the Federated Shuffle Codebook (FedSC), a federated FAS domain generalization method. Specifically, we use a feature extractor to derive style and content features from images. These features are vector-quantized with their respective codebooks, yielding discrete representations. These are then re-merged to produce comprehensive features for subsequent classification task. Notably, the gradient backpropagation from the

classification task enhances the liveness-related information in the codebook. To transfer style information cross domains, the style codebooks are shuffled randomly at the server and sent back. During the training phase, each data center’s local model leverages this stylistic information from other domains to diversify local data, achieving domain generalization. We have also devised a contrastive learning strategy to suppress the adverse effects of distribution differences among the codebooks.

Our contributions can be summarized as follows:

- We introduce a novel architecture, FedSC, which transfers distribution information across domains in a privacy-preserving manner by shuffling the codebook, addressing the FedDG issue in the FAS area.
- We developed a contrastive learning approach to minimize the differences between features from diverse style codebooks, targeting reduced distribution discrepancies. Additionally, we offer a theoretical foundation for our methodology.
- We conducted experiments on the generalization capabilities of federated models across four datasets. The results indicate that our method outperforms previous approaches, proving the efficacy of our method.

2 Related Work

2.1 Face Anti-Spoofing

Initially, Face Anti-Spoofing (FAS) used hand-crafted features like LBP and SIFT, but has since evolved with deep learning, employing techniques such as FCN for facial feature extraction and auxiliary tasks with depth, reflection maps, and rPPG for improved detection [2, 9, 10, 18, 23, 37]. Innovations like Central Differential Convolution have enhanced feature extraction by integrating intensity and gradient information [39]. Recent FAS research focuses on domain generalization, developing methods to distinguish genuine and spoof features across varying domains [14, 25]. Techniques like SSAN leverage style transfer to suppress domain-specific features, promoting generalization [35]. To address the unpredictability of new spoofing attacks, simulated attack images are now used for training, avoiding reliance on real-world samples [33].

2.2 Domain Generalization

Domain Generalization (DG) aims to enhance model generalization across unseen domains by leveraging multiple source domains, without using target domain data. Many DG techniques focus on domain alignment to achieve domain-invariant features by methods such as reducing KL Divergence, minimizing Maximum Mean Discrepancy (MMD), and employing Domain-Adversarial Learning [11, 16, 17, 43]. An approach in [44] improves feature generalization with multi-grained alignment and domain attention, whereas SA-FAS [28] aligns transitions for FAS using Invariant Risk Minimization.

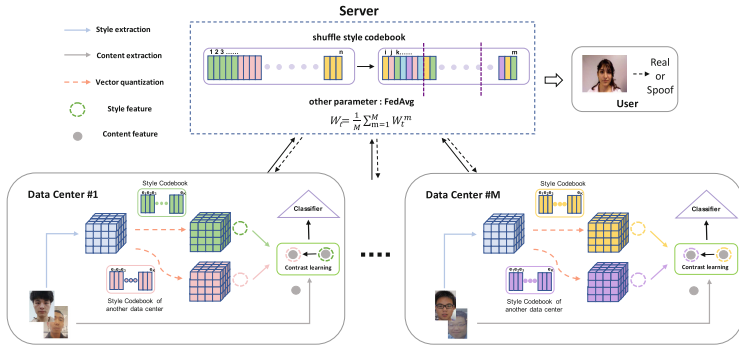


Fig. 1. The overall architecture of our method. Images undergo a feature extractor to derive style and content features, and these features are then quantized into discrete representations. The style assembly layers output the combined feature for classification and contrastive learning. Each data center downloads a collaboratively trained model from the server for local training. The server aggregates the updates using FedAvg. To facilitate domain information transfer, the style codebook is connected and shuffled. Users can download this global model from the server to detect various face presentation attacks.

Data augmentation, especially through synthetic attack images [33] and style transfer techniques like AdaIN [13], serves as another strategy for DG, aiming to prepare models for novel attack types or domains by simulating diverse conditions.

2.3 Federated Domain Generalization

Federated Learning (FL) is a collaborative, privacy-focused machine learning approach with multiple clients and a central server. Despite limited DG research within FL, one study investigates cross-domain transfer through amplitude information and meta-learning for the FedDG challenge, facing privacy risks [19]. Another proposes adjusting aggregation weights for better out-of-domain generalization, differing from our method [41]. A related strategy involves broadcasting style information for cross-domain transfer, beneficial for data augmentation but not suited for FAS, where style impacts classification [5, 35]. Our method uses vector quantization and codebooks, balancing privacy and efficiency.

3 Method

In this section, we will first introduce our model as depicted in Fig. 1. Following this, we will elaborate on the theoretical error bound, and the motivation behind the development of our proposed methodology. We summarize the steps of FedSC in Algorithm 1.

Algorithm 1. FedSC

Input: Number of data centers M , number of iterations T , initial weights \mathbf{w}^0 , data center indices $i \in \{1, 2, \dots, M\}$, λ_1, λ_2

- 1: **for** $t = 0$ to T **do**
- 2: Server sends \mathbf{w}^t and style codebooks $\{C_i\}_{i=1}^M$ to all data centers
- 3: **for** each data center i **do**
- 4: Data center i updates its local parameter via computing the loss function $L_{\text{all}} = L_{\text{cls}} + \lambda_1 \cdot L_e + \lambda_2 \cdot L_{\text{contra}}$
- 5: Data center i sends \mathbf{w}_i^t and C_i back to the Server
- 6: **end for**
- 7: Server shuffles the style codebooks $\{C_i\}_{i=1}^M$
- 8: Server aggregates the else weights as $\mathbf{w}^{t+1} = \frac{1}{M} \sum_{i=1}^M \mathbf{w}_i^{t+1}$
- 9: **end for**
- 10: Output the final weights \mathbf{w}^T

3.1 Problem Formulation

Suppose that M data centers $\{\mathcal{D}^1, \mathcal{D}^2, \dots, \mathcal{D}^M\}$ collect their own datasets, which are sampled from a joint image and label space $(\mathcal{X}, \mathcal{Y})$. A sample is represented as (x, y) with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

The federated learning paradigm involves communication between a central server and the M data centers. At each federated round t , each data center downloads the global model from the server and then trains the model using its local data. In the training phase, each sample (x, y) is input into the feature encoder E . As outlined in [35], content features $f_c(x)$ and style features $f_s(x)$ are extracted. We define the codebook $e \in R^{(K \times D)}$, where K is the size of the codebook and D denotes the dimension of each embedding vector e_k within it. The content codebook and style codebook utilize the same settings of K and D .

After model training is completed, the server collects the local model parameters from all data centers and aggregates them to update the global model. This process is repeated until the global model converges.

Let P_D and P_U denote observable data center and unseen user distribution. The model mapping function is $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, with f_θ being part of hypothesis collection \mathcal{F} . The goal is to learn a model minimizing empirical error risk $R_D[f_\theta]$ in each data center during training and enhance the model's adaptability to unseen user domain, optimizing the target error risk $R_U[f_\theta]$:

$$R_D[f_\theta] = \min_{f_\theta} E_{(x,y) \sim P_D} [\ell(f_\theta(x), y)], \quad (1)$$

$$R_U[f_\theta] = \min_{f_\theta} E_{(x,y) \sim P_U} [\ell(f_\theta(x), y)]. \quad (2)$$

3.2 Vector Quantization for Learning Codebook

Our motivation is to avoid privacy leaks associated with directly transmitting features. We require a codebook where the vectors do not represent any specific

training sample but can still represent the local data distribution. We are aware of various quantization methods that offer optimizations in terms of storage cost, complexity, and efficiency compared to Vector Quantization (VQ). However, for our method, the performance of these quantization techniques does not need to be excessively prioritized. The success of the VQ-VAE model demonstrates VQ’s capability to store data distribution information. [30] Let’s delve deeper into the specifics of this approach.

Consider a typical image denoted as (x, y) with the style feature, represented as $f_s(x)$, and the content feature, denoted by $f_c(x)$. For ease of reference and to generalize the concept, we use the notation $f(x)$ to represent both these features.

Once the features are extracted, they need to be matched with their nearest counterparts in the embedding space of codebook e . The discrete latent variables, $z(x)$, which essentially are the discrete counterparts of the original features, are computed using a straightforward nearest neighbor lookup method within this embedding space, as demonstrated in Eq. 3:

$$z(x) = e_j, \quad \text{where } j = \operatorname{argmin}_k |f(x) - e_k|_2. \quad (3)$$

The optimization process for this transformation is driven by a well-defined objective function, represented as L_e . It is illustrated as:

$$L_e = |\operatorname{sg}[f(x)] - e|_2^2 + |f(x) - \operatorname{sg}[e]|_2^2. \quad (4)$$

In the above equation, the term $\operatorname{sg}[\cdot]$ denotes the stop-gradient operation. The objective function has been meticulously designed to cater to two primary goals. The first term, by applying the stop-gradient to the features $f(x)$, ensures only the codebook e undergoes updates. This design choice is pivotal for encapsulating and preserving the distribution information within the vectors of the codebook. On the other hand, the second term aims to update the encoder E producing outputs that are as close as possible to the nearest vector in the codebook. This strategy is essential to constrain the range of the discrete embedding space, thereby avoiding any uncontrolled expansion [30].

3.3 Style Codebook Contrastive Learning

To mitigate the adverse effects of distribution information discrepancies among codebooks on classification, we establish the adaptive Style Assembly Layers (SAL) for Contrastive Learning [35], guided by the principles of Adaptive Instance Normalization (AdaIN) [13].

Given an input sequence of length N in a mini-batch, where (x_i, y_i) denotes the input sample ($i \in \{1, 2, \dots, N\}$), the feature encoder E and vector quantization are used to generate discrete style and content features, denoted as $z_s(x_i)$ and $z_c(x_i)$, respectively. Subsequently, the SAL is employed to obtain the assembled feature:

$$S(x_i) = \operatorname{SAL}(z_c(x_i), z_s(x_i)). \quad (5)$$

To enhance the liveness-related information within the codebook, We input $S(x_i)$ into the subsequent classifier for binary classification in supervised training. The model are optimized by the standard cross-entropy loss, denoted as L_{cls} .

To mitigate unnecessary inter-domain distribution discrepancies amongst codebooks, we introduce contrastive learning strategy. For constructing positive sample pairs in the context of contrastive learning, each data center downloads the style codebook of another data center from the server. By applying an identical VQ process to the style feature $f_s(x_i)$, we obtain $z_s^*(x_i)$, which facilitates cross-domain mapping of style features. Thereafter, utilizing the SAL to integrate $z_s^*(x_i)$ with $z_c(x_i)$, we obtain the positive assembled feature sample for contrastive learning:

$$S^*(x_i) = \text{SAL}(z_c(x_i), z_s^*(x_i)). \quad (6)$$

At this point, we have got two assembled features, namely $S(x_i)$ and $S^*(x_i)$. These two features are assembled from the content feature of the same image and the style feature from different domains. Therefore, we aim to reduce the distance between them. After subjecting $S^*(x_i)$ to the predictive Multi-Layer Perceptron (MLP) head, it is termed $P(x_i)$. The divergence between these two features is then quantified using negative cosine similarity:

$$\mathcal{D}(S(x_i), P(x_i)) = -\frac{S(x_i)}{\|S(x_i)\|_2} \cdot \frac{P(x_i)}{\|P(x_i)\|_2}. \quad (7)$$

Considering that $P(x_i)$ is constructed using a style codebook from another data center, making the backpropagation of its gradient for updates becomes inefficient. Therefore, we apply a stop-grad operation [6] on $P(x_i)$, anchoring its position within the feature space. We then minimize the distance between $S(x_i)$ and $P(x_i)$ for our contrastive learning loss:

$$L_{\text{contra}} = \sum_{i=1}^N \mathcal{D}(S(x_i), \text{sg}[(P(x_i))]). \quad (8)$$

It is noteworthy that although our contrastive learning methodology is akin to that employed in SSAN [35], our primary intentions and input features are distinct. SSAN utilizes contrastive learning strategy to underscore style features related to living label, while simultaneously suppressing style traits specific to certain domains. This involves style recombination using different labels, and the choice to either push away or pull closer these styles. In contrast, our method employs style features corresponding to a single sample across different codebooks, with the aim of reducing the distance between similar style codes in the style codebooks of various data centers.

3.4 Shuffle Style Codebook

To effectively utilize the information in the codebook, we perform corresponding operations on the server side based on the unique characteristics of each codebook.

In FAS, content features represent global image attributes and are consistent across domains. In contrast, style features reflect local textures, varying due to factors like lighting and equipment. Thus, we deduce: 1) Content features are uniform across domains. 2) Style features show noticeable differences.

Accordingly, we use the FedAvg method for content codebooks, allowing a shared domain embedding space. As for style codebooks, we concatenate them to form an overarching style codebook $e_{\text{all}} \in R^{MK \times D}$. We then introduce a permutation function SC to shuffle the indices of the vectors in e_{all} :

$$SC : \{1, 2, \dots, MK\} \rightarrow \{i, j, k, \dots, m\}, \quad (9)$$

where $\{i, j, k, \dots, m\}$ is a permutation of $\{1, 2, \dots, MK\}$. Using this permutation function, the shuffled overarching style codebook e'_{all} can be derived as:

$$e'_{\text{all}} = e_{\text{all}}(SC(:, :)). \quad (10)$$

After implementing the shuffled style codebook operation, it acts as a medium to share distribution information across domains, enriching the local data in subsequent training rounds. During the VQ phase, it synthesizes broader representations than those in the local distribution, effectively expanding the training dataset for better domain generalization, similar to data augmentation. Crucially, this method securely distributes information within the codebook discretely, preventing reconstruction of any specific original image, thus maintaining data privacy and meeting high privacy preservation standards.

3.5 Loss Function

Integrating all things mentioned above together, the objective of the proposed FedSC framework is:

$$L_{\text{all}} = L_{\text{cls}} + \lambda_1 \cdot L_{\text{e}} + \lambda_2 \cdot L_{\text{contra}}, \quad (11)$$

where λ_1 and λ_2 are two hyperparameters introduced to balance the proportionality of the different loss functions.

3.6 Theoretic Analysis

In the following, we briefly revisit the theoretical error bound as outlined in [1]. Consider that we have M data centers, each with its respective distribution, represented as P_D^i where $1 \leq i \leq M$. The convex hull, denoted as Λ^D , of the set of these distributions $\{P_D^i\}_{i=1, \dots, M}$ is given by

$$\Lambda_D = \left\{ P : P = \sum_{i=1}^M \pi_i P_D^i, [\pi_1, \dots, \pi_M] \in \Delta_{M-1} \right\}, \quad (12)$$

where Δ^{M-1} refers to the $(M - 1)$ -dimensional simplex which normalizes the weighting coefficients $[\pi_1, \dots, \pi_M]$.

For the unseen user with distribution P_U , we denote P_U^* as the point within the convex hull Λ^D , such that $P_U^* \in \Lambda^D$. This point signifies the smallest distance between P_U and the convex set Λ^D :

$$P_U^* = \sum_{i=1}^M \pi_i^* P_D^i, \tag{13}$$

with

$$(\pi_1^*, \dots, \pi_M^*) = \underset{[\pi_1, \dots, \pi_M] \in \Delta_{M-1}}{\operatorname{argmin}} \quad d_{\mathcal{H}} \left[P_U, \sum_{i=1}^M \pi_i P_D^i \right], \tag{14}$$

where $d_H[P', P'']$ measures the H-divergence between distributions P' and P'' , P_U^* represents the point in Λ^D nearest to P_U . As we lack knowledge about user distribution P_U , we can't directly compute the target error risk $R_U[f_\theta]$. Instead, we use the known source risk $R_D[f_\theta]$ to estimate an upper bound for the target risk.

Theorem 1. (*Upper-bounding the risk on the unseen user distribution [1]*). Given M data centers $\{P_D^i\}_{i=1}^M$ and a user P_U , the solution of (14) is defined as $\pi^* = [\pi_1^*, \dots, \pi_M^*]$. Utilizing the definition of error risk in (2), the target risk $R_U[f_\theta]$, for any $f_\theta \in \mathcal{F}$, is bounded in the context of an unseen user P_U such that $d_H[P_U^*, P_U] = \gamma$. The bound is given by:

$$R_U[f_\theta] \leq \sum_{i=1}^M \pi_i^* R_D^i[f_\theta] + \gamma + \epsilon + \min \{ \mathbf{E}_{P_U^*} [\|f_{D_{\pi^*}} - f_U\|], \mathbf{E}_{P_U} [\|f_U - f_{D_{\pi^*}}\|] \}, \tag{15}$$

where ϵ is the largest pairwise \mathcal{H} -divergence of $\{P_D^i\}_{i=1, \dots, M}$, which can also be regarded as the diameter of the convex hull Λ_D . $f_{D_{\pi^*}} = \sum_{i=1}^{N_D} \pi_i^* f_{D_i}$ is the labeling function of P_U^* .

Equation (15) presents an upper bound consisting of four components. The first component assesses the error within each data center, while the last measures the label distribution difference between data centers and user. A prevalent assumption is that data centers and user share identical conditional distributions [8], rendering the last term in the upper bound redundant.

Given this analysis, our primary objective is to minimize the factors γ and ϵ .

1)The term γ stands for $d_H[P_U^*, P_U]$, indicating the distance between the user's distribution and the convex hull Λ_D . By embedding vector quantization in our method, we substitute the features extracted by the feature extractor with the nearest vector from the codebook. During inference, the unseen user leverages the codebooks from all data centers. This ensures that even if P_U is outside Λ_D , the user's representation distribution P_{Z_U} is encapsulated within

the convex hull of the data centers’ representation distributions $\{P_{Z_D}^i\}_{i=1, \dots, M}$. In this scenario, we achieve the optimal situation where γ vanishes.

2) To reach a lower ϵ , which signifies the diameter of the data center distribution convex hull A_D , an effective strategy is to condense the distance among differing data center distributions. Building upon this concept, our method integrates a strategy of contrastive learning. This innovative approach is employed to systematically discern and minimize unnecessary variations in the codebooks across different data centers. It effectively amplifies the similarities between representations, fostering closer alignment and cohesion between data center distributions.

4 Experiments

4.1 Implementation Details

Data Preparation. Four datasets were utilized for evaluating our approach: OULU-NPU (O) [4], CASIA-MFSD (C) [42], Replay-Attack (I) [7], and MSU-MFSD (M) [36]. These datasets consist of both image and video data. All available images were utilized for the image data, while frames were extracted at specified intervals for the video data. After transforming the data into an image format, the MTCNN algorithm [40] was used for face detection. The detected faces were then cropped and resized to an input size of $256 \times 256 \times 6$, with both the RGB and HSV channels extracted from each input image.

Experimental Setup. In our study within the Federated Learning framework, we adopted a distinct testing protocol to gauge model generalization [27]. Models were trained on multiple datasets, barring one. This excluded dataset, emulating user behavior, was then used for testing. By training on both real and spoof images from the data centers, we assessed the model’s effectiveness in distinguishing living from spoofing images in the user-emulated dataset.

Training Setting. To facilitate fair comparisons, we employed the ResNet-18 [12] architecture as the shallow feature extraction network. The network was implemented using the PyTorch framework, and the training process utilized a 1080Ti GPU. The Adam optimizer with a weight decay of $1e-2$ was used for optimization. The initial learning rate was set to $3e-4$, and we adopted the CosineAnnealingLR strategy for learning rate decay, with its period set to half of the total epochs. Constraints due to the GPU memory size necessitated a training batch size of 6.

Testing Setting. In our experiments, we employed widely-accepted metrics, such as HTER [3], EER, and AUC for cross-domain evaluations. Notably, during server-side user performance evaluations on the test set, the style codebook used is an amalgamation of those from all data centers, instead of originating from just one or by averaging multiple centers’ codebooks.

4.2 Experimental Results

In this section, we compare our model with the baseline from [27], emphasizing our method’s generalization. We used three datasets from O, C, I, and M to represent

Table 1. The results of testing on OULU-NPU, CASIA-MFSD, Replay-Attack, and MSU-MFSD.

Methods	Data Centers	User	HTER (%)	EER (%)	AUC (%)	Avg. HTER	Avg. EER	Avg. AUC
Fused	O&C&I	M	34.42	23.26	81.67	35.75	31.29	73.89
	O&M&I	C	38.32	38.31	67.93			
	O&C&M	I	42.21	41.36	59.72			
	I&C&M	O	28.04	22.24	86.24			
All	O&C&I	M	21.80	17.18	90.96	27.26	25.09	80.42
	O&M&I	C	29.46	31.54	76.29			
	O&C&M	I	30.57	25.71	72.21			
	I&C&M	O	27.22	25.91	82.21			
FedPAD	O&C&I	M	19.45	17.43	90.24	32.17	28.84	76.51
	O&M&I	C	42.27	36.95	70.49			
	O&C&M	I	32.53	26.54	73.58			
	I&C&M	O	34.44	34.45	71.74			
FedGPAD	O&C&I	M	12.73	13.36	91.25	18.59	17.48	89.25
	O&M&I	C	28.69	27.55	80.58			
	O&C&M	I	10.97	11.11	95.34			
	I&C&M	O	21.95	17.91	89.85			
Ours	O&C&I	M	16.88	17.00	91.47	17.01	17.06	90.30
	O&M&I	C	19.89	20.00	87.90			
	O&C&M	I	9.86	9.85	95.70			
	I&C&M	O	21.39	21.37	86.13			

Table 2. Comparison (%) of FAS methods for domain generalization

Method	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O		Avg.	
	HTER (%)	AUC (%)	HTER (%)	AUC (%)	HTER (%)	AUC (%)	HTER (%)	AUC (%)	HTER (%)	AUC (%)
	Without Considering Privacy									
MMD-AAE	27.08	83.19	44.59	58.29	31.58	75.18	40.98	63.08	36.05	69.93
MADDG	17.69	88.06	24.50	84.51	22.19	84.99	27.98	80.02	23.09	84.39
SSDG-M	16.67	90.47	23.11	85.45	18.21	94.61	25.17	81.83	20.79	88.09
DR-MD-Net	17.02	90.10	19.68	87.43	20.87	86.72	25.02	81.47	20.64	86.43
RFMeta	13.89	93.98	20.27	88.16	17.30	90.48	16.45	91.16	16.98	90.94
NAS-FAS	19.53	88.63	16.54	90.18	14.51	93.84	13.80	93.43	16.09	91.52
SDA	15.40	91.80	24.50	84.40	15.60	90.10	23.10	84.30	19.65	87.65
ANRL	10.83	96.75	17.83	89.26	16.03	91.04	15.67	91.90	15.09	92.24
SSAN-M	10.42	94.76	16.47	90.81	14.00	94.58	19.51	88.17	15.1	92.08
SSAN-R	6.67	98.75	10.00	96.67	8.88	96.79	13.72	93.63	9.82	96.46
SA-FAS	5.95	96.55	8.78	95.37	6.58	97.54	10.00	96.23	7.83	96.42
	Considering Privacy									
FedGPAD	12.73	91.25	28.69	80.58	10.97	95.34	21.95	89.85	18.59	89.25
FedSC (Ours)	16.88	91.47	19.89	87.90	9.86	95.70	21.39	86.13	17.01	90.30

different data centers, with one reserved for user simulation, detailed in Table 1. 'Fused' indicates aggregated predictions from each center, while 'All' integrates all data for training, challenging federated learning's privacy principles.

Table 3. Evaluations of different components of the proposed method

VQ	shuffle codebook	L_{contra}	O&C&I to M		O&M&I to C		O&C&M to I		I&C&M to O	
			HTER (%)	AUC (%)	HTER (%)	AUC (%)	HTER (%)	AUC (%)	HTER (%)	AUC (%)
			22.50	81.60	21.11	87.21	26.50	77.80	26.53	80.68
✓			19.58	88.10	23.33	84.01	28.90	74.80	27.20	79.90
✓	✓		18.96	88.69	19.91	86.67	23.04	79.00	24.02	83.84
✓		✓	15.58	92.60	25.59	85.56	19.69	80.29	25.05	84.03
✓	✓	✓	16.88	91.47	19.89	87.90	9.86	95.70	21.39	86.13

Empirically, our model excels in ‘O&M&I to C’ and ‘O&C&M to I’, showing competitive performance elsewhere. Averaged across metrics, our model demonstrates strong generalization without relying on additional information, unlike FedGPAD.

Comparison With State-of-the-Art FAS Methods. To further delineate the generalization prowess of our proposed method, we have benchmarked it against a spectrum of avant-garde FAS domain generalization techniques. It is pertinent to mention that these methodologies have not specifically catered to privacy-related concerns. The array of techniques compared includes MMD-AA [16], MADDG [25], SSDG-M [14], DR-MD-Net [31], RFMeta [26], SDA [32], ANRL [20], SSAN [35], and SA-FAS [28].

As elucidated in Table 2, it is evident that our method, notwithstanding its lack of access to comprehensive source domain data from the data centers, attains an average performance on par with many established FAS domain generalization techniques. A distinctive feature of our approach is its operational simplicity, devoid of reliance on complex technologies, elaborate training methods, or auxiliary data. This inherent simplicity in concept and ease of training further corroborate the effectiveness of our method in the domain of face anti-spoofing.

Ablation Study. To validate the contributions of various components in our proposed model, we conducted experiments under identical conditions using several incomplete models. We conducted ablation studies on several components of FedSC: vector quantization, shuffle codebook, and contrastive learning. Since VQ is the foundation for the other components, we removed all components, leaving only the core model structure used for classification as baseline. The results are presented in Table 3, which demonstrate that each component of our model contributes to its performance enhancement.

4.3 Visualization and Analysis

Features Visualization. To delve into the feature space constructed by our FedSC method, we employed the t-distributed stochastic neighbor embedding (t-SNE) [21] technique for visualization. As discernible from Fig. 2, both style and

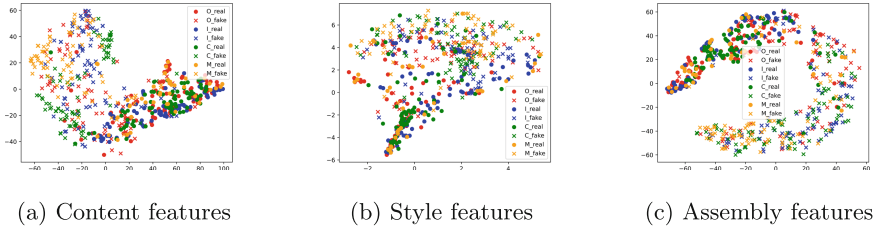


Fig. 2. The visualization of different features under protocol I&C&M to O. The graphs of (a), (b), and (c) describe the feature distribution of content features, style features, and assembly features, respectively. Different colors indicate features from different datasets. Different shapes represent different liveness label: point=living, cross=spoofing.

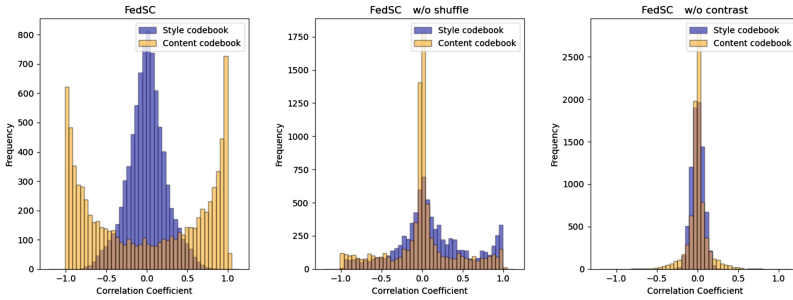


Fig. 3. The Correlation Analysis of Codebook

content features exhibit a discernible distinction between living and spoofing samples. Notably, the features derived from the style assembly layer exhibit superior cohesion within the live samples and a heightened inter-class separation. These findings further underscore the efficacy of our proposed methodology.

Correlation Analysis. To elucidate the impact of our method’s components on the codebook, we analyze their correlations in an ablation study. As depicted in Fig. 3, the full method shows the content codebook with pronounced positive and negative correlations, while the style codebook leans towards orthogonality, aligning with our style-content hypothesis. Without the shuffle codebook operation, the style codebook’s orthogonality drops, hinting at redundancy. Removing the contrastive loss function L_{contra} increases orthogonality in both codebooks, suggesting L_{contra} strengthens the correlation for identical label vectors in the codebook.

Scalability and Computational Overhead Analysis. We have listed some experimental data in Table 4. As shown, the parameters introduced by the codebook in FedSC account for only about 0.2% of the total model parameters, ensuring scalability with respect to the training data volume. Another column in Table 4 presents the time consumption for simulating the shuffle codebook operation on the server side, showing an increase of only about 3% compared to

Table 4. Parameter quantity and Time consumption

Method	Parameter quantity	Time consumption
FedAvg	5.8 million	1.13 s
FedSC	5.8 + 0.032 million	1.13 + 0.04 s

FedAvg. This increase is within an acceptable range and does not significantly burden computational overhead and latency.

5 Conclusion and Discussion

In this study, we address the domain generalization problem in FAS within the federated learning framework by introducing our FedSC framework. Aimed at creating a versatile FAS model while ensuring data privacy, we integrated style and content codebooks to store local texture and global image information from each data center. Our shuffle codebook method effectively transmits style information between data centers and employs contrastive learning to reduce unnecessary distribution information differences in the codebooks. Our experimental results demonstrate the effectiveness of our method, suggesting its potential beyond FAS, in the broader FedDG challenge, merits further investigation. In future research, we hope to enhance the model’s ability to cope with new and evolving spoofing attacks by optimizing the use of distribution information in codebooks and refining contrastive learning strategies.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant No. 62276030 and No. 62306043.

References

1. Albuquerque, I., Monteiro, J., Darvishi, M., Falk, T.H., Mitliagkas, I.: Generalizing to unseen domains via distribution matching. arXiv preprint [arXiv:1911.00804](https://arxiv.org/abs/1911.00804) (2019)
2. Atoum, Y., Liu, Y., Jourabloo, A., Liu, X.: Face anti-spoofing using patch and depth-based CNNs. In: 2017 IEEE International Joint Conference on Biometrics (IJCB), pp. 319–328. IEEE (2017)
3. Bengio, S., Mariéthoz, J.: A statistical significance test for person authentication. In: Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop, No. CONF (2004)
4. Boulkenafet, Z., Komulainen, J., Li, L., Feng, X., Hadid, A.: Oulu-npu: a mobile face presentation attack database with real-world variations. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), pp. 612–618. IEEE (2017)
5. Chen, J., Jiang, M., Dou, Q., Chen, Q.: Federated domain generalization for image recognition via cross-client style transfer. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 361–370 (2023)

6. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15750–15758 (2021)
7. Chingovska, I., Anjos, A., Marcel, S.: On the effectiveness of local binary patterns in face anti-spoofing. In: 2012 BIOSIG-Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG), pp. 1–7. IEEE (2012)
8. David, S.B., Lu, T., Luu, T., Pál, D.: Impossibility theorems for domain adaptation. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 129–136. JMLR Workshop and Conference Proceedings (2010)
9. Deb, D., Jain, A.K.: Look locally infer globally: a generalizable face anti-spoofing approach. *IEEE Trans. Inf. Forens. Secur.* **16**, 1143–1157 (2020)
10. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: LBP- top based countermeasure against face spoofing attacks. In: Computer Vision-ACCV 2012 Workshops: ACCV 2012 International Workshops, Daejeon, 5–6 November 2012, Revised Selected Papers, Part I, vol. 11, pp. 121–132. Springer (2013)
11. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180–1189. PMLR (2015)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
13. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1501–1510 (2017)
14. Jia, Y., Zhang, J., Shan, S., Chen, X.: Single-side domain generalization for face anti-spoofing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8484–8493 (2020)
15. Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: strategies for improving communication efficiency. *arXiv preprint [arXiv:1610.05492](https://arxiv.org/abs/1610.05492)* (2016)
16. Li, H., Pan, S.J., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5400–5409 (2018)
17. Li, H., Wang, Y., Wan, R., Wang, S., Li, T.Q., Kot, A.: Domain generalization for medical imaging classification with linear-dependency regularization. *Adv. Neural. Inf. Process. Syst.* **33**, 3118–3129 (2020)
18. Lin, B., Li, X., Yu, Z., Zhao, G.: Face liveness detection by RPPG features and contextual patch-based CNN. In: Proceedings of the 2019 3rd International Conference on Biometric Engineering and Applications, pp. 61–68 (2019)
19. Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.A.: Feddgc: federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1013–1023 (2021)
20. Liu, S., et al.: Adaptive normalized representation learning for generalizable face anti-spoofing. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 1469–1477 (2021)
21. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *J. Mach. Learn. Res.* **9**(11) (2008)
22. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)

23. Patel, K., Han, H., Jain, A.K.: Secure face unlock: spoof detection on smartphones. *IEEE Trans. Inf. Forens. Secur.* **11**(10), 2268–2283 (2016)
24. Qin, Y., et al.: Learning meta model for zero-and few-shot face anti-spoofing. *Proc. AAAI Conf. Artif. Intell.* **34**, 11916–11923 (2020)
25. Shao, R., Lan, X., Li, J., Yuen, P.C.: Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10023–10031 (2019)
26. Shao, R., Lan, X., Yuen, P.C.: Regularized fine-grained meta face anti-spoofing. *Proc. AAAI Conf. Artif. Intell.* **34**, 11974–11981 (2020)
27. Shao, R., Perera, P., Yuen, P.C., Patel, V.M.: Federated generalized face presentation attack detection. *IEEE Trans. Neural Netw. Learn. Syst.* (2022)
28. Sun, Y., Liu, Y., Liu, X., Li, Y., Chu, W.S.: Rethinking domain generalization for face anti-spoofing: Separability and alignment. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24563–24574 (2023)
29. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: *CVPR 2011*, pp. 1521–1528. *IEEE* (2011)
30. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. *Adv. Neural Inf. Process. Syst.* **30** (2017)
31. Wang, G., Han, H., Shan, S., Chen, X.: Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6678–6687 (2020)
32. Wang, J., Zhang, J., Bian, Y., Cai, Y., Wang, C., Pu, S.: Self-domain adaptation for face anti-spoofing. *Proc. AAAI Conf. Artif. Intell.* **35**, 2746–2754 (2021)
33. Wang, W., Liu, P., Zheng, H., Ying, R., Wen, F.: Domain generalization for face anti-spoofing via negative data augmentation. *IEEE Trans. Inf. Forens. Secur.* (2023)
34. Wang, Z., Wang, Q., Deng, W., Guo, G.: Learning multi-granularity temporal characteristics for face anti-spoofing. *IEEE Trans. Inf. Forens. Secur.* **17**, 1254–1269 (2022)
35. Wang, Z., et al.: Domain generalization via shuffled style assembly for face anti-spoofing. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4123–4133 (2022)
36. Wen, D., Han, H., Jain, A.K.: Face spoof detection with image distortion analysis. *IEEE Trans. Inf. Forens. Secur.* **10**(4), 746–761 (2015)
37. Yu, Z., Li, X., Niu, X., Shi, J., Zhao, G.: Face anti-spoofing with human material perception. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, 23–28 August 2020, Proceedings, Part VII*, 16, pp. 557–575. Springer (2020)
38. Yu, Z., Qin, Y., Li, X., Zhao, C., Lei, Z., Zhao, G.: Deep learning for face anti-spoofing: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(5), 5609–5631 (2022)
39. Yu, Z., et al.: Searching central difference convolutional networks for face anti-spoofing. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5295–5305 (2020)
40. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **23**(10), 1499–1503 (2016)
41. Zhang, R., Xu, Q., Yao, J., Zhang, Y., Tian, Q., Wang, Y.: Federated domain generalization with generalization adjustment. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3954–3963 (2023)

42. Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D., Li, S.Z.: A face antispoofing database with diverse attacks. In: 2012 5th IAPR International Conference on Biometrics (ICB), pp. 26–31. IEEE (2012)
43. Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C.: Domain generalization: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022)
44. Zhou, L., Luo, J., Gao, X., Li, W., Lei, B., Leng, J.: Selective domain-invariant feature alignment network for face anti-spoofing. *IEEE Trans. Inf. Forens. Secur.* **16**, 5352–5365 (2021)



LoHoSC: Low Order High Order Style Consistency for Syn-to-Real Domain Generalized Semantic Segmentation

Sudhakar Kumawat¹(✉)  and Hajime Nagahara² 

¹ IIT (ISM) Dhanbad, Dhanbad, India
sudhakar@ids.osaka-u.ac.jp

² Osaka University, Osaka, Japan

Abstract. In recent years, large-scale synthetic image datasets have proven to be a boon for training deep semantic segmentation models due to their easy scalability and cost-effective annotation processes. However, models trained on synthetic images often fail to generalize well when deployed in the real world. To solve this problem, various domain randomization (DR) techniques have been introduced to help generalize the models in real settings. One common aspect of such DR techniques is their usage of low-order statistics, particularly the mean and standard deviation for generating new styles during training. However, real images have more complex distributions than Gaussian and thus, high-order statistics also need to be considered for generating new styles. Towards this goal, this paper proposes Low order High order Style Consistency (LoHoSc), a new Domain Randomization framework consisting of two modules, LoSC and HoSC. During training, LoSC and HoSC generate random styles using low-order statistics (e.g., mean and standard deviation) and high-order statistics (e.g., empirical Cumulative Distribution Functions), respectively, in the feature space. The predictions corresponding to the two styles are then constrained in the loss space to learn content-relevant information while discarding any style variant information. Evaluation of LoHoSC on various benchmark datasets shows that it achieves state-of-the-art Domain Generalization capabilities, both quantitatively and qualitatively.

Keywords: Semantic Segmentation · Domain Generalization · Domain Randomization

This work was supported by JSPS KAKENHI Grant Number 22K17976. Work done when the first author was a postdoctoral researcher at Osaka University.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15303, pp. 243–258, 2025.
https://doi.org/10.1007/978-3-031-78122-3_16

1 Introduction

Semantic segmentation is one of the core computer vision tasks where the goal is to associate each pixel in the image with its object class. In the real world, it has many applications ranging from autonomous driving [1, 35], augmented reality [10, 26], robotics [17, 25] to medical imaging [16, 39]. In the last decade, with the help of good computing power and advancements in deep learning architectures, many promising results have been achieved in semantic segmentation. However, deep learning relies heavily on the availability of large-scale annotated images from the real world for their training, and developing such annotations is expensive and time-consuming since class labels must be generated at the pixel level for each of the real images. Thus, to mitigate this issue, various synthetic image datasets are introduced, such as GTAV [22] and SYNTHIA [24], which are inexpensive and fast to construct on a large scale [22].

An important challenge in using synthetic image datasets for training deep learning architectures is their tendency to overfit the synthetic domains. This leads to a huge performance drop when such models are tested on images from real domains. This phenomenon can be attributed to two factors. First, the synthetic domain may contain limited environmental variations from the real domains. Second, synthetic and real-world images have distinct style and texture variations in objects. Note that, from here onwards, we will refer to the synthetic and real as source and target domains, respectively. One of the solutions to reduce this domain gap is Domain Adaptation, which works by using unannotated images from the target domains as a reference during the supervised training of the segmentation model using images from the source domains. However, domain adaptive methods perform well only on target domains seen during the training, limiting their applicability in the real world as the model may face many unseen domains.

Compared to Domain Adaptation, Domain Generalization proves to be a more realistic and universal solution for reducing the domain gap between the source and target domains since its goal is to generalize to any arbitrary target domain that it may not have seen during training. Existing domain generalized semantic segmentation approaches can broadly be categorized into two types depending on the technique used: Domain Normalization (DN) [4] and Domain Randomization (DR) [29]. Among these two, the methods based on DR have proven to be more effective in reducing the domain gap between arbitrary domains. Technically, DR works by applying random styles to the source domain images in the image or feature space while aligning their content information in the loss space. Various recent state-of-the-art methods using DR for achieving Domain Generalized Semantic Segmentation include WEDGE [14], AdvStyle [37], SiamDoGe [29], SHADE [36], and TLDR [15]. Please refer to Sect. 2 for more details on these methods. One important feature of these DR-based methods for Domain Generalized Semantic Segmentation (DGSS) is their focus on using only mean and standard deviation to represent the style information in images. However, in the real world, this is valid only if we assume that the feature distribution in real data is strictly Gaussian. However, images from

the real-world may have more complicated feature distributions. Thus, high-order statistics other than mean and standard deviation must be considered to represent style information.

Considering these facts, we propose a new Domain Randomization method called LoHoSC (Low-order and High-order Style Consistency). Our Domain Randomization method differs from previous works in not only using mean and standard deviation for representing style information but also high-order statistics such as empirical Cumulative Distribution Functions (eCDFs) of image features. More precisely, as shown in Fig. 1, we introduce two consistency constraints, Low order Style Consistency (LoSC) and High order Style Consistency (HoSC), to encourage the segmentation model to learn style invariant information while discarding any style-related information originating from Gaussian as well as other complex distributions. For LoSC, we use MixStyle [38], which uses channel-wise mean and standard deviation for generating new styles. For HoSC, we use EFD-Mix [34], which uses high-order feature statistics for generating samples with diverse styles. We will discuss LoSC and HoSC in more detail in Sect. 3. Once the stylized samples are generated, we enforce pixel-level consistency between them in the loss space. Finally, our evaluation of LoHoSC in various settings, such as single-source and multi-source settings, shows that it significantly improves the generalization capabilities of the segmentation model. For example, in the case of the single-source setting, when trained on the GTAV dataset, it outperforms TLDR [15], the current state-of-the-art method, by a margin of 0.22% on CityScapes, 1.1% on BDD100K, and 2.25% on Mapillary datasets on the mIOU metric.

In summary, the main contributions of this work are as follows:

- We identify an important issue in existing Domain Randomization techniques used in state-of-the-art Domain Generalized Semantic Segmentation frameworks: their pure reliance on mean and standard deviation for generating new styles.
- We advocate for using high-order statistics also to generate diverse styles and propose a new Domain Randomization method called LoHoSC, which uses low-order as well as high-order statistics to generate new styles.
- We extensively evaluate LoHoSC and show that using high-order and low-order statistics for style generation improves the generalization capabilities of segmentation models quantitatively and qualitatively.

2 Related Works

Domain Generalization. As discussed in Sect. 1, the goal of Domain Generalization (DG) frameworks is to train Deep Neural Networks (DNNs) on known source domains such that they generalize well on arbitrary target domains. DG has primarily been studied for image classification, leading to the development of many state-of-the-art DG frameworks that can produce robust classification models. Over the last few years, many state-of-the-art Domain Generalized Image Classification (DGIC) frameworks have been proposed such

as MixStyle [38], EFDMix [34], RandConv [31], Progressive RandConv [3], FACT [30], and L2D [28].

Domain Generalized Semantic Segmentation. In comparison to DGIC, Domain Generalized Semantic Segmentation (DGSS) is in a nascent stage and a growing area of interest. Early works attempted to solve this problem by using techniques such as normalization and whitening, which worked by normalizing the mean (and standard deviation) and whitening the covariance of features in the source domain to remove any domain-specific features. For example, Pan *et al.* [19] proposed IBN-Net that carefully integrates Instance Normalization and Batch Normalization as building blocks in deep CNNs to eliminate domain-features such as color and style while preserving content-related features. Pan *et al.* [20] proposed Switchable Whitening, which adaptively selects appropriate whitening or standardization to decorrelate features. Choi *et al.* [4] proposed Instance Selective Whitening (ISW) to disentangle domain-specific feature and domain invariant content encoded in higher-order statistics and selectively removes any style information causing domain shifts. Peng *et al.* [21] proposed two modules, namely Semantic-Aware Normalization (SAN) and Semantic-Aware Whitening (SAW), for enforcing both intra-category compactness and inter-category separability.

Recent works in this area use Domain Randomization (DR) which works by applying random styles to the source domain images in the image or feature space while aligning their content information using consistency loss. Yue *et al.* [33] first explored DR by stylizing source domain images using ImageNet [7] images during training and enforcing content consistency among these stylized images. Huang *et al.* [11] applied DR in the frequency domain by randomizing the domain-variant frequency components while keeping the domain-invariant frequency components intact. Kim *et al.* [14] proposed WEB-image assisted Domain GEneralization (WEDGE) scheme, which exploits the diversity (styles) of web-crawled images for generalizable semantic segmentation. Very recent methods using DR include AdvStyle [37], SiamDoGe [29], SHADE [36], and TLDR [15]. In AdvStyle, Zhong *et al.* [37] proposed an adversarial style augmentation (AdvStyle) approach, which can dynamically generate stylized images during training which can effectively prevent overfitting on the source domain. In SiamDoGe, Wu *et al.* [29] used color jittering to generate a pair of random stylized images of a source domain sample and employed a Siamese architecture to learn domain-agnostic features during training. In SHADE, Zhao *et al.* [36] proposed a Style Hallucination Module (SHM) to generate style-diversified samples, which is followed by Style Consistency (SC) and Retrospection Consistency (RC) modules to align content information during training. In TLDR, Kim *et al.* [15] proposed a texture regularization loss to prevent overfitting to source domain textures by using texture features from an ImageNet pre-trained model and a texture generalization loss that utilizes random style images to learn diverse texture representations in a self-supervised manner.

3 Methodology

As illustrated in Fig. 1, the proposed LoHoSC module is composed of two sub-modules- LoSC and HoSC. In the next few sections, we discuss these modules in detail.

3.1 Low Order Style Consistency

As discussed in Sect. 1, we use mean and standard deviation to generate new styles in Low Order Style Consistency (LoSC). In this paper, we rely on the MixStyle [38] module for this purpose and use it in a plug-and-play manner. Below we discuss the background and mathematical formulation of MixStyle module in detail.

Background. Assuming that real images follow Gaussian distribution, Huang *et al.* [13] showed that the channel-wise mean and standard deviation of feature maps can represent the style information in images and proposed Adaptive Instance Normalization (AdaIN), which can be used to generate new images by integrating style and content from different images as shown in Eq. 1.

$$\text{AdaIN}(x, y) = \sigma(y) \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y) \quad (1)$$

Here, $x \in \mathbb{R}^{C \times H \times W}$ and $y \in \mathbb{R}^{C \times H \times W}$ represent the feature maps providing the content and style information, respectively. $\mu(*) \in \mathbb{R}^C$ and $\sigma(*) \in \mathbb{R}^C$ represent the channel-wise mean and standard deviation, respectively, as defined in Eqs. 2 and 3.

$$\mu(x)_{(c)} = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W x_{(c,h,w)} \quad (2)$$

$$\sigma(x)_{(c)} = \sqrt{\frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W \left(x_{(c,h,w)} - \mu(x)_{(c)} \right)^2 + \epsilon}, \quad \epsilon = 10^{-10}. \quad (3)$$

Here, the newly generated feature map borrows style information from y and content information from x . Following this, various Domain Generalization methods [38] showed that the image’s style information causes severe domain shifts, which led to the use of AdaIN as an integral part of various Domain Randomization methods in various Domain Generalization frameworks [29, 36, 38].

MixStyle Module. MixStyle [38] follows ideas from AdaIN and can be easily implemented into mini-batch training. It exploits the fact that many sub-domains exist within a domain, meaning a mini-batch may contain images from multiple sub-domains. Furthermore, since each sub-domain has distinct style

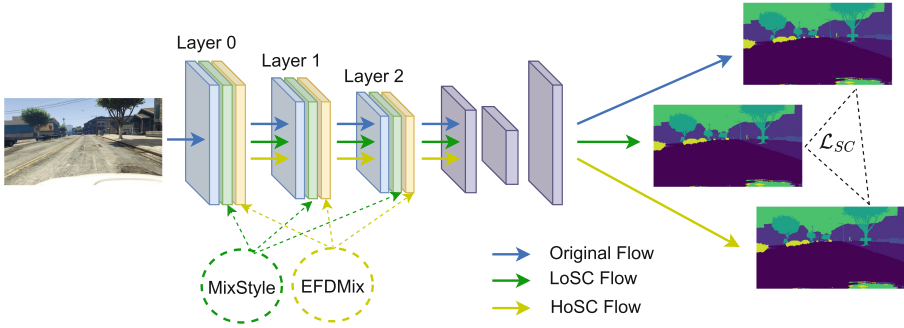


Fig. 1. An illustration of our LoHoSC-based Domain Generalized Semantic Segmentation framework. Note that the MixStyle and EFDMix modules are applied in Layer 0, 1, and 2 of the ResNet backbone.

features, MixStyle uses this fact to intermix style information between various sub-domains. Let x be a mini-batch of images. MixStyle first constructs a shuffled (random) version of x denoted by \tilde{x} . After that, it computes the mixed feature statistics using the following equations.

$$\gamma_{mix} = \lambda\sigma(x) + (1 - \lambda)\sigma(\tilde{x}) \tag{4}$$

$$\beta_{mix} = \lambda\mu(x) + (1 - \lambda)\mu(\tilde{x}) \tag{5}$$

where $\lambda \in R^B$ are instance-wise weights sampled from the Beta distribution, $\lambda \sim Beta(\alpha, \alpha)$ with $\alpha \in (0, \infty)$ being a hyper-parameter. Following the original paper, we use $\alpha = 0.1$ in this work. Finally, the mixed feature statistics are used to generate a newly stylized version of x using Eq. 6.

$$MixStyle(x) = \gamma_{mix} \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \beta_{mix} \tag{6}$$

3.2 High Order Style Consistency

As discussed in Sect. 1, we use high-order statistics, in particular, eCDFs to generate new styles in High Order Style Consistency (HoSC). In this paper, we rely on the EFDMix [34] module for this purpose and use it in a plug-and-play manner. Below, we discuss the background and mathematical formulation of the EFDMix module.

Background. As discussed earlier, the feature distributions in the real images are too complex to be modeled by Gaussian, which makes the process of feature distribution matching during style transfer less accurate. Thus, matching of high-order statistics such as empirical Cumulative Distribution Functions (eCDFs)

of image features is desired for accurate style transfer. Traditionally, such an Exact Feature Distribution Matching (EFDM) is computationally expensive; however, it is feasible with the Exact Histogram Matching (EHM) algorithm [5, 9] applied in the feature space. Furthermore, the Sort Matching [23] can improve its computational complexity. Mathematically, the goal of the EHM algorithm (Eq. 7) is to transform an image x into an output vector o whose eCDF matches the target eCDF of a target image y .

$$\text{EFDM}(x, y) : o_{\tau_i} = x_{\tau_i} + y_{\kappa_i} - \langle x_{\tau_i} \rangle \quad (7)$$

where $\{x_{\tau_i}\}_{i=1}^n$ and $\{y_{\kappa_i}\}_{i=1}^n$ are sorted values of x and y in ascending order with indexes τ and κ . $\langle * \rangle$ denotes the stop gradient operation. Kindly note that a detailed discussion of the EHM and the Sort Matching algorithm is beyond the scope of this paper, and readers may refer to the cited sources for more details.

EFDMix Module. For style augmentation, the EFDM expression can be used instead of the AdaIN, just like MixStyle as shown in Eq. 8. Note that EFDMix borrows ideas and assumptions from MixStyle regarding its implementation in a mini-batch. Furthermore, the hyperparameter λ is also the same as MixStyle.

$$\text{EFDMix}(x, y) : o_{\tau_i} = x_{\tau_i} + (1 - \lambda)y_{\kappa_i} - (1 - \lambda)\langle x_{\tau_i} \rangle \quad (8)$$

3.3 Loss Function

Let P_o denote the posterior probability of the predicted segmentation maps produced by the segmentation model corresponding to the source domain image. Similarly, let P_m , and P_e denote the posterior probabilities corresponding to the LoSC and HoSC modules, respectively. Furthermore, let P_a be the mixture of the above three probabilities defined as in Eq. 9.

$$P_a = \frac{P_o + P_m + P_e}{3} \quad (9)$$

Next, we define our style consistency loss (\mathcal{L}_{SC}) as given in Eq. 10. The goal of \mathcal{L}_{SC} is to constrain the low-order and high-order style variations to enable the model to focus on learning style-invariant content information.

$$\mathcal{L}_{SC} = \frac{\mathcal{KL}(P_a, P_o) + \mathcal{KL}(P_a, P_m) + \mathcal{KL}(P_a, P_e)}{3} \quad (10)$$

In addition to the constraining different styles, we constrain the predicted segmentation maps to the ground-truth labels (Gt) using cross-entropy loss as defined in Eq. 11.

$$\mathcal{L}_{CE} = \mathcal{L}_{CE}(P_o, Gt) + \mathcal{L}_{CE}(P_m, Gt) + \mathcal{L}_{CE}(P_e, Gt) \quad (11)$$

Our Final loss function for training is a combination of \mathcal{L}_{SC} and \mathcal{L}_{CE} as defined in Eq. 12. In this paper, we set $\alpha = 10$ in our experiments.

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \alpha \mathcal{L}_{SC} \quad (12)$$

4 Experiments

4.1 Experimental Setup

Datasets. In Table 1, we summarize the datasets used in this work. We use two synthetic datasets as source domains - GTAV [22] and SYNTHIA [24]. GTAV consists of 24,966 images split into train, validation, and test sets, where each image is of size 1914×1052 . SYNTHIA consists of 9,400 images split into train and validation sets (no test set) where each image is of size 960×720 . We use three real-world datasets as target domains - CityScapes [6], BDD-100K [32], and Mapillary [18]. CityScapes consists of 5,000 images split into train, validation, and test sets where each image is of size 2048×1024 . BDD-100K consists of 10,000 images split into train and validation sets (no test set) where each image is of size 1280×720 . Finally, Mapillary consists of 25,000 images split into train and validation sets (no test set) where each image is of size 1920×1080 .

Table 1. Details about the real and synthetic datasets used in our work.

	Synthetic		Real		
	GTAV	SYNTHIA	CityScapes	BDD100K	Mapillary
Train	12,403	6,580	2,975	7,000	18,000
Val	6,382	2,820	500	1,000	2,000
Test	6,181	–	1,525	2,000	5,000

Implementation Details. Technically, our goal in this paper is to use our proposed approach from Sect. 3 to train a semantic segmentation model on images from the synthetic source domain datasets and show improvement in its generalization capabilities on images from various real-world target domain datasets.

To show this, we conduct experiments under two different settings: single-source domain setting and multi-source domain setting. Under the first setting, the segmentation model is trained on the images from a single synthetic image dataset, which is GTAV, in our case. In the second setting, the model is trained on images from multiple synthetic image datasets, GTAV and SYNTHIA, in our case. Note that we follow these settings from previous works, such as [4], which allow us to compare these works fairly.

Following previous works [4, 15, 29, 36, 37], we use the DeepLabV3+ [2] architecture for segmentation which can be equipped with various backbones such as ResNet, MobileNet, ShuffleNet, and ResNext [4]. In this paper, we use the ResNet-50 and ResNet-101 networks as backbones that are pre-trained on the ImageNet [7] dataset. The LoHoSC module is inserted after Layer 0, Layer 1, and Layer 2 of the backbone. The hyper-parameter values for the LoSC and HoSC modules and the \mathcal{L}_{SC} loss are discussed in Sect. 3.

Table 2. Comparison with state-of-the-art methods in the single-source DG setting. All models use the ResNet-50 backbone and are trained with the GTAV train set. “Extra Data” denotes using extra real-world data such as images from ImageNet as auxiliary domains during training. (*) denotes the performance of supervised learning when the model is trained and validated on the same dataset.

Methods (GTAV)	Extra Data	mIoU (%)			
		CityScapes	BDD100K	Mapillary	Mean
Baseline	✗	28.95 (77.5)	25.14 (63.6)	28.18 (54.1)	27.42
SW [20]	✗	29.91	27.48	29.71	29.03
IterNorm [12]	✗	31.81	32.70	33.88	32.79
RandConv [31]	✗	35.38	30.92	32.43	32.91
DRPC [33]	✓	37.42	32.14	34.12	34.56
IBN-Net [19]	✗	33.85	32.30	37.75	34.63
ISW [4]	✗	36.58	35.20	40.33	37.37
Baseline + AdvStyle [37]	✗	39.62	35.54	37.00	37.39
IBN-Net + AdvStyle [37]	✗	39.32	36.42	40.82	38.85
ISW + AdvStyle [37]	✗	39.60	38.59	41.89	40.03
Pro-RandConv [3]	✗	42.36	37.03	41.63	40.34
SiamDoGe [29]	✗	42.96	37.54	40.64	40.38
SHADE [36]	✗	44.65	39.28	43.34	42.42
TLDR [15]	✗	46.51	42.58	46.18	45.09
LoSC	✗	43.89	40.55	47.36	43.93
HoSC	✗	44.67	40.96	47.41	44.35
LoHoSC	✗	46.73	43.68	48.43	46.28

We train DeepLabV3+ using the SGD optimizer with momentum 0.01, weight decay 0.00005, and the initial learning rate set to 0.005 for the backbone and 0.01 for the remaining model. Note that initializing the backbone with a lower learning rate makes sense due to the presence of pre-trained weights. We use polynomial decay with power 0.9 as a learning rate scheduler. Also, all models are trained with a batch size of 8 for 40K iterations. Furthermore, we also use various data augmentation techniques such as color jittering, Gaussian blur, random flipping, and random cropping (crop to 768×768).

Following previous works, we use the 19 semantic categories for training and evaluation and use the mean intersection-over-union (mIoU) of the 19 categories as the evaluation metric. Finally, as the baseline, we use the DeepLabV3+ architecture that is trained without the LoHoSC module and \mathcal{L}_{SC} loss i.e., for baseline, only cross-entropy loss is used during training.

Table 3. Comparison with state-of-the-art methods in the single-source DG setting. All models use the ResNet-101 backbone and are trained with all GTAV sets (train+val+test). “Extra Data” denotes using extra real-world data such as images from ImageNet as auxiliary domains during training.

Methods (GTAV)	Extra Data	mIoU (%)			
		CityScapes	BDD100K	Mapillary	Mean
Baseline	✗	32.97	30.77	30.68	31.47
ISW [4]	✗	37.20	33.36	35.57	35.38
IBN-Net [19]	✗	37.37	34.21	36.81	36.13
Baseline + AdvStyle [37]	✗	39.52	36.39	36.10	37.34
DRPC [33]	✓	42.53	38.72	38.05	39.77
ISW + AdvStyle [37]	✗	43.44	40.32	41.96	41.91
IBN-Net + AdvStyle [37]	✗	44.04	39.96	42.67	42.22
FSDR [11]	✓	44.80	41.20	43.40	43.13
SHADE [36]	✗	46.66	43.66	45.50	45.27
TLDR [15]	✗	47.58	44.88	48.80	47.08
LoSC	✗	45.62	43.25	48.56	45.81
HoSC	✗	46.13	44.10	48.89	45.04
LoHoSC	✗	48.94	45.93	50.28	48.38

4.2 Comparison with State-of-the-Art Methods

Single-Source Setting. In this section, we compare our proposed method with previous works under the single source setting where all models are trained using the GTAV dataset. Table 2 compares the performance of our approach using the DeepLabV3+ architecture with the ResNet-50 backbone and using the GTAV train set for training. We observe that, in terms of the mIOU metric, LoHoSC surpasses the baseline on all three real-world datasets by a considerable margin- Cityscapes by 17.78%, BDD100K by 18.54%, and Mapillary by 20.25%. Also, LoHoSC outperforms TLDR [15], the recent state-of-the-art method by a good margin- Cityscapes by 0.22%, BDD100K by 1.1%, and Mapillary by 2.25%. Furthermore, using just HoSC or LoSC module surpasses or achieves comparable performance to previous state-of-the-art methods, especially on the Mapillary dataset, where they improve mIOU by a good margin.

Table 3 compares the performance of our approach using the DeepLabV3+ architecture with the ResNet-101 backbone and all images (train+val+test) from the GTAV dataset are used for training. We observe that, like Table 2, our LoHoSC variant surpasses the baseline method by a considerable margin- Cityscapes by 15.97%, BDD100K by 15.16%, and Mapillary by 19.6%. Also, LoHoSC outperforms TLDR [15] by a good margin, Cityscapes by 1.36%, BDD100K by 1.05%, and Mapillary by 1.48%, a recent state-of-the-art method. Furthermore, the performance of HoSC and LoSC modules individually is consistent with our observation in Table 2.

Table 4. Comparison with state-of-the-art methods in the multi-source DG setting. All models use ResNet-50 backbone and are trained with GTAV and SYNTHIA train sets.

Methods (G+S)	mIoU (%)			
	CityScapes	BDD100K	Mapillary	Mean
Baseline	35.46	25.09	31.94	30.83
IBN-Net [19]	35.55	32.18	38.09	35.27
ISW [4]	37.69	34.09	38.49	36.75
ISW + Advstyle [37]	39.29	39.26	41.14	39.90
SHADE [36]	47.43	40.30	47.60	45.11
LoSC	45.01	43.88	47.48	45.46
HoSC	45.89	44.40	47.50	45.93
LoHoSC	47.84	45.19	48.59	47.20

Multi-source Setting. In this section, we evaluate our proposed approach under the multi-source setting where all models are trained using the train set of GTAV and SYNTHIA datasets. Table 4 compares the performance of our approach using the DeepLabV3+ architecture with the ResNet-50 backbone. We observe that in comparison to the single-source setting, the performance of all methods improves when multiple domains are used for training. This suggests that the diverse styles from different domains provide informative features for improving the generalization capabilities of models. Regarding the mIOU metric, LoHoSC surpasses the baseline method on real-world datasets by a considerable margin- Cityscapes by 12.38%, BDD100K by 20.1%, and Mapillary by 16.65%. Also, LoHoSC outperforms SHADE [36] by a good margin, Cityscapes by 0.41%, BDD100K by 4.89%, and Mapillary by 0.99%, a recent another state-of-the-art method.

Qualitative Results. Figures 3, 4, and 5 compares the predicted segmentation maps produced by LoHoSC with previous works on CityScapes, BDD100K, and Mapillary dataset, respectively. Figure 2 also provides zoom-in version of segmentation maps to highlight the differences. We select one state-of-the-art Domain Normalization-based method, i.e., ISW [4], and one state-of-the-art Domain Randomization-based method, i.e., TLDR [15] for comparison. All methods are trained on the GTAV dataset with the ResNet-50 backbone. We observe that LoHoSC produces fine-quality segmentation maps, especially at object borders, in comparison to other methods.

4.3 Ablation Study: Location of MixStyle and EFDMix Modules

In this section, we study the impact of inserting MixStyle and EFDMix modules at various locations in the ResNet-50 backbone. In particular, we investigate the placement of these modules at four locations in the ResNet-50 architecture which are denoted as **Layer 0**, **Layer 1**, **Layer 2**, and **Layer 3** in the order of increas-

Table 5. Ablation study on the location of MixStyle and EFDMix modules in the ResNet backbone. All models use ResNet-50 backbone and are trained with GTAV train set.

Location	mIoU (%)			
	CityScapes	BDD100K	Mapillary	Mean
Layer 0	45.41	42.53	47.31	45.08
Layer 0,1	46.46	43.09	48.28	45.94
Layer 0,1,2	46.73	43.68	48.43	46.28
Layer 0,1,2,3	40.36	37.09	34.91	37.45

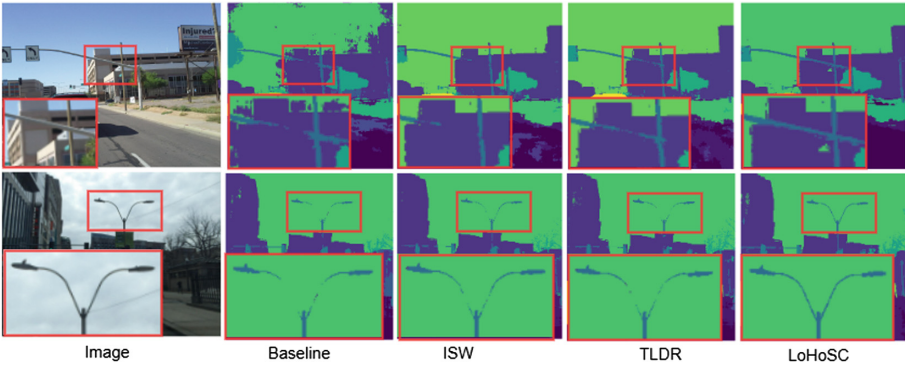


Fig. 2. Qualitative comparison of segmentation results. The regions are zoomed in to highlight the superiority of our method.

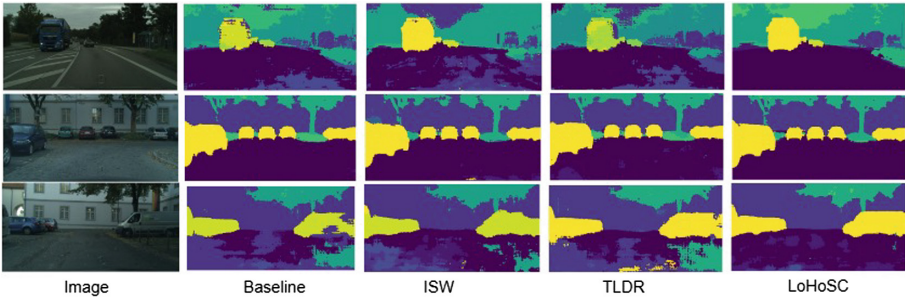


Fig. 3. Qualitative comparison of segmentation results on the CityScapes dataset.

ing depth. Table 5 shows the results of our study. We observe that applying the modules in shallow layers i.e. Layer 0, Layer 1, and Layer 2 produces segmentation models that generalize very well. However, when applying the modules in deeper layers such as Layer 3, the performance drops drastically. Following [36], we attribute two reasons for this phenomenon. First, low order statistics such as the mean and standard deviation represent style information dominantly in

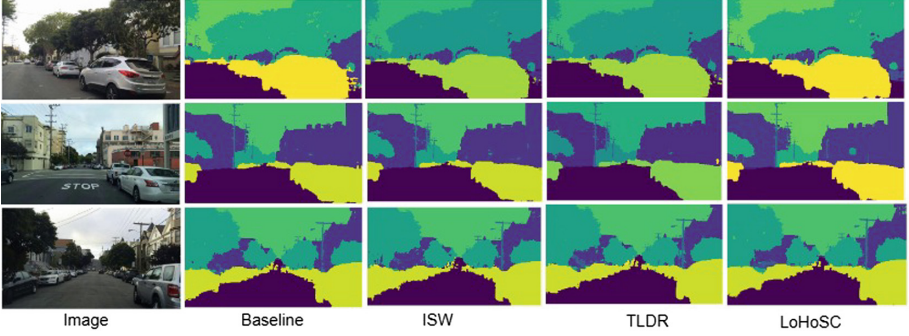


Fig. 4. Qualitative comparison of segmentation results on the BDD100K dataset.

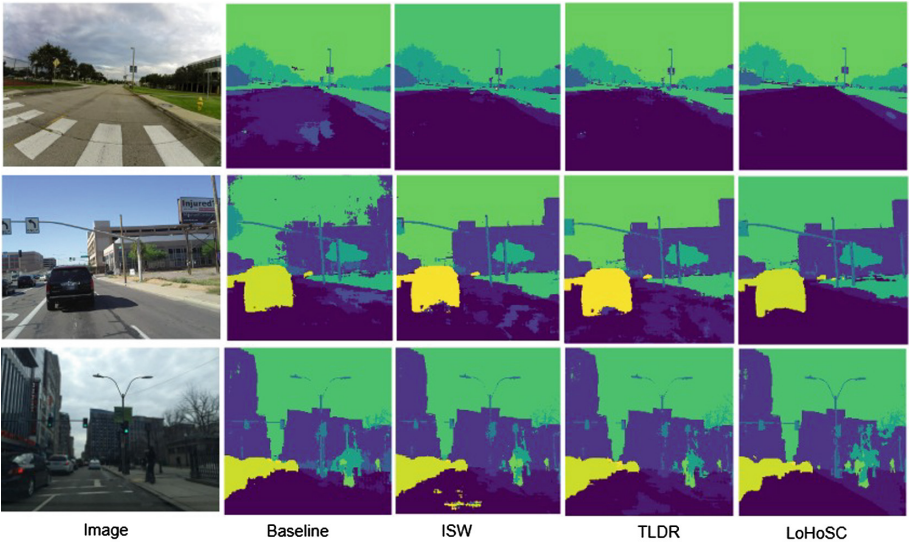


Fig. 5. Qualitative comparison of segmentation results on the Mapillary dataset.

the shallow layers. However, in deeper layers, they are dominant in representing semantic information [8, 13]. Second, in ResNet, the residual connections often lead to large peaks and small entropy in deeper layers, making the style features localized instead of encoding global style [27].

5 Conclusion

In this paper, we propose a new Domain Randomization method called LoHoSC for syn-to-real Domain Generalized Semantic Segmentation. An important feature of LoHoSC is that it uses both low and high-order statistics to generate new styles during training, enabling it to generate styles for a wide range of

complex distributions of real-world images. Our evaluation of LoHoSC shows that it achieves state-of-the-art performance on various benchmark real-world target datasets under single-source and multi-source settings.

References

1. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(4), 834–848 (2017)
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 801–818 (2018)
3. Choi, S., Das, D., Choi, S., Yang, S., Park, H., Yun, S.: Progressive random convolutions for single domain generalization. *arXiv preprint [arXiv:2304.00424](https://arxiv.org/abs/2304.00424)* (2023)
4. Choi, S., Jung, S., Yun, H., Kim, J.T., Kim, S., Choo, J.: Robustnet: improving domain generalization in urban-scene segmentation via instance selective whitening. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11580–11590 (2021)
5. Coltuc, D., Bolon, P., Chassery, J.M.: Exact histogram specification. *IEEE Trans. Image Process.* **15**(5), 1143–1152 (2006)
6. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223 (2016)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE (2009)
8. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. In: *International Conference on Learning Representations* (2016)
9. Hall, E.L.: Almost uniform distributions for computer image enhancement. *IEEE Trans. Comput.* **100**(2), 207–208 (1974)
10. Han, L., Zheng, T., Zhu, Y., Xu, L., Fang, L.: Live semantic 3d perception for immersive augmented reality. *IEEE Trans. Visual Comput. Graphics* **26**(5), 2012–2022 (2020)
11. Huang, J., Guan, D., Xiao, A., Lu, S.: FSDR: frequency space domain randomization for domain generalization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6891–6902 (2021)
12. Huang, L., Zhou, Y., Zhu, F., Liu, L., Shao, L.: Iterative normalization: beyond standardization towards efficient whitening. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4874–4883 (2019)
13. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510 (2017)
14. Kim, N., Son, T., Pahk, J., Lan, C., Zeng, W., Kwak, S.: Wedge: web-image assisted domain generalization for semantic segmentation. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9281–9288. IEEE (2023)
15. Kim, S., Kim, D.H., Kim, H.: Texture learning domain randomization for domain generalized segmentation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 677–687 (2023)

16. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571. IEEE (2016)
17. Mousavian, A., Toshev, A., Fišer, M., Košecká, J., Wahid, A., Davidson, J.: Visual representations for semantic target driven navigation. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 8846–8852. IEEE (2019)
18. Neuhold, G., Ollmann, T., Rota Bulò, S., Kotschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4990–4999 (2017)
19. Pan, X., Luo, P., Shi, J., Tang, X.: Two at once: enhancing learning and generalization capacities via ibn-net. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 464–479 (2018)
20. Pan, X., Zhan, X., Shi, J., Tang, X., Luo, P.: Switchable whitening for deep representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1863–1871 (2019)
21. Peng, D., Lei, Y., Hayat, M., Guo, Y., Li, W.: Semantic-aware domain generalized segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2594–2605 (2022)
22. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: ground truth from computer games. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, 11–14 October 2016, Proceedings, Part II, 14, pp. 102–118. Springer (2016)
23. Rolland, J.P., Vo, V., Bloss, B., Abbey, C.K.: Fast algorithms for histogram matching: application to texture synthesis. *J. Electron. Imaging* **9**(1), 39–45 (2000)
24. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3234–3243 (2016)
25. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, 7–13 October 2012, Proceedings, Part V, 12, pp. 746–760. Springer (2012)
26. Tanzi, L., Piazzolla, P., Porpiglia, F., Vezzetti, E.: Real-time deep learning semantic segmentation during intra-operative surgery for 3d augmented reality assistance. *Int. J. Comput. Assist. Radiol. Surg.* **16**(9), 1435–1445 (2021)
27. Wang, P., Li, Y., Vasconcelos, N.: Rethinking and improving the robustness of image style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 124–133 (2021)
28. Wang, Z., Luo, Y., Qiu, R., Huang, Z., Baktashmotlagh, M.: Learning to diversify for single domain generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 834–843 (2021)
29. Wu, Z., Wu, X., Zhang, X., Ju, L., Wang, S.: Siamdodge: domain generalizable semantic segmentation using siamese network. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, 23–27 October 2022, Proceedings, Part XXXVIII, pp. 603–620. Springer (2022)
30. Xu, Q., Zhang, R., Zhang, Y., Wang, Y., Tian, Q.: A Fourier-based framework for domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14383–14392 (2021)
31. Xu, Z., Liu, D., Yang, J., Raffel, C., Niethammer, M.: Robust and generalizable visual representation learning via random convolutions. In: International Conference on Learning Representations (2021)

32. Yu, F., et al.: Bdd100k: a diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2636–2645 (2020)
33. Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K., Gong, B.: Domain randomization and pyramid consistency: simulation-to-real generalization without accessing target domain data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2100–2110 (2019)
34. Zhang, Y., Li, M., Li, R., Jia, K., Zhang, L.: Exact feature distribution matching for arbitrary style transfer and domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8035–8045 (2022)
35. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890 (2017)
36. Zhao, Y., Zhong, Z., Zhao, N., Sebe, N., Lee, G.H.: Style-hallucinated dual consistency learning for domain generalized semantic segmentation. In: Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, 23–27 October 2022, Proceedings, Part XXVIII, pp. 535–552. Springer (2022)
37. Zhong, Z., Zhao, Y., Lee, G.H., Sebe, N.: Adversarial style augmentation for domain generalized urban-scene segmentation. *Adv. Neural. Inf. Process. Syst.* **35**, 338–350 (2022)
38. Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain generalization with mixstyle. arXiv preprint [arXiv:2104.02008](https://arxiv.org/abs/2104.02008) (2021)
39. Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J.: Unet++: a nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, 20 September 2018, Proceedings 4, pp. 3–11. Springer (2018)



Incorporating Spatial Locality Into Self-attention for Training Vision Transformer on Small-Scale Datasets

Yuki Igaue^(✉), Takio Kurita, and Hiroaki Aizawa

Graduate School of Advanced Science and Engineering Hiroshima University,
Higashi-Hiroshima, Japan
yuki_0304@outlook.com, {tkurita,hiroaki-aizawa}@hiroshima-u.ac.jp

Abstract. Recognizing objects in images requires capturing both global and local visual features. Vision Transformer (ViT) learns to extract these features from large-scale datasets. However, it is known that ViT struggles with local feature extraction on small-scale datasets, such as CIFAR-10, Tiny ImageNet, and ImageNet-100, which are smaller than ImageNet-1k, resulting in poor performance. To resolve this issue, we introduce a novel self-attention mechanism incorporating spatial locality constraints in the attention calculation. Specifically, our method performs the self-attention process independently in spatially divided regions. This constraint limits the receptive fields of attention and forces the local feature extraction. In addition, we provide a global attention path to compute self-attention from the entire image to achieve local and global feature extraction. Experiments demonstrate our approach outperforms ViT and enhanced variants of ViT with original self-attention on various standard small-scale datasets. Moreover, we evaluated the characteristics of learned attention weights using mean attention distance and found that our attention mechanism allows us to extract not only global features but also local features in all blocks.

Keywords: Image Recognition · Vision Transformer · Self-Attention

1 Introduction

Transformer [22] has garnered attention in the field of natural language processing because it achieves state-of-the-art performance in various tasks, including machine translation, using only the attention mechanism [3] and without using any recurrence or convolutions. Since the attention mechanism has weak inductive biases regarding data implicitly included in learning algorithms and learning models, Transformer can also handle tasks other than machine translation and can be applied in various fields.

Vision Transformer (ViT) [6] is a Transformer-based model for the computer vision field, and it outperforms Convolutional Neural Networks (CNNs) in various image recognition tasks such as image classification, object detection,

and semantic segmentation. To achieve the performance of ViT, it is necessary to perform pre-training on large-scale datasets. The original paper [6] empirically demonstrated that when using small-scale datasets for pre-training, such as ImageNet-1k [5], compared to using the large-scale datasets, such as JFT-300M [17], ViT underperforms the CNN-based model [9]. This is because CNNs have inductive biases specific to image data, such as locality, 2D neighborhood structures, and translation invariance, while the attention mechanism used in ViT, which calculates the contextual relationship between patches using dot product operation, provides a low inductive bias specific to image data. As a result, ViT requires the pre-training phase with large-scale datasets to learn the inductive bias from the data itself. Therefore, it is challenging to train ViT on datasets smaller than ImageNet-1k.

To understand the ViT and its training dynamics, the paper [16] examines how ViT extracts features to achieve performance comparable to or better than CNNs. They investigated the characteristics of its feature extraction capabilities based on the mean attention distance, which is analogous to the receptive field size of a CNN and is the average distance in image space over which information is integrated based on attention weights. From this comprehensive study, they found that when using a large-scale dataset for pre-training, ViT can capture both local and global features in the shallow blocks, while only global features are extracted in the deeper blocks. Interestingly, pre-training using a small-scale dataset makes extracting local features in all blocks difficult. To alleviate the difficulty of the local feature extraction on small-scale datasets, many researchers have proposed sophisticated approaches, such as self-supervised learning [7] and modification of the attention mechanism [12].

In this paper, we tackle the problem of training ViT from scratch using small-scale datasets that are far smaller than ImageNet-1k [5]. We call the size of a dataset with tens to hundreds of thousands of samples, such as CIFAR-10 [10], Tiny ImageNet [11], and ImageNet-100 [19], a small-scale dataset in this paper. Our hypothesis is that the capabilities of local feature extraction are key to the success of ViT on small-scale datasets. Thus, we introduce a novel attention method that can acquire local features in all blocks, even when using small-scale datasets. Specifically, the proposed attention spatially divides the input feature into several regions, independently performs self-attention in each divided region, and then aggregates the local attentional features. In addition, we design a path that extracts global features in the same way as ViT’s self-attention in order to extract global features from the entire image. Moreover, the path that extracts local features can also be attached to the attention part in enhanced ViT variants and functions as an adapter. Experiments demonstrate that our approach outperforms traditional ViT self-attention on various small-scale datasets, such as CIFAR-10, Tiny ImageNet, and ImageNet-100. Moreover, we demonstrate that our approach outperforms conventional methods on CaiT [20], PiT [8] and T2T-ViT [23], which are enhanced variants of ViT.

Our main contributions are summarized as follows:

- We propose a novel attention mechanism incorporating locality constraints when calculating attention. This mechanism allows us to learn the capabilities of local feature extraction even when training ViT on small-scale datasets.
- The forward path to extract local features works as an adapter to add local information to the model in various enhanced variants of ViT.
- We evaluate the average attention distance to reveal that our attention mechanism extracts not only global features but also local features in all blocks.

2 Related Work

2.1 Attention Mechanism

The attention mechanism [3] is a mechanism for focusing important information by capturing the relationships between tokens of the input sequence. Note that “token” is the input sequence divided into the smallest units used in the model.

Scaled dot-product attention [22] is one of the attention mechanisms used in Transformer and captures the relationship between tokens using the dot product. Specifically, this is an operation in which input information called “value” is weighted and retrieved by a similarity (attention weight) based on the dot product of a certain token called “query” and another token called “key”. Multi-head self-attention [22] is a method that captures the relationships between various tokens by considering scaled dot-product attention as one head and generating multiple attention weights.

First, we formulate the multi-head self-attention. Given the input $\mathbf{x} \in \mathbb{R}^{N \times D}$, by applying three linear layers $\mathbf{W}_Q \in \mathbb{R}^{D \times d_{QK}}$, $\mathbf{W}_K \in \mathbb{R}^{D \times d_{QK}}$ and $\mathbf{W}_V \in \mathbb{R}^{D \times d_V}$ to \mathbf{x} , three types of vectors, query $\mathbf{Q} \in \mathbb{R}^{N \times d_{QK}}$, key $\mathbf{K} \in \mathbb{R}^{N \times d_{QK}}$, and value $\mathbf{V} \in \mathbb{R}^{N \times d_V}$ are generated as follows:

$$\begin{aligned} \mathbf{Q} &= \mathbf{x}\mathbf{W}_Q, \\ \mathbf{K} &= \mathbf{x}\mathbf{W}_K, \\ \mathbf{V} &= \mathbf{x}\mathbf{W}_V, \end{aligned} \tag{1}$$

where N is the number of tokens of the input \mathbf{x} , D is the dimensions of the token vector, d_{QK} is the dimensions of the query and key, and d_V is the dimensions of the value. Note that generally $D = d_{QK} = d_V$ holds.

Next, each query, key, and value is further divided in the direction of the channel by the number of heads h . Therefore, the query, key, and value of the i -th head are respectively $\mathbf{Q}_i \in \mathbb{R}^{N \times d_{QK}/h}$, $\mathbf{K}_i \in \mathbb{R}^{N \times d_{QK}/h}$, and $\mathbf{V}_i \in \mathbb{R}^{N \times d_V/h}$.

Furthermore, after scaling the matrix product $\mathbf{Q}_i\mathbf{K}_i^T \in \mathbb{R}^{N \times N}$ by $\sqrt{d_{QK}/h}$ and applying the softmax function, the attention weight of the i -th head $\mathbf{A}_i \in \mathbb{R}^{N \times N}$ is obtained as follows:

$$\mathbf{A}_i = \text{softmax} \left(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{d_{QK}/h}} \right). \tag{2}$$

In addition, by calculating the matrix product of attention weight \mathbf{A}_i and the value \mathbf{V}_i , the scaled dot-product self-attention of the i -th head $\text{SA}(\mathbf{x})_i = \mathbf{A}_i \mathbf{V}_i \in \mathbb{R}^{N \times d_V/h}$ is calculated as follows:

$$\text{SA}(\mathbf{x})_i = \mathbf{A}_i \mathbf{V}_i = \text{softmax} \left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_{QK}/h}} \right) \mathbf{V}_i. \quad (3)$$

Then, after combining the scaled dot-product self-attention in the head direction, by applying one linear layer $\mathbf{W}_{\text{out}} \in \mathbb{R}^{d_V \times D}$, the multi-head self-attention $\text{MHSA}(\mathbf{x}) \in \mathbb{R}^{N \times D}$ is calculated, and this is used as the output

$$\text{MHSA}(\mathbf{x}) = \text{Concat}(\text{SA}(\mathbf{x})_1, \dots, \text{SA}(\mathbf{x})_h) \mathbf{W}_{\text{out}}. \quad (4)$$

2.2 Self-attention Mechanism for Local Features

Various models have been proposed that extend ViT, which captures the global features of an image, so that it can sufficiently capture the local features of the image [4, 13, 21]. Among them, Swin Transformer [13] uses two types of self-attention: Window-based Multi-head Self-Attention (W-MSA) and Shifted Window-based Multi-head Self-Attention (SW-MSA). W-MSA uses a rectangular region called the window, divides the patch evenly from the top left corner, and calculates self-attention only for the patches included in the window. When the window size is $M \times M$, one window contains M^2 patches. SW-MSA calculates self-attention using a window shifted by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ from the window position in W-MSA in order to capture the relationship between adjacent patches that belong to different windows. Note that in SW-MSA, since patches that were located far away on the input feature map before the shift are stored in windows other than the left upper one, attention masks are used to set the attention weight value to 0 for that area. Then, these two types of self-attention are applied alternately in the Swin Transformer encoder.

3 Method

3.1 Overall Architecture

We design a neural network architecture based on ViT [6] to confirm the acquisition of local features by devising attention. As shown in Fig. 1, we first divide the input $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ into $P \times P$ patches and flatten them to obtain $\mathbf{x}_p \in \mathbb{R}^{N_p \times (P^2 \cdot 3)}$. Then, linear projection is performed using the embedding layer $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{(P^2 \cdot 3) \times D}$ to generate the patch embedding $\mathbf{x}_{\text{patch}} \in \mathbb{R}^{N_p \times D}$. This process is defined as follows:

$$\mathbf{x}_{\text{patch}} = \text{Concat}(\mathbf{x}_p^1 \mathbf{W}_{\text{emb}}, \dots, \mathbf{x}_p^{N_p} \mathbf{W}_{\text{emb}}), \quad (5)$$

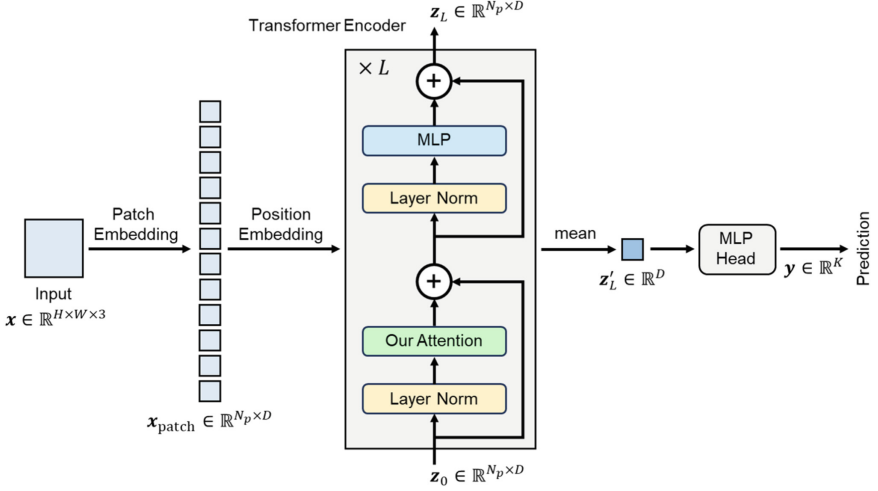


Fig. 1. Overview of ViT architecture with our attention.

where $N_p (= HW/P^2)$ is the number of patches, and $\mathbf{x}_p^i \in \mathbb{R}^{P^2 \cdot 3}$ is the i -th patch. Next, we add the positional embedding $\mathbf{W}_{\text{pos}} \in \mathbb{R}^{N_p \times D}$ to $\mathbf{x}_{\text{patch}}$, making it the input $\mathbf{z}_0 \in \mathbb{R}^{N_p \times D}$ to the Transformer encoder. \mathbf{z}_0 is obtained from

$$\mathbf{z}_0 = \mathbf{x}_{\text{patch}} + \mathbf{W}_{\text{pos}}. \quad (6)$$

Our design of the encoder block is based on the architecture [6]. We modify the Multi-Head Self-Attention (MHSA) part into our attention method. Therefore, our Transformer encoder is composed of multiple encoder blocks consisting of layer normalization [2], our attention mechanism, and a multilayer perceptron (MLP). When the Transformer encoder is composed of L encoder blocks, the output $\mathbf{z}_l \in \mathbb{R}^{N_p \times D}$ of the l -th encoder block is expressed as follows:

$$\mathbf{z}'_l = \text{Attention}(\text{LN}(\mathbf{z}_{l-1})) + \mathbf{z}_{l-1}, \quad (7)$$

$$\mathbf{z}_l = \text{MLP}(\text{LN}(\mathbf{z}'_l)) + \mathbf{z}'_l, \quad (8)$$

where $\text{LN}(\cdot)$ is layer normalization, $\text{Attention}(\cdot)$ is our attention, and $\text{MLP}(\cdot)$ is the MLP. Finally, we input $\mathbf{z}'_L \in \mathbb{R}^D$, which is the average of the output of the Transformer encoder \mathbf{z}_L in the patch direction, to the MLP head. The MLP head consists of layer normalization and one linear layer $\mathbf{W}_{\text{head}} \in \mathbb{R}^{D \times K}$, where K is the number of classes to be classified. The output of the MLP head $\mathbf{y} \in \mathbb{R}^K$ is expressed as follows:

$$\mathbf{y} = \text{LN}(\mathbf{z}'_L) \mathbf{W}_{\text{head}}. \quad (9)$$

3.2 Self-attention for Local Feature Extraction

We propose an attention method for enhancing local feature extraction as shown in Fig. 2. First, the input $\mathbf{z} \in \mathbb{R}^{N_p \times D}$ is transformed into $\mathbf{z}' \in \mathbb{R}^{H/P \times W/P \times D}$ so

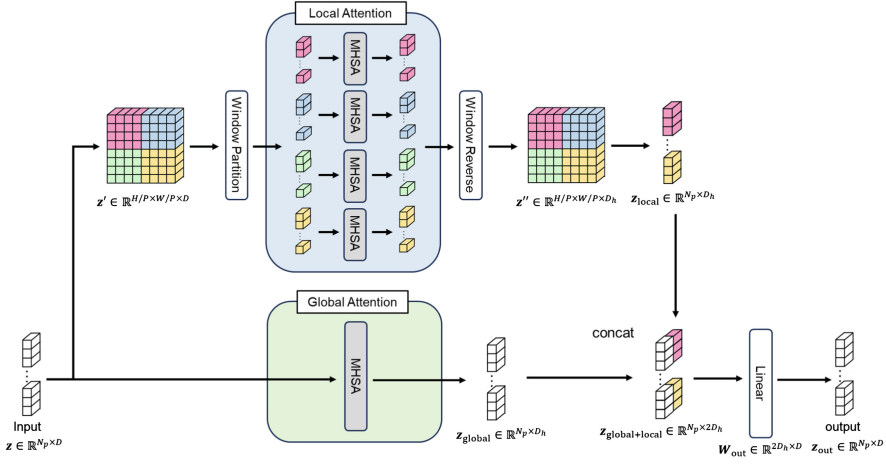


Fig. 2. Our attention.

that each patch has the same arrangement as the input image. Next, we apply window partition(WP) to z' and use an $M \times M$ window to divide the patches. Based on the paper [13], there are two types of windows used at this time: window W_A , which is divided evenly from the upper left patch, and window W_B , which is shifted by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ from the position of window W_A as shown in Fig. 3. Note that our method differs from the paper [13] in that we split the input into four windows with window size $M \times M = (H/2P) \times (W/2P)$. Since the number of windows remains the same regardless of whether W_A or W_B is used when the i -th window is $w_i \in \mathbb{R}^{(H/2P) \times (W/2P) \times D}$,

$$WP(z') = w_1, w_2, w_3, w_4. \tag{10}$$

Then, after applying MHSA for each window (hereinafter, this is called local attention), we apply window reverse(WR) to combine the windows ($z'' \in \mathbb{R}^{H/P \times W/P \times D_h}$). Furthermore, z'' is flattened to generate $z_{local} \in \mathbb{R}^{N_p \times D_h}$. Therefore, z_{local} is expressed as follows:

$$z_{local} = \text{Flatten}(\text{WR}(\text{MHSA}(w_1), \text{MHSA}(w_2), \text{MHSA}(w_3), \text{MHSA}(w_4))). \tag{11}$$

Finally, the output is $z_{out} \in \mathbb{R}^{N_p \times D}$, in which one linear layer $W_{out} \in \mathbb{R}^{2D_h \times D}$ is applied to the combination of $z_{global} \in \mathbb{R}^{N_p \times D_h}$, which is the result of applying MHSA to the input z (hereinafter, this is called global attention) and z_{local} in the channel direction. Thus, z_{out} is expressed as follows:

$$z_{out} = \text{Concat}(z_{global}, z_{local})W_{out}. \tag{12}$$

Note that, as shown in Fig. 4, our method does not apply the final linear layer in MHSA. Hence, using equation(13), $\text{MHSA}(x)$ is the combination of $\text{SA}(x)_i$

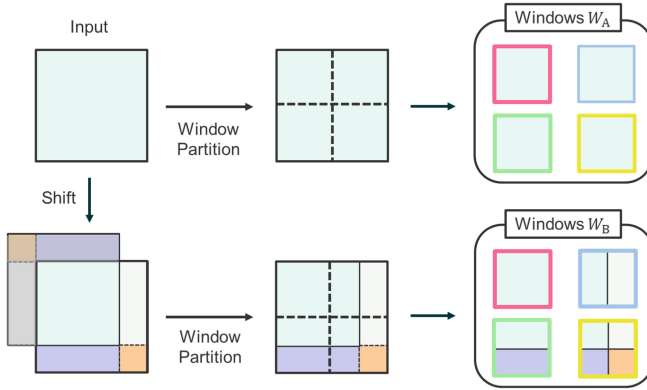


Fig. 3. Two types of windows in window partition. (Color figure online)

in the direction of the head as shown below:

$$\text{MHSA}(\mathbf{x}) = \text{Concat}(\text{SA}(\mathbf{x})_1, \dots, \text{SA}(\mathbf{x})_h). \quad (13)$$

4 Experiments

4.1 Experimental Setup

All models were trained for 500 epochs with a batch size of 256 and an initial learning rate of 0.001. We used AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay=0.05) [14] with the cosine learning rate scheduler [15]. Note that warm-up was not performed in the cosine learning rate scheduler.

We used CIFAR-10 [10] with 10 classes and 60k images, Tiny ImageNet [11] with 200 classes and 110k images, and ImageNet-100 [19] with 100 classes and 135k images. ImageNet-100 is a subset of the ILSVRC-2012 ImageNet [5] dataset. For training, we resized the shorter side to 256 and cropped the center to 256×256 , then applied RandomCrop, RandomHorizontalFlip, Mixup [25], Cut-Mix [24], and RandomErasing [26] to obtain 224×224 images. For evaluation, we resized the shorter side to 256, then cropped the center to 224×224 . Furthermore, we applied label smoothing [18] to the labels and used the cross entropy loss adapted to label smoothing as the loss function:

$$\mathcal{L} = (1 - \epsilon)\mathcal{L}_{\text{CE}}(i) + \epsilon \frac{1}{K} \sum_{j=1}^K \mathcal{L}_{\text{CE}}(j), \quad (14)$$

where ϵ is a parameter, i is the truth label, K is the number of classification classes, and \mathcal{L}_{CE} is the cross-entropy loss, which is expressed as follows:

$$\mathcal{L}_{\text{CE}}(k) = -t_k \log(p_k), \quad (15)$$

where t_k is one-hot vector of the truth label and p_k is the softmax probability for the k -th class.

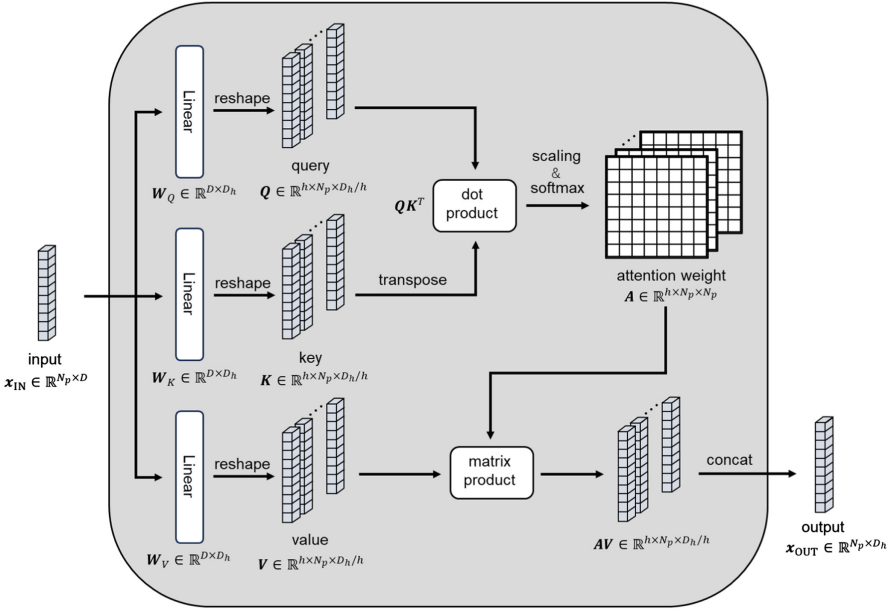


Fig. 4. MHSA in our method.

Table 1. Configuration of the model used in this experiment.

Model	Patch size	Dimensions(D)	Blocks	Heads	Params
ViT-S/16(baseline)	16	384	12	6	21.6M
baseline+	16	384	16	6	28.7M
ViT-S/16 w/ our attention	16	384	12	6	28.7M
Swin-T	4	[96, 192, 384, 768]	[2, 2, 6, 2]	[3, 6, 12, 24]	27.6M
CaiT(XXS-24)	16	192	24+2	4	11.8M
CaiT(XXS-24) w/ our attention	16	192	24+2	4	15.3M
T2T	16	384	2+12	6	21.7M
T2T w/ our attention	16	384	2+12	6	28.8M
PiT-S	16	[144, 288, 576]	[2, 6, 4]	[3, 6, 12]	22.7M
PiT-S w/ our attention	16	[144, 288, 576]	[2, 6, 4]	[3, 6, 12]	30.2M

4.2 Image Classification

Comparison with ViT Model. To evaluate the performance on the standard ViT model, we used ViT-S/16 published in timm¹ as a baseline. As summarized in Table 1, since we changed only the MHSA part in our method, the basic configuration is the same as in the baseline. In addition, two types of baselines

¹ <https://github.com/huggingface/pytorch-image-models>.

Table 2. Comparison of the best score of validation top1-accuracy with baseline. We show the results for the case of using only window W_A in the W_A column and the results for the case of using window W_A and window W_B alternately for each block in the W_A/W_B column.

	baseline	baseline+	ours	
			W_A	W_A/W_B
ImageNet-100	80.48	79.28	81.24	81.60
CIFAR-10	93.77	93.12	94.66	94.94
Tiny ImageNet	60.77	58.05	61.79	61.39

Table 3. Comparison of the best score of validation top1-accuracy using ImageNet-100.

Model	top1-acc
ViT-S/16(baseline)	80.48
ViT-S/16 w/ our attention	81.24
Swin-T	84.18
CaiT(XXS-24)	82.06
CaiT(XXS-24) w/ our attention	83.02
T2T	84.36
T2T w/ our attention	84.94
PiT-S	83.64
PiT-S w/ our attention	83.76

are used: baseline, which differs only in the attention part compared to our attention, and baseline+, which has almost the same number of parameters as our attention by setting the number of blocks to 16.

In our method, we evaluated the case of using only window W_A (hereafter, this is called method W_A) and the case of using window W_A and window W_B alternately for each block (hereafter, this is called method W_A/W_B) in the window partition part of our attention. Note that due to the structure of our attention, method W_A and method W_A/W_B have the same number of parameters.

We compared the performance of baseline, baseline+ and our methods in terms of the best score of validation accuracy with CIFAR-10, Tiny ImageNet and ImageNet-100. The results are summarized in Table 2. Our method outperforms baseline and baseline+ on all datasets. Also, method W_A/W_B is slightly superior when using ImageNet-100 and CIFAR-10, and method W_A is slightly superior when using Tiny ImageNet.

Comparison with the Enhanced Variants of ViT. We compared our method W_A with Swin Transformer [13], CaiT [20], T2T-ViT [23] and PiT [8] as the enhanced variants of ViT using ImageNet-100 dataset. Here, except for T2T-

ViT, we used Swin-T, XXS-24, and PiT-S published in timm¹ as Swin Transformer, CaiT, and PiT, respectively. We also conducted similar experiments with variants that adapted local attention to each attention part. However, the training of the Swin transformer on small-scale datasets is quite unstable, and we needed to change the learning rate of the Swin Transformer from the baseline setting to achieve stable training (initial learning rate: 0.001→0.0005). We were unable to train the Swin Transformer with our attention because training was unstable even after changing the learning rate.

The results are summarized in Table 3. ViT-S/16 with our attention was less accurate than any variant other than ViT-S/16. On the other hand, when comparing the original version and the local attention adapted version with our attention for each variant, our methods outperformed the original variants in CaiT(XXS-24), T2T and PiT-S.

4.3 Analysis with Mean Attention Distance

We used the mean attention distance, the same evaluation index as in the original paper [16], to evaluate the receptive field size in each model. The attention distance is the distance between two patches multiplied by the corresponding attention weight, and the mean attention distance is the average of the sum of the attention distances of each patch. Therefore, it can be said that the smaller the mean attention distance, the more local features are extracted, and the larger the mean attention distance, the more global features are extracted. Now, we will formulate a method for finding the mean attention distance. When the number of patches is N , the attention weight \mathbf{A} is $N \times N$. First, we create a distance matrix $\mathbf{X} \in \mathbb{R}^{N \times N}$ that summarizes the distance between two patches based on the input image in the arrangement corresponding to the attention weight. Next, by taking the Hadamard product of \mathbf{A} and \mathbf{X} , we generate the attention distance matrix $\mathbf{Y} \in \mathbb{R}^{N \times N}$ as follows:

$$\mathbf{Y} = \mathbf{A} \odot \mathbf{X}. \quad (16)$$

Note that \odot represents the Hadamard product, that is, the product of each element of two matrices. In addition, we prepare coordinates that extend the x -axis to the right and the y -axis to the bottom, with the upper left patch of the input image divided into $P \times P$ patches as the origin. At this time, letting any two of the coordinates of the patches be (x_1, y_1) and (x_2, y_2) , respectively, the distance d between the two patches stored in the distance matrix is calculated as follows:

$$d = P \times \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \quad (17)$$

When the input is composed of $N_H \times N_W$ patches, the coordinates of the two patches with the largest distance between them are $(0, 0)$, $(N_W - 1, N_H - 1)$. Therefore, the maximum distance between two patches stored in the distance matrix d_{\max} is as follows:

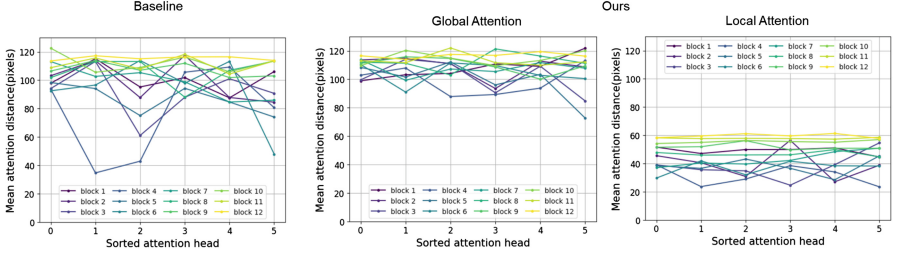


Fig. 5. Comparison of average mean attention distance per head for each block.

$$\begin{aligned}
 d_{\max} &= P \times \sqrt{((N_W - 1) - 0)^2 + ((N_H - 1) - 0)^2} \\
 &= P \times \sqrt{(N_W - 1)^2 + (N_H - 1)^2}.
 \end{aligned} \tag{18}$$

The mean attention distance is calculated by calculating the total attention distance for each patch and then taking the average of them. Therefore, when the element in the i -th row and j -th column of \mathbf{Y} is \mathbf{Y}_{ij} , the mean attention distance y is expressed as follows:

$$y = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \mathbf{Y}_{ij}. \tag{19}$$

We calculated the average of 100 data points for the mean attention distance for each attention head. The results are shown in Fig. 5. Some blocks or heads with small mean attention distances can be seen in the baseline, but there are almost no blocks or heads with small mean attention distances in the global attention of our method. Furthermore, the mean attention distance of global attention is larger, and the mean attention distance of local attention is smaller in all blocks. In this experiment, the input consists of 14×14 patches, the patch size is 16, and the window size is 7. Therefore, from equation (18), the maximum value of the distance between two patches stored in the distance matrix in global attention is $d_{\max}^G = 16 \times \sqrt{(14 - 1)^2 + (14 - 1)^2} \approx 294.16$, and the maximum value of the distance between two patches stored in the distance matrix in local attention is $d_{\max}^L = 16 \times \sqrt{(7 - 1)^2 + (7 - 1)^2} \approx 135.76$. Therefore, it is obvious that the mean attention distance of local attention is less than half of that of global attention.

Considering these facts, it can be said that the global attention of our method extracts more global features than the attention of the baseline, even though it processes the same amount of input as the baseline. On the other hand, we cannot conclude that local attention of our method extracts local features, but since global attention is specialized for extracting global features, it is highly likely that local attention is responsible for extracting local features to supplement it.

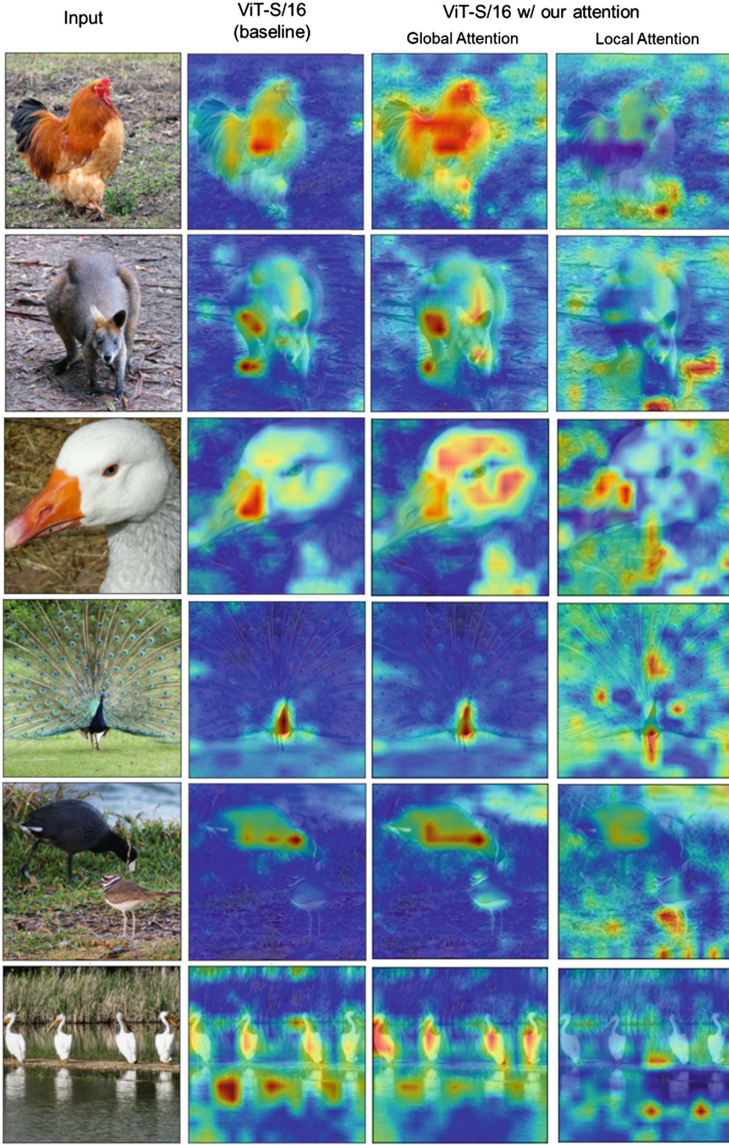


Fig. 6. Comparison of attention maps of baseline (ViT-S/16) and global attention map, local attention map in the case of using method W_A . (Color figure online)

4.4 Visualization of Attention Map

In order to show which patch of the image each model focused on for image classification, we used a method called attention rollout [1] to visualize an attention map that maps the attention weight. We will describe the case of applying atten-

tion rollout in MHSA. Suppose that the input to MHSA is $\mathbf{z} \in \mathbb{R}^{(N_p+1) \times D}$, where $N_p + 1$ is the number of class token and patches. When the Transformer encoder consists of B blocks and the number of heads of MHSA is h , the attention weight of the i -th head of the b -th block is set to \mathbf{A}_i^b ($b = 1, 2, \dots, B, i = 1, 2, \dots, h$). In attention rollout, first, we take the average attention weight for each block. The average attention weight of the b -th block $\overline{\mathbf{A}}^b \in \mathbb{R}^{(N_p+1) \times (N_p+1)}$ is as follows:

$$\overline{\mathbf{A}}^b = \frac{1}{h} \sum_{i=1}^h \mathbf{A}_i^b. \quad (20)$$

Next, we add the identity matrix $\mathbf{I} \in \mathbb{R}^{(N_p+1) \times (N_p+1)}$ to $\overline{\mathbf{A}}^b$ and multiply it sequentially starting from the first block. The average of the entire attention weight $\overline{\mathbf{A}} \in \mathbb{R}^{(N_p+1) \times (N_p+1)}$ is obtained as follows:

$$\overline{\mathbf{A}} = \prod_{b=1}^B (\overline{\mathbf{A}}^b + \mathbf{I}). \quad (21)$$

Furthermore, we extract the attention weight between the class token and each patch $\overline{\mathbf{A}}_{\text{cls}} \in \mathbb{R}^{N_p}$ from matrix $\overline{\mathbf{A}}$. Finally, each patch is transformed so that it has the same layout as the input image, and is resized to the same size as the input image to generate an attention map. Note that our model does not use class token, as shown in Fig. 1, so instead of $\overline{\mathbf{A}}_{\text{cls}}$, we use $\hat{\mathbf{A}} \in \mathbb{R}^{N_p}$, which is the average of the entire attention weight $\overline{\mathbf{A}}' \in \mathbb{R}^{N_p \times N_p}$, further averaged in the patch direction. If the element in the i -th row and j -th column of $\overline{\mathbf{A}}'$ is $\overline{\mathbf{A}}'_{ij}$, then $\hat{\mathbf{A}}$ is expressed as follows:

$$\hat{\mathbf{A}} = \frac{1}{N} \sum_{j=1}^N \overline{\mathbf{A}}'_{ij}. \quad (22)$$

In local attention, attention maps are generated for each window. Since window W_A divides the input into four equal parts without shifting it from the top left of the patch, it is possible to visualize the local attention map by combining the attention maps.

Using the trained models in the baseline and our models (method W_A), we generated the attention maps for several picked images. Figure 6 shows the original image and images with an attention map superimposed on the original image. The top two are single-object samples with equal proportions of subject and background in the image, the middle two are single-object samples with a high proportion of subject and almost no background, and the bottom two are multiple-object samples. Note that, as shown in Fig. 2, our attention has two types of attention: global attention and local attention, whereas baseline has only one type of attention.

It can be observed that our global attention is more focused on the background outside the target object compared to the baseline, but more focused on the inside of the target object compared to local attention. This is especially

noticeable in the multiple-object samples. Moreover, if we focus on the middle two samples where the proportion of the background is small, local attention focuses on parts or patterns of the target object that global attention does not capture.

4.5 Ablation Study

We conducted ablation studies regarding the method for merging global and local feature vectors and the number of heads. Note that all experiments are performed based on method W_A , and ImageNet-100 is used as the dataset. The results are summarized in Table 4.

Feature Merging. As mentioned in Sect. 3.2, in our attention, we finally concatenated the global attention output $\mathbf{z}_{\text{global}} \in \mathbb{R}^{N_p \times D}$ and the local attention output $\mathbf{z}_{\text{local}} \in \mathbb{R}^{N_p \times D}$ in the direction of the channel and applied one linear layer $\mathbf{W}_{\text{out}} \in \mathbb{R}^{2D \times D}$ to merge the features. We performed ablation by changing the concatenating part in the channel direction to a simple addition. Note that since the dimensions of the addition of $\mathbf{z}_{\text{global}}$ and $\mathbf{z}_{\text{local}}$ is D , the one linear layer applied after the addition is $\mathbf{W}'_{\text{out}} \in \mathbb{R}^{D \times D}$, and the learning parameters are slightly reduced. In addition, in order to confirm whether the improvement in accuracy depends on merging global and local feature vectors, we performed ablation using only local features without feature merging. Note that if only global features are used, it is the same as the baseline.

The results indicated that when using addition, the accuracy improved by 0.58% compared to when using concatenating. However, the accuracy when using method W_A/W_B is 81.58% (−0.02%). Also, when the dataset was CIFAR-10, the accuracy of method W_A is 94.39% (−0.27%), and the accuracy of method W_A/W_B is 94.77% (−0.17%). Therefore, in other cases, using concatenating provides superior performance, and it cannot be concluded that applying addition instead necessarily contributes to better performance. Furthermore, when using only local feature, the accuracy decreased by 4.10% compared to when using concatenating, and also decreased by 3.34% compared to the baseline. Therefore, the improvement in accuracy from our attention method is not due to the use of local features, but to the merging of global and local features.

The Number of Heads. We performed ablation by varying the number of MHSA heads for global attention and all local attention. When the number of global attention heads was unchanged, and the number of local attention heads was set to 3, the accuracy improved by 0.22%. In addition, when the number of global attention heads was set to 8, the accuracy improved by 0.20% regardless of the number of local attention heads. However, when the number of baseline heads was set to 8, the accuracy improved by 0.22%. Therefore, it can be said that the accuracy improvement effect of our attention has hardly changed.

Table 4. Ablation study. The numbers in parentheses shows the change in accuracy from before the change (to the left of the arrow).

Ablation	Variant	Top-1 accuracy(%)
Features merging	Concat \rightarrow Add	81.82(+0.58%)
	Concat \rightarrow None(only local feature)	77.14(-4.10%)
The number of heads	baseline(6) \rightarrow baseline(8)	80.70(+0.22%)
G : Global attention	$(G, L) = (6, 6) \rightarrow (G, L) = (6, 3)$	81.46(+0.22%)
L : Local attention	$(G, L) = (6, 6) \rightarrow (G, L) = (8, 8)$	81.44(+0.20%)
	$(G, L) = (6, 6) \rightarrow (G, L) = (8, 2)$	81.44(+0.20%)

5 Conclusion

We proposed a method to address the issue of ViT, which suffers from relatively poor performance when learning with small datasets. The proposed method is a novel attention method that can extract both global and local features in all blocks, based on the fact that local features are hardly extracted when training is performed using a dataset that is not large. In image classification, our method, which uses both global attention and local attention, outperformed the conventional method that uses only global attention. It has also been demonstrated that local attention can be applied not only to ViT but also to various enhanced variants of ViT, contributing to improving recognition accuracy.

However, for some variants, such as Swin Transformer, the training using small-scale datasets was very unstable, making it difficult to apply the proposed method. Therefore, further modification of the proposed method is necessary to apply it to variants with unique structures such as Swin Transformer. In addition, since we calculated the attention for each window as well as the attention for the entire input in our method, the computational cost was slightly larger than the original attention. Therefore, it is necessary to consider a more computationally efficient method. Furthermore, it is necessary to verify whether our approach is effective for other vision tasks, such as object detection and semantic segmentation.

References

1. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. arXiv preprint [arXiv:2005.00928](https://arxiv.org/abs/2005.00928) (2020)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint [arXiv:1607.06450](https://arxiv.org/abs/1607.06450) (2016)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473 (2014)
4. Chen, C.F., Panda, R., Fan, Q.: Regionvit: Regional-to-local attention for vision transformers. arXiv preprint [arXiv:2106.02689](https://arxiv.org/abs/2106.02689) (2021)

5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
6. Dosovitskiy, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
7. Gani, H., Naseer, M., Yaqub, M.: How to train vision transformer on small-scale datasets? arXiv preprint [arXiv:2210.07240](https://arxiv.org/abs/2210.07240) (2022)
8. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. arXiv preprint [arXiv:2103.16302](https://arxiv.org/abs/2103.16302) (2021)
9. Kolesnikov, A., et al.: Big transfer(bit): General visual representation learning. arXiv preprint [arXiv:1912.11370](https://arxiv.org/abs/1912.11370) (2020)
10. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)
11. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7):3 (2015)
12. Lee, S.H., Lee, S., Song, B.C.: Vision transformer for small-size datasets. arXiv preprint [arXiv:2112.13492](https://arxiv.org/abs/2112.13492) (2021)
13. Liu, Z., et al.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint [arXiv:2103.14030](https://arxiv.org/abs/2103.14030) (2021)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)
15. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint [arXiv:1608.03983](https://arxiv.org/abs/1608.03983) (2017)
16. Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A.: Do vision transformers see like convolutional neural networks? arXiv preprint [arXiv:2108.08810](https://arxiv.org/abs/2108.08810) (2021)
17. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. arXiv preprint [arXiv:1707.02968](https://arxiv.org/abs/1707.02968) (2017)
18. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. arXiv preprint [arXiv:1512.00567](https://arxiv.org/abs/1512.00567) (2015)
19. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. arXiv preprint [arXiv:1906.05849](https://arxiv.org/abs/1906.05849) (2019)
20. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. arXiv preprint [arXiv:2103.17239](https://arxiv.org/abs/2103.17239) (2021)
21. Tu, Z., et al.: Maxvit: Multi-axis vision transformer. arXiv preprint [arXiv:2204.01697](https://arxiv.org/abs/2204.01697) (2022)
22. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
23. Yuan, L., et al.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. arXiv preprint [arXiv:2101.11986](https://arxiv.org/abs/2101.11986) (2021)
24. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. arXiv preprint [arXiv:1905.04899](https://arxiv.org/abs/1905.04899) (2019)
25. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint [arXiv:1710.09412](https://arxiv.org/abs/1710.09412) (2017)
26. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. arXiv preprint [arXiv:1708.04896](https://arxiv.org/abs/1708.04896) (2017)



Cross-Domain Calibration and Boundary Denoising Network for Weakly Supervised Semantic Segmentation

Zhoufeng Liu¹(✉), Bingrui Li¹, Shumin Ding¹, Jiangtao Xi², and Chunlei Li¹

¹ School of Information and Communication Engineering, Zhongyuan University of Technology, Henan 450007, China
lzhoufeng62@163.com

² School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NewSouthWales, Australia

Abstract. Weakly Supervised Semantic Segmentation (WSSS) methods based on image-level labeling alleviate the burden of annotation due to image-level labels only providing object categories. However, the generated Class Activation Maps (CAM) can only localize the most discriminative region rather than the entire object region. To remedy this issue, a framework of WSSS algorithms for Cross-Domain Calibration and Boundary Denoising Network (CD-CBN) is presented. Specifically, a Spatial Feature Calibration Network (SFCN) is proposed to align cross-dimensional features with class prototypes, focusing on intra-class feature consistency. Then, a Class-Specific Distance Model (CSDM) is adopted to separate features from different classes, and feature activation in the object region surpasses the background area. Finally, a Full-domain-aware Noise Reduction Model (FNRM) is designed to refine the object boundary pixels by capturing global contextual features and further filtering out pixel-level noise. A comprehensive experimental evaluation of the highly challenging Pascal VOC 2012 dataset and MS COCO 2014 is presented in this study, illustrating the effectiveness of our suggested approach.

Keywords: Weakly Supervised · Semantic Segmentation · Spatial Feature Calibration · Class Prototypes · Full-Domain-Aware

1 Introduction

Thanks to the rapid advancements in deep neural networks, the field of semantic segmentation has experienced substantial progress. The performance of existing semantic segmentation methods relies on dense pixel-level labels. However, obtaining pixel-level annotation is unbearably expensive for a fully-supervised semantic segmentation network. For instance, the task of annotating each image within the Cityscapes dataset typically requires approximately 90 min [1]. As such, WSSS methods employ image-level annotations and have garnered significant attention due to their crucial practical applications. Consequently, to reduce

the dependence on extensive annotation efforts, researchers are increasingly focusing on weakly supervised learning approaches. “Weak” indicates image-level labels that can be readily acquired, e.g., scribbles [2], bounding boxes [3], points [4], and image-level labels [5]. Among them, image-level labels prove to be the most efficacious, which significantly lowers the cost of data annotation and has become a focal point in current research.

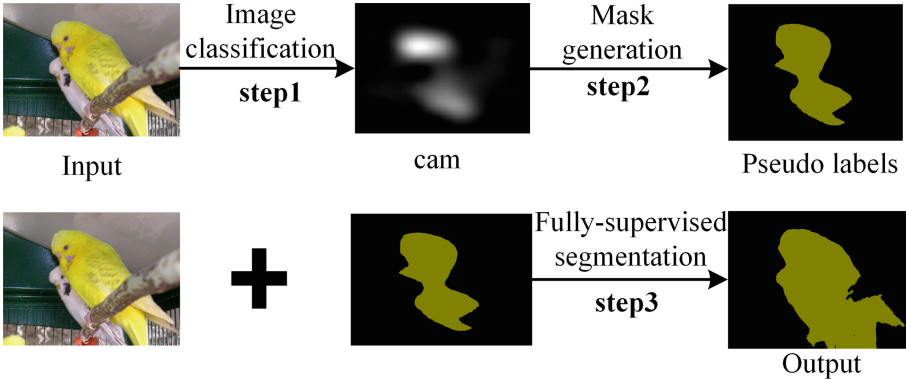


Fig. 1. Two-stage Weakly-supervised semantic segmentation.

The current WSSS methods predominantly rely on CAM as a fundamental approach, as illustrated in Fig. 1. The crucial stage in training a semantic segmentation model involves extracting CAM [6] from a classification model. To be more precise, the overall workflow of WSSS involves three key stages: 1) training a multi-label classification model utilizing the image-level labels; 2) deriving class-specific CAMs to produce binary masks (commonly known as seed masks), which are often refined to generate pseudo-masks; 3) employing all class pseudo-labels to train a fully-supervised model. It is evident that the quality of CAMs obtained in the initial step significantly impacts the performance of the ultimate semantic segmentation network. However, the first challenge is that CAM only identifies the most distinguishing regions of objects. The presence of tiny and sparsely activated areas poses challenges in acquiring pixel-level labels of high quality, and has a higher mean Intersection over Union (mIoU) for pseudo-labels does not necessarily indicate the superiority of the segmentation model. To address this, a prevalent number of subsequent efforts strive to learn more complete object regions, and the second challenge is that classification networks yield noisy pseudo-labels. Inspired by this observation, existing studies on learning from noisy labels primarily concentrate on the classification task, e.g., robust architecture [7], robust regularization [8], loss adjustment [9], and sample selection [10]. However, the above method suffers from incomplete object regions as well as focusing on the noisy pixels generated by the classification network. Given the above-mentioned analysis, we propose

a cross-domain calibration and boundary denoising network (CD-CBN), which can capture integral object regions by combining cross-dimensional features with class prototypes and improve the learning of noisy pixels. Specifically, a spatial feature calibration network (SFCN) is proposed to synchronize salient region features with the prototype of the object class, which fosters the activation of a greater number of object regions within the classification network. In contrast to the current context fusion models, SFCN provides a location feature while maintaining higher performance. Besides, another class-specific distance model (CSDM) [11] is adopted to separate features from different classes and keep feature activation in the background area significantly lower than observed in the object. A full-domain-aware noise reduction model (FNRM) is designed to refine the object boundary pixels by combining pseudo-labels with initial prediction, which captures global contextual features and further filters out pixel-level noise.

Our work has made the following major contributions :

- 1) A cross-domain calibration and boundary denoising network (CD-CBN), which consists three parts (i.e., SFCN, CSDM and FNRM) is presented to extract the complete object area in cross-dimensional feature spaces, and further denoising the pixel-level labels, ensuring sufficient utilization of object features.
- 2) SFCN is elaborately designed to align cross-dimensional features with class prototypes and focus on intra-class feature consistency.
- 3) FNRM is proposed to refine the object boundary information and further filter out pixel-level noise.
- 4) Experimental results on two widely used datasets demonstrate our proposed method has the superior performance compared with other state-of-the-art approaches.

2 Related Work

2.1 Semantic Segmentation

In recent years, semantic segmentation has been extensively studied, with the fully convolutional network [12] emerging as the most distinguished framework. Numerous methods in this field are built upon this foundational structure. For instance, the initial research on SegNet [13] proposed a deep convolutional encoder-decoder structure that was engineered to restore the spatial resolution of the input image. Lin et al. [14] proposed a versatile multipath refinement network aimed at generating high-resolution segmentation maps, which employs a recursive merging method that combines low-resolution semantic features with fine-grained low-level features in a hierarchical manner. Recently, the concept of self-attention has been investigated in [15], emphasizing the adaptive integration of local features while considering their global dependencies. The research in [15] utilized non-local neural networks for capturing distant correlations, whereas DANet [16] proposed both position and channel-level awareness models, designed

to understand the spatial and channel-wise interrelations of feature representations. CCNet [17] proposed an innovative criss-cross attention model aimed at gathering contextual information along a criss-cross path. Nonetheless, due to semantic gaps between feature maps of varying levels, where high activation regions contain more semantic details, methods like aggregation of features or concatenation yield suboptimal results. Compared to the aforementioned methods, we focus on capturing the complete object area in cross-dimensional feature spaces and further denoising the pixel-level labels.

2.2 Weakly Supervised Semantic Segmentation

Approaches in WSSS have been developed to mitigate the high costs associated with labeling in fully-supervised semantic segmentation, and which resorts to solving the problem of expensive labels. The majority of current WSSS methods employ a multi-stage process [5]. Among these methods, image-level labels are commonly utilized as a form of weak supervision [5] due to their easy availability. The initial phase of segmentation involves employing image-level labels and creating CAMs for the training datasets. Subsequently, pseudo-masks can be derived from these CAMs. Given the rough localization of object regions by CAMs, the resulting pseudo-masks often lack precision. To mitigate this issue, several methods employ extra labels or data sources (such as saliency [11,31], and [33]) for enhanced supervision in the training phase. However, the use of saliency maps is primarily aimed at identifying the edges of prominent areas rather than objects. Thus, we leverage image-level labels as a means of supervision in our approach, and tackle the challenges of both background and object incompleteness.

2.3 Sturdy Training Under Noisy Label

Handling noisy labels is a crucial task [18] in the field of machine learning, which can be divided into four distinct categories based on their underlying approaches: robust architectural design, robust regularization strategies, robust loss design, and selective sampling. Various robust architectures [7] have been proposed to effectively model the noise transition matrix in datasets with label noise. But these methods often yield unsatisfactory results when faced with a high level of noise. In response, some researchers have shifted their focus towards using robust regularization methods, such as early robust learning [7,10], and Mixup [19], to alleviate this issue. However, designing a reliable method to distinguish noisy samples poses a significant challenge and can potentially lead to accumulated errors due to incorrect selections. In contrast to previous studies that primarily address classification tasks, our work focuses on robust segmentation learning. Notably, ADELE [20] dynamically adjusts noisy annotations by leveraging the phenomenon of early learning in semantic segmentation. Different from existing works on segmentation, we propose the full-domain-aware noise reduction model specifically designed for robust segmentation learning.

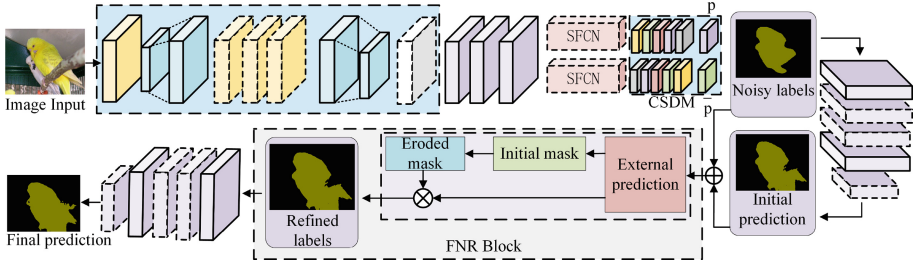


Fig. 2. The overall architecture of CD-CBN. These are separated by a Spatial Feature Calibration Network (SFCN), a Class-Specific Distance Model (CSDM), and a Full-domain-aware Noise Reduction (FNR).

3 Methodology

3.1 Overview

Figure 2 provides an overview of our proposed CD-CBN, comprising three key components: a SFCN, a CSDM, and a FNR block. First, an input image is fed into the proposed network, which has undergone pre-training on ImageNet [21] for extracting features. Then, the extracted features were divided into two parts, object and background areas, and sent into the SFCN, which is designed to align multi-dimensional features with class prototypes in order to capture inter-class consistent features. Subsequently, the CSDM is employed to integrate features from the SFCN outputs in a manner that enhances semantic complementarity. Then, we combine the initial noisy pseudo-labels generated by the classification network with the initial predictions obtained by the segmentation network to further complement the semantic information, which leads to an explicit distinction between object and background information. Finally, the features are fed into the FNR, the original noisy pseudo labels are refined by training to eliminate noise pixels and further enhance target features, resulting in improved pseudo labels that achieve better segmentation performance on complex images.

3.2 Spatial Feature Calibration Network (SFCN)

SFCN is proposed to aggregate the global context in Fig. 3. In many other WSSS methods, object locations are identified using CAMs, which often highlight the most distinctive features of objects.

Specifically, given the input X . the output for the c -th channel at a particular height h can be written as

$$z_c^h(h) = \frac{1}{W} \sum_{0 \leq i \leq W} x_c(h, i). \tag{1}$$

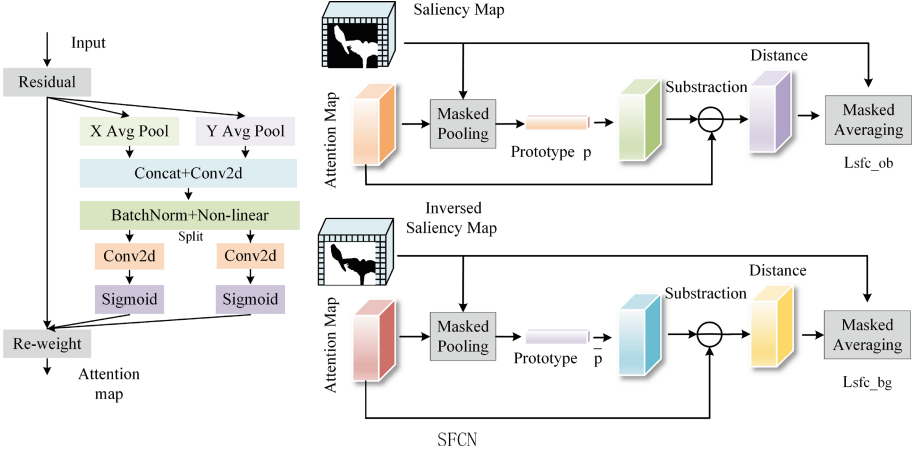


Fig. 3. Spatial Feature Calibration Network(SFCN).

Likewise, the output of the c -th channel at width w can be expressed as

$$z_c^w(w) = \frac{1}{H} \sum_{0 \leq j \leq H} x_c(j, w) \tag{2}$$

As described above, Eqs. (1) and (2) facilitate a comprehensive receptive field and encode detailed positional data. Specifically, they are concatenated and passed for transformation $F1$.

$$f = \sigma(F_1([z^h, z^w])) \tag{3}$$

Here, generated by concatenating horizontally and vertically aggregated features, is divided into two tensors.

$$g^h = \sigma(F_h(f^h)) \tag{4}$$

$$g^w = \sigma(F_w(f^w)) \tag{5}$$

Here, $\sigma(\cdot)$ is the sigmoid function. The outputs g^h and g^w focus on crucial spatial features while maintaining efficiency.

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \tag{6}$$

For training the classification network, we utilize the multi-label soft margin loss, defined as follows:

$$L_{cls} = -\frac{1}{C} \sum_{c=1}^C y^c \log(\sigma(q^c)) + (1 - y^c) \log(1 - \sigma(q^c)) \tag{7}$$

y_c represents the C -th class at the image level, to generate the CAM for each object category.

$$A^c = \frac{\text{ReLU}(F^c)}{\max(F^c)} \tag{8}$$

While CAM can identify objects within images, it often emphasizes distinctive areas. Additionally, without boundary information, the object activations revealed by CAMs can extend into the background. Aiming at this issue, we propose a Spatial Feature Calibration Network (SFCN), as illustrated in Fig. 3. For images containing a single object category, the class-independent saliency map serves as an approximate object mask, and extracts a class prototype vector from the features. Subsequently, the element in the prototype vector p can be computed as:

$$p^c = \frac{\sum_{i=1, j=1}^{h, w} S_{ij} \cdot F_{ij}^c}{\sum_{i=1, j=1}^{h, w} S_{ij}} \quad (9)$$

Here, h and w correspond to the dimensions of the saliency map, representing its height and width respectively. Our approach aims to enhance the learning process by encouraging the development of coherent feature encoding, specifically in the target area, thereby enhancing the overall activation of objects in CAMs. The prototype vector is up-sampled to match the spatial dimensions of the attention map, and the feature distance D is computed using element-wise subtraction. The corresponding loss can be expressed as:

$$L_{sfc.ob} = \frac{1}{\sum_{i=1, j=1}^{h', w'} S_{ij}} \sum_{i=1, j=1}^{h', w'} \left(S_{ij} \cdot \frac{1}{C} \sum_{c=1}^C (D_{ij}^c)^2 \right) \quad (10)$$

Here h' , w' , and C correspond to the height, width, and channel dimensions of the attention map, respectively. Our proposed spatial feature calibration model offers a dual benefit. Firstly, by minimizing variations within the features of the same category, the network is encouraged to expand the entire object region indicated by the saliency map. Secondly, matching features with the prototype moderates the high level activation in CAMs' most distinctive regions, compelling the network to activate non-distinctive areas for maintaining classification performance. However, only focusing on intra-class relations may result in a positional offset, causing the activation to shift towards background regions, this can diminish the localization capability of CAMs. To tackle this challenge, we also utilize our spatial feature calibration model with the inverse saliency map to ensure the compactness and consistency of background features. The calculation of the background prototype follows a similar approach to Eq. (11).

$$\bar{p}^c = \frac{\sum_{i=1, j=1}^{h, w} \bar{S}_{ij} \cdot F_{ij}^c}{\sum_{i=1, j=1}^{h, w} \bar{S}_{ij}} \quad (11)$$

Here, $\bar{S} = 1 - S$ represents the inverse saliency map. Computing the feature distance \bar{D} with its prototype in the background area, the spatial feature calibration loss for the background region can be defined in a similar manner as Eq. (12) as follows:

$$L_{sfc.bg} = \frac{1}{\sum_{i=1, j=1}^{h', w'} \bar{S}_{ij}} \sum_{i=1, j=1}^{h', w'} \left(\bar{S}_{ij} \cdot \frac{1}{C} \sum_{c=1}^C (\bar{D}_{ij}^c)^2 \right) \quad (12)$$

CSDM. By incorporating the intra-class relation constraint into the spatial feature calibration model mentioned above, the variance of features within or outside the salient region is effectively minimized. Nevertheless, the presence of features exhibiting greater intra-category consistency does not necessarily imply that the activation of the object region will surpass that of the background area. To relieve the dilemma, we adopt a Class-Specific Distance Model (CSDM) to create a clear separation between object and background features and promote higher activation in the object region compared to the background. As illustrated in Fig. 2, y is image-level label and the class-specific distance loss is defined as follows:

$$L_{csd} = y \cdot \bar{p} - y \cdot p \quad (13)$$

Here, p and \bar{p} refer to the object and background prototypes, respectively, as defined in Equations (9) and (11).

With our proposed SFCN and CSDM, we can represent the classification loss in the training as:

$$L = L_{cls} + \lambda_{ob}L_{sfc.ob} + \lambda_{bg}L_{sfc.bg} + \lambda_{csd}L_{csd} \quad (14)$$

Here, λ_{ob} , λ_{bg} and λ_{csd} are the spatial feature calibration losses for objects and background, and the CSDM loss, respectively.

3.3 Full-Domain-Aware Noise Reduction (FNR)

Due to their ability to provide detailed object boundaries, saliency maps derived from existing saliency detection models have been extensively utilized in WSSS tasks, which mitigate the problem of easily confused object boundaries. Leveraging the category information available through image-level labels, we employ class-agnostic saliency mapping to generate pseudo-labels that are specific to each class. Nevertheless, the identified salient regions may not align with the objects or may encompass irrelevant background areas. As such, labels containing image-level noise often exhibit a high noise rate, leading to the degradation of network training performance. Aiming at this issue, we propose a Full-domain-aware Noise Reduction (FNR) model to alleviate the mislabeled pixel problems. Based on the assumption that the class label in the salient region is correct with high probability, first, a comprehensive activation region is obtained by combining the noisy pseudo-label and the initial prediction to supplement the incomplete object region in the initial prediction. Subsequently, an initial mask is generated based on the activated feature, which is then subjected to a denoising process to reduce false negatives among pixels. This method yields refined pseudo-labels with distinct boundaries, enhancing the label accuracy for subsequent segmentation tasks, which leverage the inherent self-correction capability of the segmentation network to enhance the final predictions of the model.

The specific process of extracting the initial mask is: firstly, given an input image, assume that its gray image is $G_{i,j}$ and define a threshold value T_{mg} , according to which the gray scale image is transformed into a binary image; then, the region

of the binary image containing a pixel value of 1 is labeled as the foreground target, and the region with a pixel value of 0 is labeled as the background region.

$$M_{ij} = \begin{cases} 1, & \text{if } G_{ij} > T_{mg} \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

4 Experiments

4.1 Datasets and Evaluation

Datasets. To assess the effectiveness of our approach, we conducted a comprehensive set of experiments on two widely recognized and popularly used datasets, the PASCAL VOC 2012 dataset [22] and the MS COCO 2014 dataset [23].

PASCAL VOC 2012 consists of 21 classes, including 20 object categories and the background class. It comprises a total of 10,582 training images (which are extended by the SBD dataset [24]). Additionally, there are 1,449 validation images and 1,456 test images available for evaluation.

MS COCO 2014 consists of 81 classes of objects, including one background class, and it has 80K images for training and 40K images for validation.

Evaluation. Consistent with previous studies [25]–[37], we employed the standard mean Intersection over Union (mIoU) and Pixel Accuracy (PixAcc) as the evaluation metrics to assess the effectiveness of our model.

4.2 Implementation Details

Our classification network is built upon the VGG-16 model [11], which is pre-trained on the ImageNet dataset [21]. We initialize the learning rate to 10^{-3} and reduce it by a factor of 10 after the 5th and 10th epochs. The classification network is trained for 30 epochs using a batch size of 5. As for the segmentation network, we adopt the DeepLab-v2 framework following the approach in [11]. The segmentation network undergoes training for 10,000 iterations, utilizing a batch size of 10. These settings ensure that both the classification and segmentation networks are effectively trained to achieve optimal performance on the respective tasks.

4.3 Evaluation and Analysis

Ablation Studies: To demonstrate the effectiveness of our method, we performed a comprehensive set of experiments on the Pascal VOC 2012 dataset to evaluate the performance of the key components in our model. Specifically, we studied the impact of SFCN, CSDM, and FNR individually. The results of the ablation studies are presented in Table 1, showing the performance of each component.

It is evident from the results that by employing the SFCN, can attain better global context features and recognize precise locations. As shown in Fig. 4, the first row represents the input image, followed by the ground truth in the second row, the saliency map in the third row, and the final result visualization in the last row. The results demonstrate that the network can be effectively guided to focus on the comprehensive regions corresponding to class prototypes. Furthermore, the inclusion of CSDM in the network leads to improved performance, which furthers the separation of target and contextual features. Nonetheless, due to the coarse features extracted by both SFCN and CSDM, a FNR block is proposed to refine the noisy labels and enhance performance. By integrating the SFCN, CSDM, and FNR components within a unified network, our method achieves superior accuracy compared to other variants.

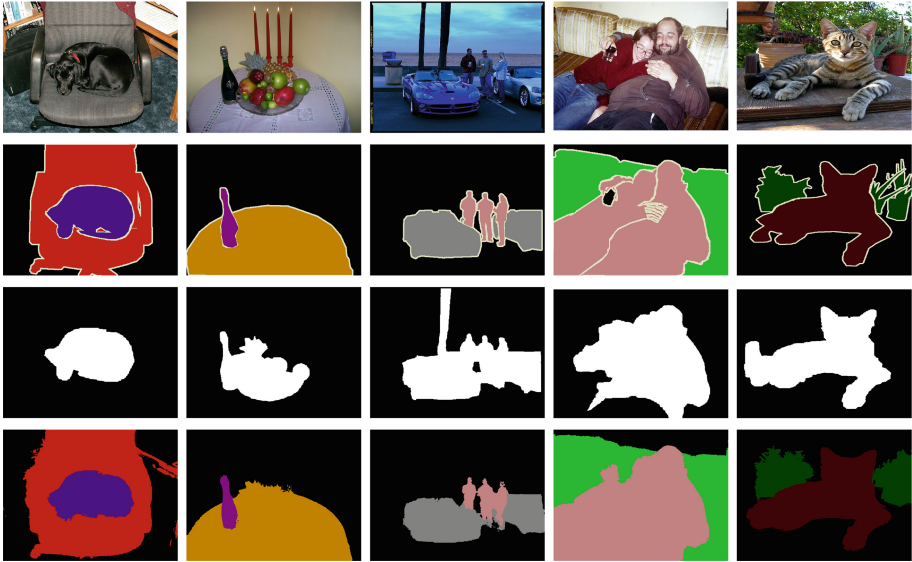


Fig. 4. The visualizations on Pascal VOC 2012 dataset.

Comparison with State-of-the-Art Networks: In Table 2 and Table 3, we conducted a comprehensive comparison of our results with state-of-the-art methods on the two widely-used datasets.

Quantitative Results on PASCAL VOC 2012 [21]: To demonstrate the effectiveness of the proposed CD-CBN, we compare it against the existing state-of-the-art WSSS approaches on the validation and test datasets of PASCAL

Table 1. Ablative studies on Pascal VOC 2012. SFCN: Spatial Feature Calibration network; CSDM: class-specific distance model; FNR:full-domain-aware noise reduction.

Baseline	SFCN	CSDM	FNR	MIoU (%)	PixAcc (%)
✓	-	-	-	64.8	91.4
✓	✓	-	-	65.0	91.1
✓	✓	✓	-	69.3	92.6
✓	✓	✓	✓	72.2	92.9

Table 2. Comparisons with the state-of-the-art approaches on Pascal VOC 2012 dataset. I: image-level labels, S: saliency maps.

Methods	Publication	Sup.	Val(%)	Test(%)
DSRG [25]	CVPR18	I+S	61.4	63.2
Affinity Net [5]	CVPR18	I	61.7	63.7
FickleNet [26]	CVPR19	I+S	64.9	65.3
OAA [27]	ICCV19	I+S	65.2	66.4
SEAM [28]	CVPR20	I+S	64.5	65.7
CONTA [29]	NIPS20	I	66.1	66.7
Li et al. [30]	AAAI21	I+S	68.2	68.5
NSROM [31]	CVPR21	I+S	68.3	68.5
ECS-Net [32]	ICCV21	I	66.6	67.6
I2CRC [11]	IEEE22	I+S	69.3	69.5
W-OoD [33]	CVPR22	I	70.7	70.1
L2G [34]	CVPR22	I+S	72.1	71.7
ACR [35]	CVPR23	I	71.9	71.9
MDBA [36]	IEEE23	I+S	72.0	71.5
CD-CBN(ours)	-	I+S	72.2	74.3

VOC 2012. As shown in Table 2, Specifically, Compared to some multi-stage semantic segmentation methods with only image-level class labels, DSRG [25], Affinity Net [5], FickleNet [26], OAA [27], SEAM [28], CONTA [29], Li et al. [30], NSROM [31], ECS-Net [32], I2CRC [11], W-OoD [34], L2G [35], ACR [36], MDBA [37]. Compared to these methods, our method can achieve 72.2% on the validation dataset and 74.3% on the test dataset, which can demonstrate our method’s superior ability to accurately locate and expand the complete object region, thereby enhancing segmentation performance.

Quantitative Results on MS COCO 2014 [22]: In this section, we present the experimental results on the highly challenging MS COCO 2014 dataset [23]. Table 3 illustrates the performance of our approach and other weakly supervised methods [25,30], [40], and [33] on the validation set. Our CD-CBN model surpasses the existing WSSS models, achieving a mIoU of 31.9%. These results

Table 3. Comparisons with the state-of-the-art approaches on MS COCO 2014 validation set. sup: supervision, I: image-level label, S: saliency maps.

Methods	Publication	Sup.	Backbone	Val(%)
DSRG [25]	CVPR18	I+S	VGG16	26.0
IAL [37]	IJCV20	I+S	VGG16	27.7
GWSM [30]	AAAI21	I+S	VGG16	28.4
I2CRC [11]	TMM22	I+S	VGG16	31.2
CD-CBN(ours)	-	I+S	VGG16	31.9

present the capability of our proposed method to enhance segmentation quality by mitigating background incompleteness and addressing object incompleteness.

5 Conclusion

In this work, we propose an innovative CD-CBN method for WSSS, CD-CBN is composed of three key components: SFCN, CSDM and FNR block. SFCN is used to align class prototypes with features in spatial features with cross-dimensional feature fusion. CSDM promotes the network to produce stronger activation for the object prototype compared to the background. FNR is proposed to filter out pixel-level noise and refine semantic segmentation. We conducted extensive comparisons with other state-of-the-art approaches on two widely-used datasets, the results demonstrate the effectiveness and robustness of our method.

References

1. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)
2. Lin, D., Dai, J., Jia, J., He, K., and Sun, J.: ScribbleSup: scribble-supervised convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016)
3. Dai, J., He, K., and Sun, J.: BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: International Conference on Computer Vision (ICCV) (2015)
4. Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L.: What’s the point: semantic segmentation with point supervision. In: European Conference on Computer Vision (ECCV) (2016)
5. Ahn, J. and Kwak, S.: Learning pixel-level semantic affinity with image level supervision for weakly supervised semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
6. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

7. Chen, X., Gupta, A.: Webly supervised learning of convolutional networks. In: ICCV (2015)
8. Xia, X., et al.: Robust early-learning: hindering the memorization of noisy labels. In: ICLR (2020)
9. Han, B., et al.: CoTeaching: robust training of deep neural networks with extremely noisy labels. In: NeurIPS (2018)
10. Jiang, L., Zhou, Z., Leung, T., Li, L.-J., Fei-Fei, L.: MentorNet: learning data-driven curriculum for very deep neural networks on corrupted labels. In: ICML (2018)
11. Chen, T., Yao, Y., Zhang, L., Wang, Q., Xie, G., Shen, F.: saliency guided inter- and intra-class relation constraints for weakly supervised semantic segmentation. *IEEE Transactions on Multimedia (TMM)*, vol. 25, pp. 1727–1737 (2022)
12. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015)
13. Badrinarayanan, V., Kendall, A., and Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for image segmentation, vol. 39, no. 12, pp. 2481–2495 (2017)
14. Lin, G., Milan, A., Shen, C., Reid, I.: RefineNet: multi-path refinement networks for high-resolution semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1925–1934 (2017)
15. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803 (2018)
16. Fu, J., et al.: Dual attention network for scene segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3146–3154 (2019)
17. Huang, Z., et al.: CCNet: criss-cross attention for semantic segmentation. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 603–612 (2019)
18. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.-G.: Learning from noisy labels with deep neural networks: A survey. *IEEE Trans. Neural Netw. Learn. Syst. (TNNLS)* **2**, 3 (2022)
19. Zhang, H., Cisse, M., Dauphin, Y. N., Lopez-Paz, D.: mixup: beyond empirical risk minimization. In: ICLR (2018)
20. Liu, S., Liu, K., Zhu, W., Shen, Y., Fernandez-Granda, C.: Adaptive early-learning correction for segmentation from noisy annotations. In: CVPR (2022)
21. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009)
22. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision* **88**(2), 303–338 (2010)
23. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: *Proceedings of the European Conference on Computer Vision*, pp. 740–755 (2014)
24. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV, pp. 991–998, IEEE (2011)
25. Huang, Z., Wang, X., Wang, J., Liu, W., Wang, J.: Weakly-supervised semantic segmentation network with deep seeded region growing. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7014–7023 (2018)

26. Lee, J., Kim, E., Lee, S., Lee, J., Yoon, S.: FickleNet: weakly and semi supervised semantic image segmentation using stochastic inference. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp. 5267–5276 (2019)
27. Jiang, P.-T., Hou, Q.-B., Cao, Y., Cheng, M.-M., Wei, Y., Xiong, H.: Integral object mining via online attention accumulation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2070–2079 (2019)
28. Wang, Y., Zhang, J., Kan, M., Shan, S., Chen, X.: Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12275–12284 (2020)
29. Zhang, D., Zhang, H., Tang, J., Hua, X.-S., Sun, Q.: Causal intervention for weakly-supervised semantic segmentation. In: Advances in Neural Information Processing Systems, vol. 33 (2020)
30. Li, X., Zhou, T., Li, J., Zhou, Y., Zhang, Z.: Group-wise semantic mining for weakly supervised semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1984–1992 (2021)
31. Yao, Y., et al.: Non-salient region object mining for weakly supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2623–2632 (2021)
32. Sun, K., Shi, H., Zhang, Z., Huang, Y.: ECS-Net: improving weakly supervised semantic segmentation by using connections between class activation maps. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7283–7292 (2021)
33. Lee, J., Oh, S. J., Yun, S., Choe, J., Kim, E., Yoon, S.: Weakly supervised semantic segmentation using out-of-distribution data. In: CVPR, pp. 16897–16906 (2022)
34. Jiang, P.-T., Yang, Y., Hou, Q., Wei, Y.: L2G: A Simple Local-to-Global Knowledge Transfer Framework for Weakly Supervised Semantic Segmentation. In: CVPR, pp. 16886–16896 (2022)
35. Kweon, H., Yoon, S.-H., Yoon, K.-J.: Weakly supervised semantic segmentation via adversarial learning of classifier and reconstructor. In: CVPR, pp. 11329–11339 (2023)
36. Chen, T., Yao, Y., Tang, J.: Multi-granularity denoising and bidirectional alignment for weakly supervised semantic segmentation. *IEEE Transactions on Image Processing (TIP)*, vol. 32, pp. 2960–2971 (2023)
37. Wang, X., Liu, S., Ma, H., Yang, M.-H.: Weakly-supervised semantic segmentation by iterative affinity learning. *Int. J. Comput. Vision (IJCV)* **128**(6), 1736–1749 (2020)



EFLLD-NET: Enhancing Few-Shot Learning with Local Descriptors

Guangtong Lu¹, Weidong Du², and Fanzhang Li¹(✉) 

¹ School of Computer Science and Technology, Soochow University, Suzhou, China
gtlu@stu.suda.edu.cn, lfzh@suda.edu.cn

² Focusight Technology, Jiangsu, China
davy@focusight.net

Abstract. Few-shot image classification aims to learn a model to correctly classify images with a few labeled data, however, feature extractors often fail to extract more generalized and discriminative features in low-data scenarios. Previous work has shown promising improvements by utilizing local descriptors of images. Unfortunately, they do not effectively suppress local descriptors where background noise is located and local descriptors that favour the classification of other classes. In this paper, we propose exploiting the relationship between the support set itself and the relationship between the support set and the query set in a task to enhance the discriminative power of local descriptors while suppressing the local descriptors associated with background noise in the images. In addition, we explore how to combine global features of images with local descriptors organically. Extensive experiments demonstrate that our method outperforms the state of the art on both standard datasets for few-shot learning.

Keywords: Deep Learning · Few-Shot Learning · Meta Learning · Local Descriptors

1 Introduction

With the advancement of deep learning, it has achieved remarkable success in the field of computer vision [5, 6, 16]. Despite its numerous advantages, deep learning also exhibits limitations and challenges. One of the primary challenges is the requirement for a substantial amount of annotated data for training, which can be challenging to obtain and financially costly. Few-shot learning aims to address the image classification problem in scenarios with limited data, thus mitigating the challenge of data scarcity faced by deep learning to some extent.

Few-shot Learning (FSL) [3, 12, 17, 18, 20] aims to imitate humans learning new concepts through limited examples. To achieve human-like capabilities, few-shot learners first learn deep knowledge from a base training dataset and generalize this knowledge through a few examples to identify new unseen classes.

However, the data distribution formed by a small number of samples often

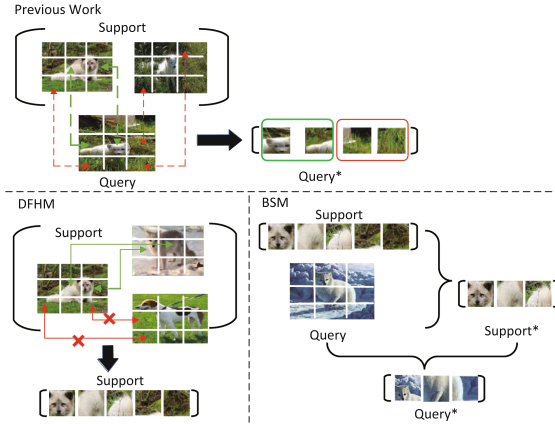


Fig. 1. Previous work did not take into account that images in a task would have the similarity background, so the background descriptors would be used as discriminative feature. At the same time, through the DFHM and BSM modules, we can allow discriminative features to play a greater role while suppressing the background descriptors. In DFHM, where the red bi-directional solid arrows indicate that the DFHM suppresses the background noise in the current category of images with the help of other categories of images in the support set. Meanwhile, the green bi-directional solid arrows represent that the DFHM uses the images of the same category in the current support set to filter out the more discriminative features. In BSM, we first use the query set to further filter the more discriminative features Support* in the support set which after DFHM processing, and then we use Support* to select the more discriminative Query* in the query set. This is what we call two-way selection. (Color figure online)

deviates significantly from the true data distribution. Training models on imbalanced data distributions can lead to severe overfitting, greatly undermining the model’s generalization ability. Therefore, it becomes crucial to identify more discriminative features in images, specifically the characteristics of the foreground object, while suppressing adverse features such as cluttered background noise that hampers image classification.

In order to utilize local descriptors to achieve the goal of enhancing the foreground of an image while suppressing cluttered background noise, previous work such as DN4 [10] lets the similarity calculation between each descriptor in a query image and the nearest neighbour descriptor in each support class. Finally, the similarity of the query descriptors is accumulated as the similarity of the query set image to the current class, [11] introduce the Discriminative Mutual Nearest Neighbor Neural Network (DMN4) which leverages the relationships between query sets and support sets to alleviate the cumulative impact of aggregating background clutter; specifically, they use the support set that emphasizes the more discriminative local descriptors in the query set by the mutual nearest neighbor selection. However, in our perspective, although DMN4 [11] addresses the fact that DN4 [10] directly uses all local descriptors for classification without considering the effect of background noise local descriptors on the experiments, we believe that the use of a pool of local descriptors consisting of local descriptors from all classes in the support set for filtering the query set is also affected

by background noise. As shown in Fig. 1, when the backgrounds of images in a task are strikingly similar, the local descriptors representing the background noise in the query image will be considered as discriminative features, which in turn will affect the classification results. In addition, both ignore the global features of the image.

Therefore, in our work, we propose a Discriminative Feature Highlighting Module to mitigate the effects of background local descriptors in the support set with the help of relationships between different classes within the support set. In addition, the previous work only used the support set to filter the query set without considering to understand the relationship between support set and query set. So, we propose a Bi-directional Selection Module for bi-directional selection between query set and support set to obtain more generalized feature by the relationship of them. Preferably, we propose a Feature Fusion Module to effectively explore how to combine an image’s global features with local descriptors to fully utilize the information of a image.

In summary, our main contributions can be summarized as follows:

- A Feature Fusion Module (FFM) is proposed to fuse global features with local descriptors to obtain more discriminative local descriptors.
- A module called Discriminative Feature Highlighting (DFH) is proposed for emphasizing discriminative features and suppressing background features within a support set.
- A Bidirectional Selection Module (BSM) is proposed to emphasize the discriminative features in the support set and query set.
- We experimentally demonstrate that our method reaches competitive accuracy on popular benchmark datasets.

2 Related Work

2.1 Few-Shot Learning

Few-shot learning methods can be classified as metric-based [12, 17, 18, 20] and optimization-based [3, 7, 14, 23] methods depending on how the model works. Metric-based methods execute classification by performing a similarity calculation between the feature representations of the acquired images. Optimization-based meta-learning involves training a meta-learner to acquire optimized parameters that enable efficient adaptation to new tasks with limited data. It aims to improve the learning process itself by fine-tuning the model parameters through iterative updates, facilitating rapid generalization to novel tasks.

3 Methodology

3.1 Problem Definition

The Few-shot classification is specified as an N-way K-shot classification, in which we should classify the N classes and each of them has K-labeled data. Two

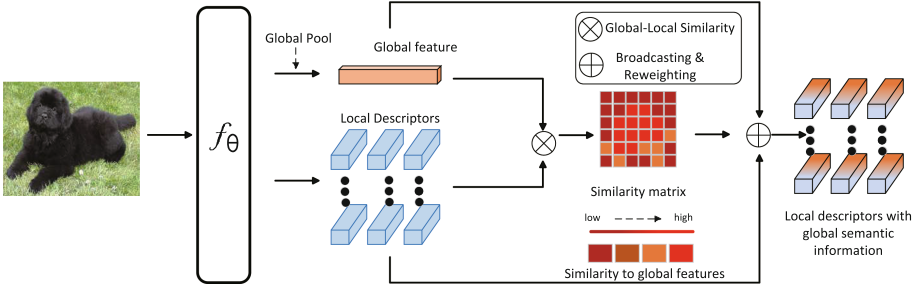


Fig. 2. Feature fusion module. Instead of indiscriminately adding the same global semantic information to all local descriptors, we set the weights based on the cosine similarity between the local descriptors and the global features.

databases are given to us, which are named base dataset D_b and novel dataset D_n respectively, $D_b = \{(x_b^i, y_b^i)\}_{i=0}^{n_b}$, where $y_b^i \in C_b$ and the $D_n = \{(x_n^i, y_n^i)\}_{i=0}^{n_n}$, where $y_n^i \in C_n$. It is important to emphasize that these two data sets are disjoint which means $C_b \cap C_n = \emptyset$. The base dataset has sufficient labeled training data, whereas the novel dataset has only a small number of labeled training data. FSL does not sample images as traditional image classification does, but samples the task to train. A task consists of the support set $S = \{(x_n^i, y_n^i)\}_{i=0}^{N \times K}$ and the query set $Q = \{(x_n^i, y_n^i)\}_{i=N \times K}^{N \times K + N \times Q}$, where Q is the number of images in each of the N categories in the query set.

3.2 Feature Fusion Module

We will first explore how to effectively combine an image’s global and local features to obtain more discriminative local descriptors, besides, it also can avoid DFHM module misclassifying discriminative descriptors as background noise due to the same coat colour but different categories (e.g. white dog and white cat, under the premise of only looking at the torsos of white cats and white dogs, they are highly similar). It can be effectively improved by the local descriptors with global semantic information, because at this time the local descriptors where the torsos of the white dog and the white cat are located already have the category information of the dog and the cat, respectively, so their similarity will be reduced a lot also to a certain extent to avoid the misclassification of DFHM. The procedure is shown in Fig. 2, for an image x , we feed it into the feature extraction network $f(\cdot)$ to obtain its feature map $f(x) \in R^{hw \times d}$, which represents that we have divided a picture x into $r = hw$ local descriptors, each of dimension d , and next get the global feature $g(x) \in R^d$ of the image after a global pooling operation on $f(x)$, and then we reshape it to $g'(x) \in R^{r \times d}$ so that it can work on every local descriptor. Next, we need to think about how they fit together organically. If we simply apply the global features of the image equally to each local descriptor may hinder the functioning of our subsequent modules, because the local descriptor where the background is located will also have some

of the feature information of the foreground of the image. Therefore, we need to consider how only the local descriptor where the foreground object is located can benefit from the global features. We can note that the global features of an image tend to express the foreground of the image as the learning process progresses, so we can weigh the global features of the image before working on the local descriptors. Therefore, the local descriptor with global semantic information can be defined as

$$f'(x) = f(x) + \mathcal{M}(f(x), g(x))g'(x), \quad (1)$$

$$\text{where, } \mathcal{M}(f(x), g(x)) = f(x)g(x) \in R^r$$

$\mathcal{M}(\cdot, \cdot)$ is a matrix multiplication operation that calculates the similarity of each local descriptor to the global feature. In this way, local descriptors with greater similarity to global features can benefit more from global features, and conversely will benefit less from global features.

3.3 Discriminative Feature Highlighting Module

After obtaining FFM-processed local descriptors with global semantic information we can further explore how to filter out the more discriminative local descriptors, as well known, the image local descriptors that have high similarity with different categories will essentially belong to the background noise features of the image. Therefore, we let all local descriptors of a class in the support set calculate the similarity with the local descriptors of other classes in the support set. Suppose that a local descriptor exists with greater similarity to the other classes. In that case, we will primarily consider it to be the background and, therefore, we give it a lower weight. As indicated by the solid red bidirectional arrows in the DFHM which in the Fig. 1, two different classes of foregrounds yet have the same grass background. Therefore, most local descriptor where the grass is located are filtered out by our DFHM. By the way, it is worth mentioning that in the 5-way 5-shot scenario, our DFHM module can also be used to enhance discriminative features with the help of images of the same category. This is because images belonging to the same category must have the same foreground, so if a local descriptor of one image has a high degree of similarity with the local descriptor of another image of the same category, then it is likely be the local descriptor of the foreground of the image, it is show as the green solid bidirectional arrows in the Fig. 1.

We use the bold font $\mathbf{d}_c = \{d_c^{kr} | k = 1, \dots, K, r = 1, \dots, hw\}$ represents local descriptors from the same class $c \in C$ in the support set, where d_c^{kr} represents a local descriptor of an image which label is c . Thus the formula that emphasizes the discriminative features of each class within the support set is shown below:

$$\text{weight}_c = \frac{1}{(N-1)Kr} \sum_{i=1}^{N/c} \text{Similarity}(\mathbf{d}_c, \mathbf{d}_i), \quad (2)$$

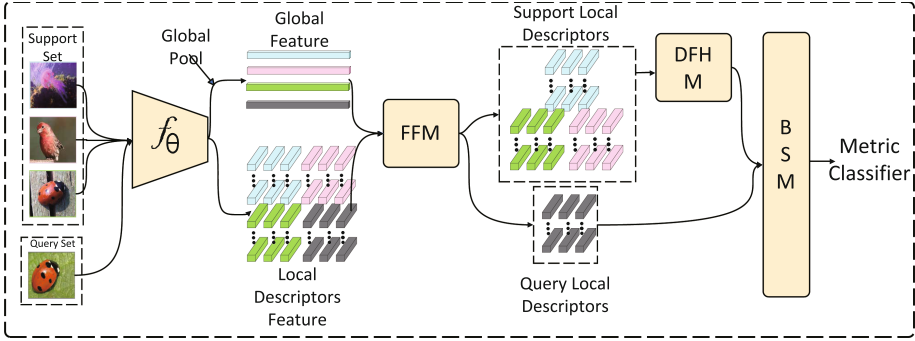


Fig. 3. The illustration of the whole network architecture of our work and our Feature Fusion Module. We first obtain local descriptors with global semantic information through the FFM, then use the DFHM module to obtain more discriminative local descriptors for the support set, and finally obtain discriminative local descriptors for the support and query sets through the BSM.

$$d_c = d_c \odot \frac{1}{weight_c}, \tag{3}$$

where \odot represents Element-wise Product, meanwhile, the $Similarity(d_i, d_j)$ represents the cosine similarity of local descriptors between class i and class j within the support set, so the higher its value means the higher the probability that the current local descriptor is where the background is located, so we have to give it a lower weight, and in turn, we have to give him a higher weight.

3.4 Bidirectional Selection Module

Our DFHM can theoretically perform well if the environments in which the different classes of objects in the support set exist are highly similar, but if the environments in which the different classes of objects exist are also different our DFHM may not perform as well as expected. For a query set in a task, we do not know what each image class is, but we know that there will be Q samples for each class, and the classes correspond to the support set. Therefore we need to make use of the query set as well, and not just as an object to be classified. With this information, we can use the query set to highlight discriminative local descriptors in the support set before filtering the query set by the support set and finally feed back into the filtering of the query set, which we call bidirectional selection. As shown in Fig. 1, the local descriptors filtered out by the DFHM may still have background noise. Therefore, in our BSM, we first use the query set to further filter the local descriptors in the support set to get a more discriminative local descriptor Support*, and then use Support* to filter the local descriptors Query* in the query set that are discriminative. Specifically, we use bold font $\mathbf{q} = \{q^l | l = 1, \dots, NQr\}$ to represent the local descriptor pool consisting of all local descriptors of the query set, while q^l represents a local descriptor of an

image in the query set. We define ϕ_c^{kr} , which is computed with the help of the query set, as the relevance of a local descriptor d_c^{kr} in \mathbf{d}_c to class c .

$$\phi_c^{kr} = \frac{1}{n} \sum_{i=1}^n \text{Topn Similarity}(d_c^{kr}, q^i). \tag{4}$$

Therefore, the support set after query set selection is defined as follows:

$$\mathbf{d}_c = \Phi(\mathbf{d}_c, \mathbf{q}) \odot \mathbf{d}_c, \tag{5}$$

where, $\Phi(\mathbf{d}_c, \mathbf{q}) = \{\phi_c^{kr} | k = 1, \dots, K, r = 1, \dots, hw\}$, Topn represents we choose n local descriptors in the \mathbf{q} to calculate the correlation between a local descriptor in \mathbf{d}_c and the foreground object of class c . Finally, we will use the support set after the query set filtering to feed the process of filtering the discriminative query set local descriptors, we use $q_i = \{q_i^j | j = 1, \dots, r\} (1 \leq i \leq Q)$ to represent an image in the query set, where q_i^j represents a local descriptor of the image. Then we define $\psi_i^{j,c}$ as the similarity of each local descriptor in q_i to the category c in the support set as follows:

$$\psi_i^{j,c} = \frac{1}{r} \sum_{l=1}^r \text{Similarity}(q_i^j, d_c^l), \tag{6}$$

thus, q_i becomes $q_{i,c}$ after being filtered by the local descriptors of all the support sets belonging to class c :

$$q_{i,c} = q_i \odot \Psi(q_i, \mathbf{d}_c). \tag{7}$$

That is, q_i will be dynamically adjusted to $q_{i,c}$ to suit the different categories in the support set, instead of using q_i for all classes as in previous work, where, $\Psi(q_i, \mathbf{d}_c) = \{\psi_i^{j,c} | j = 1, \dots, r\}$, so the similarity between q_i and class n can be defined as follows:

$$s_{q_i,n} = \sum_{l_q=1}^{|q_{i,n}|} \max_{l_n=1}^{|d_n|} \text{Similarity}(q_{i,n}^{l_q}, d_n^{l_n}). \tag{8}$$

Subsequently, the loss function can be defined as follows:

$$\begin{aligned} \mathcal{J}(\phi) &= -\frac{1}{|Q|} \sum_{q_i \in Q} y \log p_\phi(\hat{y} = y | q_i) \\ p_\phi(\hat{y} = c | q_i) &= \frac{\exp(s_{q_i,c})}{\sum_{c'=1}^N \exp(s_{q_i,c'})} \end{aligned} \tag{9}$$

where y is the label of the query image q_i . The overall architecture is illustrated in Fig. 3.

4 Experiments

Table 1. 5-way 1-shot and 5-shot classification accuracies on Mini-ImageNet and Tiered-ImageNet datasets using ResNet-12 backbones with 95% confidence intervals. All the results of comparative methods are from the exiting literature ('-' not reported).

Method	Backbone	Mini-ImageNet		Tiered-ImageNet	
		1-shot	5-shot	1-shot	5-shot
MatchingNet [20]	ResNet-12	75.99 ± 0.15	68.50 ± 0.92	80.60 ± 0.71	
ProtoNet [17]	ResNet-12	62.33 ± 0.12	80.88 ± 0.41	68.40 ± 0.14	84.06 ± 0.26
RelationNet [18]	ResNet-12	60.97	75.12	64.71	78.41
MetaOptNet [7]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
DN4 [10]	ResNet-12	65.35	81.10	69.60	83.41
DeepEMD [24]	ResNet-12	65.91 ± 0.82	82.41 ± 0.56	71.16 ± 0.87	86.03 ± 0.58
RFS-Simple [19]	ResNet-12	62.02 ± 0.63	79.64 ± 0.44	69.74 ± 0.72	84.41 ± 0.55
RFS-Distill [19]	ResNet-12	64.82 ± 0.60	82.14 ± 0.43	71.52 ± 0.69	86.03 ± 0.49
BML [25]	ResNet-12	67.0 ± 0.63	83.6 ± 0.29	68.9 ± 0.50	85.4 ± 0.34
FRN [22]	ResNet-12	66.45 ± 0.19	82.83 ± 0.13	72.06 ± 0.22	86.89 ± 0.14
DMN4 [11]	ResNet-12	66.58	83.52	72.10	85.72
DSKT-2 [4]	ResNet-12	67.33 ± 0.82	84.19 ± 0.50	72.11 ± 0.89	86.69 ± 0.59
SPRM [9]	ResNet-12	66.35 ± 0.34	82.24 ± 0.27	70.70 ± 0.33	85.40 ± 0.25
EFLLD-Net(ours)	ResNet-12	67.98 ± 0.52	84.39 ± 0.33	73.63 ± 0.13	86.78 ± 0.53

4.1 Datasets and Implementation Details

We experimented on three data sets that are more mainstream in few-shot learning.

Mini-Imagenet [20] contains 100 different classes with 600 images in each class. In this case, 64 classes are used as the training set, 16 classes are used as the validation set and the remaining 20 classes are used as the test set.

Tiered-Imagenet [13] consists of 608 classes with 779,165 images. Out of these, 351 categories were used for the training set, 97 categories were used for the validation set, and 8 categories were used for the test set.

CUB-200 [21] is a fine-grained dataset for bird, which contains 200 different bird species. In the dataset, we used 100 classes as the training set, 50 classes are used as the validation set and the remaining 50 classes are used as the test set.

Network Architecture. We use Conv-4 and ResNet-12 [5] as feature extraction networks f_θ . In our experiments, the images in each dataset were

Table 2. 5-way 1-shot and 5-shot classification accuracies on Mini-ImageNet and Tiered-ImageNet datasets using Conv-4 backbones with 95% confidence intervals. All the results of comparative methods are from the exiting literature ('-' not reported).

Method	Backbone	Mini-ImageNet		Tiered-ImageNet	
		1-shot	5-shot	1-shot	5-shot
MatchingNet [20]	Conv-4	43.56 ± 0.84	55.31 ± 0.73	-	-
ProtoNet [17]	Conv-4	51.20 ± 0.26	68.94 ± 0.78	53.45 ± 0.15	72.32 ± 0.57
RelationNet [18]	Conv-4	50.44 ± 0.82	65.32 ± 0.70	54.48 ± 0.93	71.31 ± 0.78
MetaOptNet [7]	Conv-4	52.87 ± 0.57	68.76 ± 0.48	54.71 ± 0.67	71.79 ± 0.59
CovaMNet [10]	Conv-4	51.19 ± 0.76	67.65 ± 0.63	54.98 ± 0.90	71.51 ± 0.75
DN4 [10]	Conv-4	51.24 ± 0.74	71.02 ± 0.64	52.89 ± 0.23	73.36 ± 0.73
DeepEMD [24]	Conv-4	51.72 ± 0.20	65.10 ± 0.39	51.22 ± 0.14	65.81 ± 0.68
RFS-Simple [19]	Conv-4	55.25 ± 0.58	71.56 ± 0.52	56.18 ± 0.70	72.99 ± 0.55
RFS-Distill [19]	Conv-4	55.88 ± 0.59	71.65 ± 0.51	56.76 ± 0.68	73.21 ± 0.54
ATL-Net [2]	Conv-4	54.30 ± 0.76	73.22 ± 0.63	-	-
FRN [22]	Conv-4	54.87	71.56	55.54	74.68
DMN4 [11]	Conv-4	55.77	74.22	56.99	74.13
EFLLD-Net(ours)	Conv-4	56.69 ± 0.74	75.36 ± 0.31	57.66 ± 0.72	75.87 ± 0.43

scaled to 84×84 , so for the feature extractor with ResNet-12 as the skeleton, a feature map whose shape is $19 \times 19 \times 64$ of the image will be obtained, Conv-4 will get a feature map of size $5 \times 5 \times 512$.

Training and Evaluation. For both Conv-4 and ResNet-12, we performed pre-training followed by fine-tuning in the meta-training phase. For Conv-4, we fine-tuned it for 100 epochs using the Adam optimizer with a learning rate of $1e-3$, decaying it by a factor of 0.1 every 30 epochs. For ResNet-12, we also fine-tuned it for 100 epochs using the Adam optimizer with a learning rate of $1e-4$, decaying it by a factor of 0.2 every 30 epochs. The topn belonging to the BSM in miniImageNet, tieredImageNet and CUB-200 are set to 15,15,20, respectively.

4.2 Comparisons with State-of-the-Art Methods

Results on Mini-ImageNet. Table 1 and Table 2 show that EFLLD-NET achieves state-of-the-art. Compared with the local descriptor based on Conv-4, we have 1.14% and 0.92% improvement on 5-way 5-shot and 1-shot, respectively, at the same time, we have 2.14% and 0.81% improvement on 5-way 5-shot and 1-shot respectively, compared with other non-local descriptor-based methods. On ResNet-12, we have 0.2% and 0.65% improvements on 5-way 5-shot and 1-shot, respectively, compared to the local descriptor-based, while we have 0.79% and 0.98% improvements, respectively, compared to the non-local descriptor-based methods.

Table 3. 5-way 1-shot and 5-shot classification accuracies on CUB-200 dataset using ResNet-12 backbones with 25% confidence intervals.

Method	Conv-4		ResNet-12	
	1-shot	5-shot	1-shot	5-shot
ProtoNet [17]	63.73	81.50	66.09	82.50
DN4 [10]	73.42	90.38	-	-
DSN [15]	66.01	85.41	80.80	91.19
CTX [1]	69.64	87.31	78.47	90.90
DeepEMD [24]	-	-	77.14	88.98
FRN [22]	73.48	88.43	83.16	92.59
DMN4 [11]	78.36	92.16	-	-
IAM [8]	77.17	89.90	84.17	93.30
DSKT-2 [4]	-	-	78.32	91.47
EFLD-Net(ours)	81.37	93.42	86.74	93.87

Results on Tiered-ImageNet. The experiments results on Tiered-ImageNet are also shown in Table 1 and Table 2. For local descriptor-based methods with Conv-4 as backbone, we have 0.67% and 1.74% improvement on 5-way 1-shot and 5-shot, respectively, while at the same time we have 0.86% and 1.19% improvement compared with other non-local descriptor-based method, respectively. Meanwhile, the result on ResNet-12, compared with local descriptor-based methods, we have 1.53% and 1.06% improvements on 5-way 1-shot and 5-shot, respectively. Compared to other non-local descriptor based sota our results have a 1.57% improvement on the 5-way 1-shot, but at 5-shot we are 0.11% lower compared to sota.

Results on CUB-200 Dataset. The experiments results on CUB-200 are shown in Table 3, it can be clearly seen that our proposed method works effectively on fine-grained datasets like CUB-200. Compared with the state-of-the-art approach using Conv-4 as the backbone network, we achieve an improvement of 3.01% and 1.26% on the 5-way 1-shot and 5-shot, respectively. Moreover, compared to the state-of-the-art approach using ResNet-12 as the backbone network, we achieve 2.57% and 0.57% improvements in 5-way 1-shot and 5-shot, respectively.

5 Ablation Study

5.1 Effective of Submodules

In this section, we will verify the validity of each submodule of our network by performing 5-way 1-shot and 5-shot ablation experiments on Mini-ImageNet and CUB-200 with ResNet-12 as our backbone. The results are shown in Table 4. It is

Table 4. Ablation study on Mini-ImageNet and CUB-200 datasets with ResNet-12.

EFLLD-NET			Mini-ImageNet		CUB-200	
FFM	DFHM	BSM	1-shot	5-shot	1-shot	5-shot
✗	✗	✗	64.35	80.67	73.24	89.47
✓	✗	✗	65.54	81.64	82.74	91.27
✓	✓	✗	66.67	82.71	84.67	92.83
✓	✗	✓	66.58	82.63	85.26	92.78
✗	✓	✓	67.21	82.95	85.43	92.87
✓	✓	✓	67.98	84.38	86.74	93.87

clear that when we use only FFM, the results obtained on Mini-ImageNet have an improvement of 1.19% and 0.97% on 5-way 1-shot and 5-shot, respectively, compared to the blank group. At the same time, there is an improvement of 9.47% and 1.8% in CUB-200 on 1-shot and 5-shot, respectively. This can strongly indicate the discriminability of the local descriptors with global semantic information that we obtain through FFM. The results obtained when we combined FFM and DFHM(or BSM) were significantly improved compared to the blank group and the group using only FFM, especially 12.02% and 2.52% on the cub-200, respectively. Compared to the blank group, at the same time, there was also a 1.13% and 1.07% improvement in 1-shot and 5-shot on Mini-ImageNet compared to the FFM-only group, and 2.52% and 1.51% on the CUB-200, respectively. This, on the one hand, can illustrate that our DFHM can effectively highlight the role of foreground objects in the picture through the relationship between different classes within the support set and the effectiveness of the BSM module. On the other hand, it should also prove the effectiveness of the combination of the DFHM (or BSM) and the FFM. Finally, when all three are combined, the best results are obtained in Mini-ImageNet and CUB-200. This is a strong indication of the effectiveness of our EFLLD-NET.

Table 5. The experiments results on Mini-ImageNet and CUB-200 dataset with different TopN using ResNet-12.

TopN	Mini-ImageNet		CUB-200	
	1-shot	5-shot	1-shot	5-shot
5	65.32	81.26	82.65	91.23
10	66.47	82.71	83.14	92.77
15	67.98	84.38	84.16	92.59
20	67.52	83.63	86.74	93.87
25	66.33	82.59	85.97	92.38

5.2 Choice of TopN in BSM

As mentioned above, for a local descriptor in the support set, Q local descriptors in \mathbf{q} will be highly similar. So, theoretically, we should use Q local descriptors in \mathbf{q} to select each sample in the support set, but this is not rigorous, so we conducted experiments on the Mini-ImageNet and CUB-200 datasets with different `topn` to illustrate the reasonableness of the `topn` we chose. The results are shown in Table 5. It can be seen that our EFLLD-NET does not work as well as it should when the value of `topn` is less than 15 on Mini-ImageNet, probably because there are too few local descriptors for it to work well when the value of `topn` is greater than 15, the effect is also worse, this is because when too many local descriptors are involved in the selection, much useless information will be added to interfere with the experiment. Instead, it is the theoretical 15 that works best. The situation applies to CUB-200, but the best choice is 20 instead of the theoretical 15. This is because he is a fine-grained bird dataset, so more local descriptions are favourable to the support set's choice.

6 Conclusion

We propose to enhance few-shot learning with the local descriptors(EFLLD-Net). First, to fully use an image's information, we propose a Feature Fusion Module (FFM) to organically combine the global and local features of an image to obtain local descriptors with global semantic information. Next, in order to reduce the influence of the local background descriptors in the image on the classification results, we propose the Discriminative Feature Highlighting Module (DFHM) to emphasize the more local discriminative features of each category by the relationships between the categories in the support set. Finally, we filter the more discriminative features in the support set through the query set in our proposed BSM and subsequently use the processed support set to feed the process of emphasizing the more discriminative features in the query set. Extensive experimental results and visualizations demonstrate that our approach is practical and achieves state-of-the-art on the few-shot classification benchmark.

Acknowledgments. This work is supported by National Key R&D Program of China (2018YVFA0701700, 2018YFA0701701), NSFC (62176172, 61672364).

References



1. Doersch, C., Gupta, A., Zisserman, A.: Crosstransformers: spatially-aware few-shot transfer. *Adv. Neural. Inf. Process. Syst.* **33**, 21981–21993 (2020)
2. Dong, C., Li, W., Huo: Learning task-aware local representations for few-shot learning. In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 716–722 (2021)
3. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: *Proceedings of Machine Learning Research*, vol. 70, pp. 1126–1135. PMLR (2017)

4. He, K., Pu, N., Lao, M., Bakker, E.M., Lew, M.S.: Dual selective knowledge transfer for few-shot classification. *Appl. Intell.* **53**(22), 27779–27789 (2023). <https://doi.org/10.1007/S10489-023-04994-7>
5. He, K., Zhang, X., et al., S.R.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, June 27-30, 2016, pp. 770–778. IEEE Computer Society (2016)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks, pp. 1106–1114 (2012)
7. Lee, K., Maji, S., Ravichandran, A., Soatto, S.: Meta-learning with differentiable convex optimization, pp. 10657–10665. Computer Vision Foundation / IEEE (2019)
8. Lee, S.B., Moon, W., Seong, H.S., Heo, J.: Task-oriented channel attention for fine-grained few-shot classification. *CoRR* **abs/2308.00093** (2023). <https://doi.org/10.48550/ARXIV.2308.00093>
9. Li, W., Xie, L., Gan, P., Zhao, Y.: Self-supervised pairwise-sample resistance model for few-shot classification. *Appl. Intell.* **53**(18), 20661–20674 (2023). <https://doi.org/10.1007/S10489-023-04525-4>
10. Li, W., Xu, J., Huo, J., Wang: Distribution consistency based covariance metric networks for few-shot learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 8642–8649 (2019)
11. Liu, Y., Zheng, T., and, J.S.: DMN4: few-shot learning via discriminative mutual nearest neighbor neural network, pp. 1828–1836. AAAI Press (2022)
12. Oreshkin, B., Rodríguez López, P., Lacoste, A.: Tadam: task dependent adaptive metric for improved few-shot learning. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
13. Ren, M., Triantafillou, E., Ravi: Meta-learning for semi-supervised few-shot classification. arXiv preprint [arXiv:1803.00676](https://arxiv.org/abs/1803.00676) (2018)
14. Rusu, A.A., et al., D.R.: Meta-learning with latent embedding optimization. In: 7th International Conference on Learning Representations, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019)
15. Simon, C., Koniusz: Adaptive subspaces for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4136–4145 (2020)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
17. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning, pp. 4077–4087 (2017)
18. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning, pp. 1199–1208. Computer Vision Foundation / IEEE Computer Society (2018)
19. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking few-shot image classification: a good embedding is all you need? In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV*, pp. 266–282. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58568-6_16
20. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
21. Wah, C., Branson, S., Welinder, P., Perona: The caltech-ucsd birds-200-2011 dataset (2011)

22. Wertheimer, D., Tang, L., Hariharan, B.: Few-shot classification with feature map reconstruction networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8012–8021 (2021)
23. Xu, J., Ton, J.F., Kim, H., Kosiorek, A., Teh, Y.W.: Metafun: meta-learning with iterative functional updates. In: International Conference on Machine Learning, pp. 10617–10627. PMLR (2020)
24. Zhang, C., Cai, Y., Lin, G., Shen: Deepemd: few-shot image classification with differentiable earth mover’s distance and structured classifiers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12203–12213 (2020)
25. Zhou, Z., Qiu, X., Xie, J., Wu, J., Zhang, C.: Binocular mutual learning for improving few-shot classification. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, pp. 8382–8391. IEEE (2021)



Using Multiscale Information for Improved Optimization-Based Image Attribution

Aniket Singh^(✉)  and Anoop Namboodiri 

International Institute of Information Technology, Hyderabad, India
aniket.singh@research.iiit.ac.in, anoop@iiit.ac.in

Abstract. Image Attribution is the task of ascribing importance to regions of the input, for the final decision of a classifier. Many methods for attribution exist, and a recent development has been to use the image at multiple scales to improve the performance of some lightweight attribution methods. These gains have been demonstrated for methods that require a single or multiple but independent forward passes through the network; however, they haven't been explored in the context of optimization-based attribution (OA) methods. As compared to other techniques, like CAMs or axiomatic methods, OA is attractive as it lends a natural formulation of the task without the need for any heuristic rules. However, the iterative way of solving the optimization problem presents challenges to straightforward utilization of multiple scales. We investigate this scenario and develop a novel 2-step approach that first discovers *promising areas* across scales and locations from the input and then runs Optimization based Attribution on them. We find that while a naive incorporation of image crops is unsuccessful, this 2-stage pipeline leads to improvements in performance. We provide qualitative and quantitative evidence for this and investigate the reasons for the improvement via multiple ablation experiments.

1 Introduction

As the complexity of machine learning models increases, there is a growing need to understand their decision making process. This becomes essential before such complex models are adopted for critical tasks, where failure can be catastrophic.

Measures such as accuracy, precision and recall etc. provide single number summaries that help us choose between approaches. However, it is prudent to explore the behavior of these black boxes comprehensively before relying on them for sensitive tasks.

Model Explanation is one such endeavor, which tries to understand the reasons behind the model's decision. Instead of looking at this from the lens of mathematical operations, it tries to recast the process into human understandable concepts.

An example of such a concept is *what*, i.e. "what input patterns influence the features of the network". This question is addressed by *feature visualization*

methods [13] [29] [32] [24], which map the feature maps back to images, and expose phenomenon like input invariances for the network and preferred patterns for neurons.

Another concept is *where*, i.e. “where are the parts of the input that are important for the network’s decision”. This is answered by *attribution methods*, which are the focus of the current work. Attribution results in saliency maps or heat maps highlighting regions that strongly influence the output.

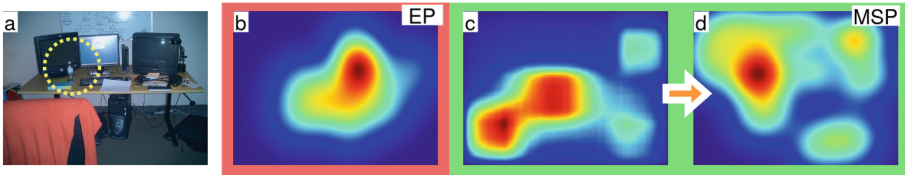


Fig. 1. Comparison of the working of Multiscale Perturbation (MSP) and Extremal Perturbation (EP). (a) Original image. For illustrative purposes the object of interest (Bottle) is highlighted. (b) Attribution produced by EP misses the object. (c,d) MSP first produces a coarse map of *Promising Crops* (c), which is refined into the final attribution (d) which captures the object

While attribution may be realized in many different ways, and we review many such methods in Sect. 2, one way of organizing these techniques is on the basis of the amount of computation required to generate the heat map. Using the number of queries to the classifier as the cost, *Single-Pass* methods like Gradient [22], GradCAM [20], and Guided Backprop [26] require a single forward-backward pass through the network. In contrast, *Batch-Pass* methods like RISE [16] and ScoreCAM [30] perform multiple but independent passes, where their independence allows them to be done in parallel, upto the limits of memory. The assets produced from these independent passes, i.e. the feature maps and the feature gradients, are used in various ways to produce the final heat map. Finally, *Sequential/Iterative* methods like Meaningful Perturbations (MP) [4] or Extremal Perturbations (EP) [3] build the attribution map in an iterative fashion, also requiring multiple forward-backward passes, but unlike batch-pass methods these cannot be parallelized.

Recent work has demonstrated the benefits of using the image at multiple scales in attribution algorithms. For e.g. CAMERAS [9] presents a modification of GradCAM by using multiple zoomed in versions of the image. Another work, SESS [28] presents a general framework that samples crops from the zoomed versions of the image. It then runs a base attribution method like GradCAM on these crops, and aggregates the crop saliencies to yield the final attribution. Note, the CAMERAS pipeline is based on a Single-Pass method, while SESS has been explored on Single-Pass and Batch-Pass methods.

However, neither of these ideas can directly be adapted to sequential attribution schemes. For sequential methods, the attribution for any crop would require

multiple iterations to complete, making the SESS’s methodology of *attribute, then aggregate* impractical. Alternately, a strategy like CAMERAS, to perform attribution on the entire image at any scale, would be prohibitive in computation and memory at higher scales. Further, the batch processing of multiple scales in tandem is complicated due to the difference in their sizes.

To address these challenges, we develop a novel method, that at any step selects a single *promising crop* from the image or its zoom, and runs a single step of an iterative attribution method like EP on it. Central to the scheme is the *crop sampling* module, that identifies crops likely to contain object regions, and extracts them efficiently from the image. Iterative attribution is computationally more demanding than single and batch-pass attribution, which requires the crop sampling module to be lightweight. To work within this constraint, we use a Bayesian model, a Gaussian Process (GP) regressor, to model a single image’s crop importances. Bayesian models quantify their own uncertainty about a prediction, allowing us to pick confidently high scoring crops as well as crops with high uncertainty, to feed to the base iterative attribution method. This ability relaxes the need for a large or carefully collected set of crops from an image for fitting this model.

Figure 1 highlights some aspects of the approach: while Extremal Perturbations (EP) gets the location of a small object (bottle) wrong, our approach, Multiscale Perturbations (MSP), is able to refine the crude importance map of its GP regressor to correctly locate the object.

To summarize, our contributions in the present work are: **(1)** An improved Optimization based Attribution method, by effectively utilizing multi-scale information. Our method is a novel two-stage pipeline that uses a Bayesian model to feed crops for optimization **(2)** Rigorous baselines where we establish the necessity for this approach by demonstrating the ineffectiveness of simpler ways of using multiscale information **(3)** Comprehensive comparisons against contemporary attribution methods, both qualitative and quantitative, to establish the efficacy of our scheme. Our experiments are performed across different CNN architectures to verify generality of the improvements.

2 Related Work

Image Attribution. There are a host of image attribution techniques, which may be grouped according to various commonalities and differences. While we discussed categorization on the basis of passes through the network in Sect. 1, another distinction is between *Heuristic-based* methods and *Perturbation-based* methods.

Heuristic based methods commonly prescribe a set of rules that creates the attribution by combining the quantities created during the forward and backward passes for the image. The utility of gradients as a marker for importance was first discussed in [22], and the subcategory of *Modified Backpropagation* methods specify rules for the class gradients to yield improved attribution. Some

examples include Guided-Backpropagation [26], Layerwise Relevance Propagation [15], Contrastive Layerwise Relevance Propagation [6], DeepLIFT [21], Excitation Backprop [34] and Deconvolutional Networks [33].

Another sub-category, the *Class Activation Map* (CAM) family of methods, define a set of rules to incorporate the feature maps created during the forward pass for attribution. For e.g., GradCAM [20] functions at a late convolutional layer where it combines altered gradients with the feature map to produce the attribution. Subsequent works like LayerCAM [10], GroupCAM [35] and RelevanceCAM [12], GPNN-CAM [25] also use a combination of the features and gradients, differing in the exact manner in which the combination happens.

Perturbation-based methods, test candidate masks by applying them to the input and measuring the effect on neural network outputs. Within this category, *Optimization* based methods solve for the mask by using gradient descent on a loss that differentially measures the change in network output. Extremal Perturbation (EP) [3], IGOS & IGOS++ [18] [11] and Meaningful Perturbations (MP) [4] fall in this family.

The CNN output loss is usually not sufficient to find sensible masks, and inductive biases in the form of auxiliary losses are needed. MP proposes smoothness and area minimality losses as the inductive biases. EP enforces smoothness via a smoothness operator instead of a loss, and replaces area minimality by regressing the mask size towards a fixed pre-chosen area. IGOS++ replaces anisotropic smoothness with a term that takes into consideration the structure of the underlying image.

While Optimization based attribution utilizes gradients to create the mask, *forward-only* schemes forgo the backpropagation step. Starting from a cache of candidate masks, the importance of each mask is measured by masking and forward passing. The candidates are aggregated considering these importances to produce the final attribution. RISE [16] considers random binary masks as the candidates. To reduce the search space, these are upsampled from a small size. The final attribution is a weighted combination of the candidates, where weights are proportional to the class scores predicted by the CNN for the masked images. LIME [19] proposes using a lightweight regressor to model the behavior of the classifier instead of a simple weighted combination of masks. ScoreCAM [30] considers the channels (after normalization) of a late stage feature map as the prospective masks, before using a RISE-like masking-testing-aggregation scheme.

An important design decision for the Perturbation-based schemes is the masking module. Some variants include replacing the values to be suppressed by a constant like 0 or the median color, e.g. in RISE, or replacing the suppressed values by a highly blurred version of the image, e.g. in Meaningful Perturbations and Extremal Perturbations.

Multi-scale information in Attribution. Recent work has explored the effects of presenting the image at multiple zooms. CAMERAS [9] proposes a modification to GradCAM: feature maps and gradients are computed for images

at multiple zooms. After resizing to the common size of the largest feature map, these are aggregated in a manner similar to GradCAM.

SESS [28] proposes a general framework that is in principle applicable to any attribution method. Its procedure may be described as a 2-staged pipeline of attribute and then aggregate. In the inner loop, non-overlapping crops are sampled from the image at multiple scales and the attribution is computed independently for each crop. Finally, these attributions are aggregated in a weighted manner such that crops with higher class scores are weighted more.

Note, these methods first resize the raw image to CNN input size, and then create the zooms from this down-sampled version. Thus, it is fair to compare multi-scale and conventional single scale methods as they start from the same “amount of information”.

As discussed in the introduction 1, both these strategies of utilizing multi-scale information have challenges when adapted for a sequential attribution method like Extremal Perturbation.

Bayesian Models. Our proposed method is a 2 step procedure that builds a proxy model for the action of the classifier in the first step, and in the second step use it to feed regions for Extremal Perturbation (EP) to run on. We use a Gaussian Process model [31] for the proxy, as it can quantify the uncertainty in its predictions. This is useful as it allows us to perform an active learning flavored sampling, where we select both regions the proxy scores highly, as well as regions where the proxy shows high uncertainty. Previously, Gaussian Processes have been explored in connection to attribution in [14], where the methodology was to use it to sample sizes and locations of occluding windows to apply on the image. Here, the notion of scale referred to the sizes of the occlusion windows rather than the zoom of the image.

3 Background: Extremal Perturbations

Extremal Perturbations (EP) [3] is an optimization based attribution method that is a central element of our approach. EP seeks the attribution, represented as a mask m , for an image I and the class of interest c , as seen by a function f representing the CNN. m takes on values in $[0, 1]$ representing the importance of the pixel. The class score (pre-softmax logit) predicted by CNN for class c is represented by f_c , while the predicted probability (post-softmax) by p_c .

We measure the efficacy of the mask by applying it to the image and measuring the impact on the CNN output. We represent the application of the mask to the image by the $\text{Perturb}(I, m)$ operator. While we don’t delve into the details, the crux of Perturb is that the mask modulates the blur level, with higher values retaining pixel information from the original image while lower values replacing them with values from a blurred version of the image. Thus higher mask values retain information while lower values discard it.

Under this setup, EP seeks to find a mask that can maximize the class score predicted by the CNN. We define the confidence loss:

$$\mathcal{L}_f(I, m) = -f_c(\text{Perturb}(I, m)) \quad (1)$$

which is minimized using gradient descent with m as the optimized entity.

Optimizing the confidence loss in isolation, however, is not enough to produce high quality attributions, for multiple reasons:

(i) Ideally, we seek a *sufficient mask*, which reveals the least amount of pixels required for the object to be recognizable. However, the CNN loss L_f seeks to maximize the predicted score, allowing for greedy solutions where the mask may be too large. To address this, EP affixes the size of the sought mask by introducing a hyperparameter a for the area of the mask relative to the image area. This is enforced via a size regularization loss: The mask pixels are sorted by value, and the top a proportion of pixels are encouraged to be 1 and the rest are pushed towards 0. This area loss is represented by $\mathcal{L}_a(m)$, whose exact mathematical form is unnecessary here.

(ii) A second challenge is the adversarial phenomenon [27], observed while using class gradients to optimize the input of a neural network: The optimization produces artefacts indistinguishable from noise to humans, but have an unexpectedly large effect on the network output. In the case of attribution, this manifests as masks with complicated boundaries and exhibiting large variations in importance within small neighbourhoods (high frequency artefacts). Such masks do not correspond to human notions of attribution, but instead highlight adversarial vulnerabilities of the network, which is not the goal here.

To circumvent this problem, EP constrains the search to only smooth masks. We define a pre-mask \hat{m} , of a size smaller than the image, that is upsampled and smoothed by a non-linear convolution to produce the actual mask m . We represent the computation of m from \hat{m} by the `Smooth` operator, where $m = \text{Smooth}(\hat{m})$

This operator may also be understood from the lens of Occam’s razor, which prescribes that simpler explanations for a phenomenon must be preferred: For attribution, a mask with smooth and simple structure should be preferred over a mask with complicated boundaries.

The final EP problem is to solve for the pre-mask \hat{m} (instead of the mask m) with the loss (where λ_a is a weighing factor):

$$\mathcal{L} = \mathcal{L}_f(I, \text{Smooth}(\hat{m})) + \lambda_a \mathcal{L}_a(\text{Smooth}(\hat{m})) \quad (2)$$

Here, λ_a trades off between the emphasis on increasing class score and keeping the span of the mask small. In practice, we anneal λ_a starting from a lower value and increasing it across the iterations. This allows the mask to encompass object areas before it is reduced in size. The scheduling is a delicate balance between the two objectives: setting λ_a to high values too early might not allow L_f to find the object, while if λ_a is too low, the mask encapsulates too large an area, not sufficiently distinguishing the important parts of the image from the rest.

4 Multiscale Perturbations

We now describe our proposed pipeline, Multiscale-Perturbations (MSP). In essence, at any step of MSP we sample a crop from the mask and image to

run a single step of Extremal Perturbation (EP) on. The major novelty of the approach is in the manner these crops are selected. This process of cropping and optimizing is repeated for T iterations. Algorithm 1 lays out the steps involved in the pipeline, while Fig. 2 depicts the steps graphically. The technique involves 2 stages, where a crude model for crop importances is fit in stage I, and used to choose crops for attribution in stage II

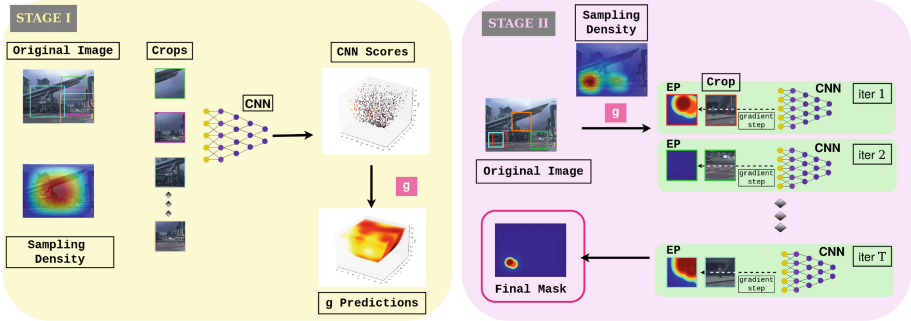


Fig. 2. Multiscale Perturbations Pipeline: **(Left)** In Stage I, we sample crops at random scales and centers from the image. Their classifier scores are used to fit a Gaussian Process g **(Right)** In Stage II, at every step g is used to sample a promising crop from the image, which is used for a step of EP optimization. This process is repeated for T iterations to yield the final attribution.

The crops used in the process are specified by their center points in the original image, (μ, ν) , the zoom factor s relative to the original image, and the crop size (H_w, W_w) . We define the crop operator acting on the image I : $\text{Crop}(I; s, \mu, \nu, H_w, W_w)$. As we use the same crop size throughout, we make it implicit as $\text{Crop}(I; s, \mu, \nu)$, and when speaking of a solitary crop, we further abbreviate it as $\text{Crop}(I)$. Using these crops within EP losses, any stage of MSP optimizes:

$$\mathcal{L} = \mathcal{L}_f(\text{Crop}(I), \text{Crop}(\text{Smooth}(\hat{m}))) + \lambda_a \mathcal{L}_a(\text{Smooth}(\hat{m})) \quad (3)$$

Note, as compared to Eq. 2, while the size regularization \mathcal{L}_a is still applied to the entire mask, the CNN loss \mathcal{L}_f now utilizes the crops.

The core of the algorithm is the *Crop Sampling* module, which defines the selection strategy for the crop coordinates as well as how to efficiently extract them from the image.

Algorithm 1. Multiscale Perturbations

Require: classifier f , image I , class-of-interest c , number of EP steps T , EP area hyperparameter a , budget B of training crops for proxy g (Stage I), budget B_A of crops for multinomial distribution (Stage II).

procedure STAGE I: FITTING PROXY MODEL

- (1.1) Collect B random crops
 $\{\bar{I}_i\} = \{\mathbf{Crop}(I, (s_i, \mu_i, \nu_i))\}$
 at locations $(s_i, \mu_i, \nu_i)_{1..B}$
- (1.2) Acquire scores of crops $y_i = \text{Log}(p_c(\bar{I}_i))$
- (1.3) Fit Gaussian Process g on training set
 $\mathcal{D}_g = \{(s_i, \mu_i, \nu_i), y_i\}_{1..B}$

end procedure

procedure STAGE II: RUN EP ON CROPS

- (2.1) Initialize pre-mask \hat{m}
- for** $t \leftarrow 1$ **to** T **do**
- (2.2) Create smooth mask $m = \text{Smooth}(\hat{m})$
- (2.3) Randomly sample B_A crop locations
 $(s_j^A, \mu_j^A, \nu_j^A)_{1..B_A}$
- (2.4) Sample scores for crops from
 $g : \hat{y}_j^A \sim g((s_j^A, \mu_j^A, \nu_j^A))$
- (2.5) Build Multinomial distribution q_A for the crops with weights as
 $\exp(\hat{y}_j^A)$
- (2.6) Sample crop coordinate as $(s, \mu, \nu) \sim q_A$
- (2.7) Extract mask crop $\bar{m} = \mathbf{Crop}(m; s, \mu, \nu)$
 and image crop $\bar{I} = \mathbf{Crop}(I; s, \mu, \nu)$
- (2.8) Run 1 step of EP using m, \bar{m} and \bar{I} as in Eq 3

end for

end procedure

4.1 Crop Sampling Module

An image may contain large areas that are unrelated to the class of interest. Due to the large number of possible crops, extracting and running attribution on these crops is wasted computation. To address this issue, we develop a 2 stage pipeline.

Fitting a Proxy Model (Fig. 2, Stage-I) (Algorithm 1, Step 1.1): At stage one, we extract a small budget B of crops $\bar{\mathcal{D}}_g := \{\mathbf{Crop}(I; s_i, \mu_i, \nu_i)\}_{1..B}$ from the image at randomly selected (scale, crop-center) tuples, (s_i, μ_i, ν_i) . In Fig. 2 stage I, the *sampling density* depicts the chances of a pixel being seen in these uniformly sampled patches. (Note, the number of patches involving pixels near the boundaries is lesser than the those involving pixels near the center. Uniformly sampling of patches leads to higher pixel sampling density near the center.)

(Algorithm 1, Step 1.2): We capture class probabilities output by the CNN for these crops as $\{p_i\}$. (Algorithm 1 Step 1.3): We then fit a Gaussian Process (GP) [31] model g to predict the value of a crop, $y_i = \log(p_i)$, given its coordinates (s_i, μ_i, ν_i) . Once fit, we can use g to get the value for any candidate

crop by just using its coordinate (s, μ, ν) . Thus, the model g acts as a *proxy*, removing the need for an actual forward pass of $\mathbf{Crop}(I)$ through the CNN f .

In practice, the expensive steps required in stage I are **(i)** the forward pass through the CNN f to collect $\{y_i\}$, and **(ii)** fitting the Gaussian Process g that requires an inversion of a $B \times B$ Covariance matrix. In light of this, we'd prefer the budget B of initial crops be small, but this leads to a trade-off of reduced quality of g 's predictions. In this scenario using a Bayesian model like Gaussian Processes has benefits, as it can explicitly quantify the uncertainty in its predictions. This property is utilized in stage II of the pipeline.

Once g has been fit, we wish to utilize it to feed promising crops to the EP optimization step, whose predicted value according to g is high. However, crops with low predicted value from g but high uncertainty shouldn't be ignored either, as their value hasn't been captured well by the training data and may actually contain object regions. To meet these two requirements, we devise an active learning flavored approach. The process of using the crops for attribution is depicted in Fig. 2, stage II.

Sampling Promising Crops (Algorithm 1 Step 2.3): We randomly select an *initial query set* consisting of B_A crop coordinates $(s_j^{(A)}, \mu_j^{(A)}, \nu_j^{(A)})$ to query g . If g were a simple linear regressor, this would have yielded a scalar prediction for any query. A Gaussian Process, however, predicts a normal distribution for any query, known as the *posterior predictive*. The covariances of this distribution communicate to uncertainty the model has in its predictions. To leverage this information, we *sample* the scores \hat{y}_j^A for the queries from the posterior predictive distribution. High values in \hat{y}_j^A can occur for two types of crops: **(i)** where g has high predicted mean, **(ii)** where g has high uncertainty. We term these 2 types of crops as *Promising Crops*.

(Algorithm 1 Step 2.5): Once we have the sampled scores for the initial query set, we must select a single crop location from it. To encourage exploration we perform a second round of sampling: we build a multinomial distribution q_A with $\exp(\hat{y}_j^A)$ as the importance weights, from which we sample an index k . This refers to the coordinate $(s, \mu, \nu) = (s_k^{(A)}, \mu_k^{(A)}, \nu_k^{(A)})$ (Algorithm 1 Step 2.6). This second round of sampling allows crops whose predicted value may not be the maximum in the first sampling, to be still be selected. In Fig. 2 stage II, the *sampling density* shows which regions are favored by this sampling procedure.

Sampling from g consistently, i.e. respecting covariances, is computationally expensive in the number of crop coordinates B_A . In order to perform this efficiently, we implement our model using GPytorch [5] and use the fast predictive variances method proposed in [17].

Efficient Crop Extraction (Algorithm 1 Step 2.7): Once we have chosen the crop location (s, μ, ν) , we must extract the crop from the image (and the mask). Two straightforward ways of crop extraction are **(i)** Scale & Crop: where we upsample the image by the factor s and then extract the crop of size (H_w, W_w) centered at $(s\mu, s\nu)$, and **(ii)** Crop & Scale: where we extract a crop of size $(\frac{H_w}{s}, \frac{W_w}{s})$ from the image at location (μ, ν) and then upsample it by s .

There are shortcomings with either of these approaches:

- At higher scales, most of the computation done in Scale & Crop is wasted as the sampled area is a fraction of the total area of the upsampled image.
- In Scale & Crop, upsampling by a factor of s might not lead to integral sizes (sH, sW), requiring us to alter s . For Crop & Scale, the initial crop must have boundaries at integral locations in the original image, which may require cropping a larger area than required. Subsequently, a second crop is performed after upsampling to get the correct area. These scenarios make the implementation tedious.
- An implementation detail for EP is that it concurrently searches for multiple masks of different target areas (i.e. different a 's in \mathcal{L}_a in Sect. 3). Thus, we need to supply crops for each of these a 's. As both Scale & Crop, and Crop & Scale produce intermediate quantities of arbitrary dimensions, neither method lends itself to a batched implementation.

We can address these needs of a batched implementation that is memory efficient, with a Spatial Transformer [8] based mechanism.

Assume we are interested in the crop $\mathbf{Crop}(I; s, \mu, \nu)$. We consider how the value at u, v of the crop is computed. This maps to the location u', v' in the original image I . This location may be fractional and not map to an actual pixel in I . So, we use Bilinear Interpolation to do a weighted average of its 4 real neighboring pixels. We define the top-left (tl), top-right (tr), bottom-left (bl), bottom-right (br) pixels as:

$$\begin{aligned} I_{tl} &= I[\lfloor u' \rfloor, \lfloor v' \rfloor] & I_{tr} &= I[\lfloor u' \rfloor, \lceil v' \rceil] \\ I_{bl} &= I[\lceil u' \rceil, \lfloor v' \rfloor] & I_{br} &= I[\lceil u' \rceil, \lceil v' \rceil] \end{aligned} \quad (4)$$

where $\lceil \cdot \rceil$ is the ceiling operator and $\lfloor \cdot \rfloor$ is the floor operator. The value at (u, v) is then given by:

$$\mathbf{Crop}(I)[u, v] = I_{tl}w_{tl} + I_{tr}w_{tr} + I_{bl}w_{bl} + I_{br}w_{br} \quad (5)$$

where the bilinear weights w vary inversely with distance:

$$\begin{aligned} w_{tl} &= (1 - (u' - \lfloor u' \rfloor))(1 - (v' - \lfloor v' \rfloor)) & w_{tr} &= (1 - (u' - \lfloor u' \rfloor))(1 - (\lceil v' \rceil - v')) \\ w_{bl} &= (1 - (\lceil u' \rceil - u'))(1 - (v' - \lfloor v' \rfloor)) & w_{br} &= (1 - (\lceil u' \rceil - u'))(1 - (\lceil v' \rceil - v')) \end{aligned} \quad (6)$$

These equations lend themselves to an efficient implementation that is parallelizable across pixels and crops. These crops can then be fed for 1 step of EP optimization to be performed ([Algorithm 1 Step 2.8](#)).

5 Experiments

Experimental Setup: To validate our approach we conduct experiments on the Pascal/VOC-2007 [2] dataset with 2 different network architectures. We benchmark on the 2007 Test set consisting of 4952 images from 20 classes.

We run experiments using the VGG16 [23] and Resnet50 [7] architectures. This choice is made keeping in mind the dissimilar nature of the two architectures, and is standard in attribution literature. We base our experimentation code on the Torchray [3] library to ensure reproducibility and comparability to previous work.

Implementation Details: Our proxy model g is an exact Gaussian Process regressor with the RBF covariance kernel. While fitting g on the crop scores, the hyperparameters optimized are the *variance* and *lengthscale* associated with the RBF kernel, the *noise variance* associated with the likelihood and the *mean* of the GP. We use the Adam optimizer with a learning rate of 0.01 to optimize these hyperparameters for 50 steps. While sampling crops, we sample from zooms in range $[1, 4]$. We use crops of size $(H_w, W_w) = (224, 224)$. In Stage I, we use $B = 1024$ initial random crops to fit g , while in Stage II, we sample $B_a = 100$ initial crop locations in every step to query g .

EP (and the EP module within MSP) is run with standard hyper-parameters as in Torchray [1]: EP upsamples a pre-mask which is smaller than the image size by a factor of 7. Upsampling is performed using *softmax-pooling*, with a coldness factor of 20. The size regularization weight λ_a is annealed from 300 by a factor of 1.0035 at every iteration. In every alternate iteration, a horizontal flip is applied to the image and mask as an augmentation policy. The EP optimization is run for 800 steps.

Notation. The i^{th} image of the dataset is I_i . There are C classes, and the attribution for I_i for class c is $m_{(i,c)}(u, v)$ where (u, v) refer to spatial dimensions. For I_i , there might be multiple ground-truth bounding boxes for class c , and the j^{th} such box is $\text{bbox}_{(i,c)}^j$.

Pointing Game. To quantify performance we use the *Pointing Game* benchmark defined in [34]: We capture the coordinate of the maximum point of the attribution map. The evaluation scores a hit if this point lies within one of the bounding boxes for the class of interest, and a miss otherwise. Per-class accuracy is calculated and averaged to give the Pointing Game score for the method. Formally,

$$\begin{aligned} \text{Hit}_{(i,c)} &:= \operatorname{argmax}_{(u,v)}(m_{(i,c)}(u, v)) \in \bigcup_j \text{bbox}_{(i,c)}^j \\ \text{Miss}_{(i,c)} &:= \operatorname{argmax}_{(u,v)}(m_{(i,c)}(u, v)) \notin \bigcup_j \text{bbox}_{(i,c)}^j \\ \text{PG} &= \frac{1}{C} \sum_c \frac{\sum_i \text{Hit}_{(i,c)}}{\sum_i \text{Hit}_{(i,c)} + \sum_i \text{Miss}_{(i,c)}} \end{aligned} \quad (7)$$

The protocol in [34] also defines a separate evaluation on a *difficult* subset of images: This consists of samples where the object of the class of interest is

Table 1. (a): Pointing Game Comparison between MSP and several popular methods on the Pointing Game benchmark (Sect. 5). (* denotes average over 3 runs with different random seeds) Best in bold, second-best underlined. *Grad*: gradient-based *Ptb*: Perturbation-based *Scale*: uses multiscale information **(b):** Comparison of MSP against several baselines (see Sect. 5). Best in bold, second-best underlined. Simpler methods of incorporating multi-scale crops into the pipeline (EP+crops & EP+crops+W) are unsuccessful at improving over EP. The proxy model (GP) when used by itself is a poor attribution method

Method	VOC07 Test (All/Diff)			Grad	Ptb	Scale
	VGG16	ResNet50				
Grad [22]	76.3/56.9	72.3/56.8		✓	✗	✗
DConv [33]	67.5/44.2	68.6/44.7		✓	✗	✗
Guid. [26]	75.9/53.0	77.2/59.4		✓	✗	✗
MWP [34]	77.1/56.6	84.4/70.8		✓	✗	✗
cMWP [34]	79.9/66.5	90.7/82.1		✓	✗	✗
RISE * [16]	86.9/75.1	86.4/78.8		✗	✓	✗
GCAM [20]	86.6/74.0	90.4/82.3		✓	✗	✗
SESS [28]	90.4/80.8	<u>93.0/86.1</u>		✓	✗	✓
CAMERAS [9]	86.2/76.2	94.2/88.8		✓	✗	✓
EP * [3]	88.0/76.1	88.9/78.7		✓	✓	✗
MSP (Ours)*	<u>90.3/79.9</u>	90.8/81.5		✓	✓	✓

Method	VOC07 Test (All/Diff)	
	VGG16	ResNet50
EP	88.0/76.1	<u>88.9/78.7</u>
MSP (Ours)	90.3/79.9	90.8/81.5
EP+crops	86.7/74.1	87.2/75.6
EP+crops+W	<u>88.8/77.3</u>	88.1/74.0
GP	69.5/54.8	70.4/59.4

smaller than 25% of the total image area, and is present alongside at least one object of another class (referred to as *distractors*).

Table 1 (a) compares the performance of Multiscale Perturbations (MSP) against contemporary methods. While SESS or CAMERAS score higher on the benchmark, MSP leads to tangible improvements over the standard Extremal Perturbation algorithm, validating the usage of crops. Table 2 compares the Pointing Game performance per class on the VOC-2007 test set. It can be seen that the improvements stem from generally improved performance across most categories.

Ablation Studies. We investigate these reasons behind the improvements: **(i)** Our proposed pipeline prioritizes the sampling of *Promising Crops* (defined in Sect. 4). We compare this approach against a baseline where the crops are sampled uniformly without consideration of their potential value. We dub this the *EP+crops* variant. As table 1 (b) shows, this approach is unsuccessful in improving performance over standard EP. **(ii)** While MSP uses importance sampling, an alternate way of incorporating crop importances is to sample uniformly but weigh the loss of each patch in proportion to its importance. Specifically, we use the ratio of the predicted class probability for the crop vs the class probability for the original image as the weight. The class probability for the crop is acquired via a separate forward pass through the network. This leads to the *EP+crops+W* variant. Table 1 (b) shows that this approach too does not lead

Table 2. Per class comparison between MSP and EP on the Pointing Game. Table compares scores on the 20 VOC-2007 classes. MSP improves upon EP on most classes

Model	Method	Classes 0-9									
		Airplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
VGG16	EP	97.5	96.7	92.9	90.7	60.8	91.4	86.0	97.5	64.3	95.3
	MSP	98.0	95.1	93.8	92.8	65.4	94.4	87.9	98.0	66.2	98.1
ResNet50	EP	97.1	93.7	88.7	89.0	57.5	94.3	88.9	96.0	71.9	100.0
	MSP	96.1	95.9	94.3	90.9	62.1	92.0	89.6	98.1	73.1	98.7
Model	Method	Classes 10-19									
		Dining-Table	Dog	Horse	Motor-Bike	Person	Potted-Plant	Sheep	Sofa	Train	TV-Monitor
VGG16	EP	83.7	96.4	97.1	96.8	90.5	74.6	94.8	84.8	93.8	81.2
	MSP	84.0	98.3	97.4	96.2	93.5	73.5	95.9	91.9	97.2	87.9
ResNet50	EP	87.9	95.9	97.1	96.4	89.1	75.4	97.9	81.2	92.7	79.0
	MSP	88.6	99.2	97.6	98.6	93.5	76.6	97.5	90.7	96.9	86.6

to improved performance. The performance of such baselines suggests that sampling less valuable patches is detrimental to performance, motivating the need for importance sampling.

In principle, the Gaussian Process g that is the proxy model in for our approach (see Sect. 1) is also an attribution method on its own, as it captures the importance of regions of the image. In light of this, our 2 stage pipeline can be seen as refining the importance captured by g . We can measure the performance of standalone g as GP -saliency on the Pointing Game in table 1 (b), where it scores poorly. This suggests that while it is useful as a feeder of promising crops, its modeling is too crude to capture importances completely.

Pointing Game++ While the original Pointing Game defines a difficult split consisting of smaller objects alongside distractors, we propose a system for rating the difficulty for any sample. The intuition is that the smaller the ground truth area occupied by the class, the more difficult it is for an attribution method to score a hit. In contrast, objects that occupy almost all of the image are near impossible to miss. Thus we propose a new metric, Pointing Game ++ (PG ++), which takes into account the difficulty of scoring a hit while aggregating the results. Formally,

$$\text{WeightedHit}_{(i,c)} := \text{Hit} \times \frac{\text{Area}(\bigcup_j \text{bbox}_{(i,c)}^j)}{\text{Area}(I_i)} \quad (8)$$

$$\text{PG}++ = \frac{1}{C} \sum_c \frac{\sum_i \text{WeightedHit}_{(i,c)}}{P_c^*} \quad P_c^* = \sum_i \frac{\text{Area}(\bigcup_j \text{bbox}_{(i,c)}^j)}{\text{Area}(I_i)} \quad (9)$$

Here P_c^* is a normalizer that represents the maximum weighted hits that can be achieved for a class. It can be seen in table 3 that MSP improves upon EP on PG++, implying a general ability to do better on smaller (harder) cases.

Table 3. Pointing Game ++ (PG++) Energy Pointing Game (EPG) scores of EP and MSP for VGG16 and ResNet50. MSP shows improvement in correctly placing the saliency peak within harder (smaller) object as well as aligning the saliency within the object region

Method	VGG16		ResNet50	
	PG++ ↑	EPG ↑	PG++ ↑	EPG ↑
EP	0.82	0.54	0.83	0.55
MSP	0.84	0.56	0.85	0.56

Energy Pointing Game. The Energy Pointing Game (EPG) [30] studies the overall structure of the attribution map instead of just the peak point as in the Pointing Game. Considering that an attribution contains values in the range $[0, 1]$, the energy of a map is defined as the sum of its values. The EPG score is then defined as the ratio of the energy within the ground truth region to the total energy of the map. Formally,

$$\begin{aligned}
 \text{TotalEnergy} &:= \sum_{u,v} m_{(i,c)}(u,v) \\
 \text{EnergyRatio} &:= \frac{\sum_{u,v} m_{(i,c)}(u,v) \cap (\bigcup_j \text{bbox}_j(i,c))}{\text{TotalEnergy}(m_{(i,c)})} \\
 \text{EPG} &= \frac{1}{C} \sum_c \frac{\sum_i \text{EnergyRatio}_{(i,c)}}{\sum_i 1} \tag{10}
 \end{aligned}$$

We use EPG to compare MSP and EP in our experimental settings in Table 3, where MSP shows improvement in aligning the saliency with the ground truth regions.

Comparison with SESS and CAMERAS. We hypothesize that the smoothing involved in EP, and consequently MSP, might be limiting its performance compared to SESS and CAMERAS. To explore this idea, we compared the total energy (Eq. 10) of these methods. In table 4a, it can be seen for any model, CAMERAS and SESS have smaller energy than MSP and EP, implying more succinct heatmaps. While MSP improves over EP, it falls short of SESS and CAMERAS. This finding is consistent with the hypothesis that the smoothing operator in EP (and MSP) might be spreading the saliency over a larger area, leading to decreased performance.

Table 4. (a) Average value (Total Energy) of the saliency maps produced by EP, MSP, CAMERAS, SESS. Lower values imply more succinct heat maps. MSP improves upon EP, yet falls short of CAMERAS and SESS. The smaller, more precise heat maps might explain the superior performance of CAMERAS and SESS (b) Time (in seconds) taken by MSP and EP to finish a run. MSP matches EP in speed on VGG16, while incurs a slightly higher time cost on ResNet50.

(a) Average Total Energy			(b) Time (s) per Run		
Method	Average Energy ↓		Average Time ↓		Model
	VGG16	ResNet50	VGG16	ResNet50	
EP	0.19	0.18			
MSP	0.18	0.18	59.7 ± 0.71	37.6 ± 0.67	MSP
SESS	0.10	0.10			
CAMERAS	0.04	0.06			EP
			60.23 ± 3.32	35.4 ± 1.63	

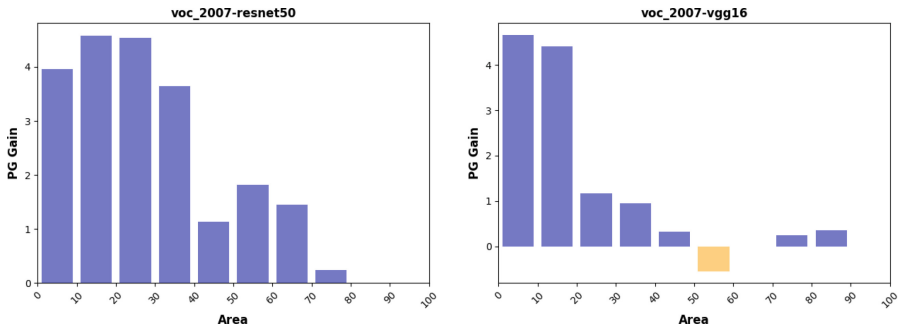


Fig. 3. Difference in MSP and EP performance as a function of the object size: We bin the object sizes into 10 bins and measure the improvement in Pointing Game performance per bin. Left: ResNet50, Right: VGG16. For ResNet50, MSP outperforms EP at all sizes, while for VGG16, it falls behind at just the 50% – 60% bin.

Timing Comparison. In table 4b, we compare the average time taken for computing MSP and EP over 500 images of VOC-2007 on a RTX-3060 GPU. The results are interesting: MSP is as fast as EP on VGG-16, and only slightly more expensive on ResNet-50 (approximately 6%). It may seem paradoxical that a 2-stage procedure incurs such minimal overhead. This can be explained with the help of an example: Consider an image of original size (353, 500), which is resized to (224, 317) for attribution. In the case of EP, the optimization steps are performed on (224, 317) sized input for all the iterations. In contrast, for MSP, there is an initial time cost for aggregating the CNN scores for B zoom-crops of size (224, 224) from the (224, 317) sized image, and fitting a Gaussian Process model. However, the subsequent optimization steps are performed on the smaller (224, 224) sized crops, which is computationally cheaper. This gain in speed accrues over the optimization iterations, allowing MSP to catch up to EP. Thus, the improved performance comes essentially for cheap, in terms

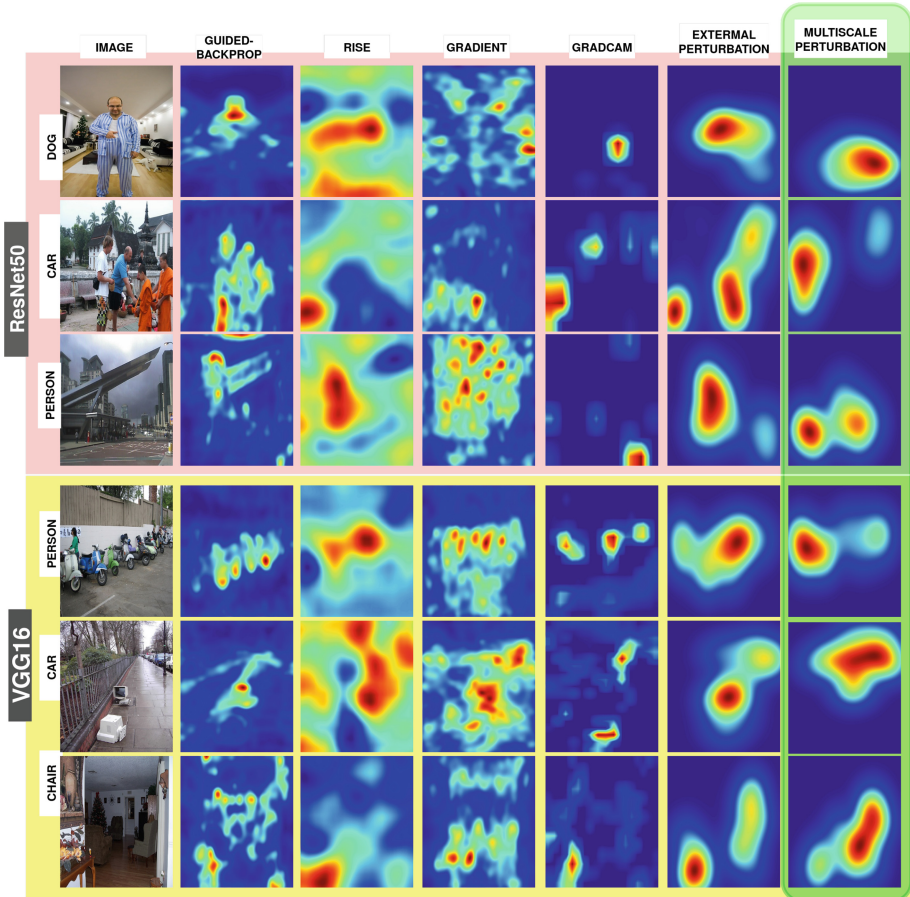


Fig. 4. Qualitative Comparison of MSP with several other methods, on ResNet50 (Top, in pink), and VGG16 (Bottom, in yellow). (Color figure online) on VOC-2007. Zoom in to see small objects. See Sect. 5 for details.

of computational cost. Further, the EP times show higher variance than MSP times, suggesting sensitivity of computation time to image sizes.

Size Study. We compare the performance of MSP and EP as a function of the object size. As an image may have multiple instances of the class, we take the area of the union of all bounding boxes for the class. We quantize the spans into 10 bins, and measure the pointing game scores for MSP and EP for these bins. In Fig. 3 we plot the difference in PG accuracy between MSP and EP as a histogram: For ResNet50, MSP outperforms EP at all sizes, while for VGG16, it falls behind at just the 50% – 60% bin. This scenario is interesting for further

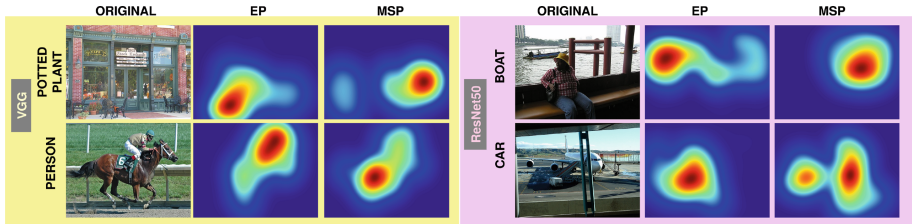


Fig. 5. Failure cases on VGG16 (left, in yellow) and ResNet50 (right, in pink). On these VOC-2007 samples EP performs better than MSP. These are subject for further investigation (Color figure online)

study. For any model, the gains are the steepest at the smallest sizes, validating the zoom-crop methodology of MSP.

Qualitative Comparison: Figure 4 compares importance maps created by various attribution methods. It can be seen that MSP performs favorably against several competing methods. Note that for EP & MSP, the attribution is computed independently for 4 different values of the area hyper-parameter a (see Eqs. 2 and 3), and smoothed to give the final attribution seen in the figure. This is the standard protocol defined by the original EP implementation [1], where the areas are [2.5%, 5%, 10%, 20%] of the image size. In some cases, however, MSP shows worse performance than EP (Fig. 5). These cases are subject of further investigation.

6 Conclusion

We investigated the challenges and benefits of utilizing the image at multiple scales for Optimization based Attribution. While such ideas have been explored with lightweight attribution schemes like single-pass and batch-pass methods, they face challenges with regard to iterative/sequential attribution methods (see Sect. 1).

Our pipeline improves upon such an iterative method, Extremal Perturbations (EP) [3], by incorporating crops from the image at various scales. Via comparative studies we establish the ineffectiveness of naively introducing crops to the EP algorithm. To meet this challenge, we devise a novel two-stage pipeline, where stage I fits a lightweight regressor for modeling crop importances of an image, and stage II leverages it to sample *promising crops* to feed to the EP module. Owing to the computational needs of iterative attribution, such a pipeline must limit the amount of overhead it adds. We discuss aspects of the scheme such as the regressor type and the design of the crop extraction module that meet these demands. Finally, we quantify the effectiveness of our approach, Multiscale Perturbations (MSP), by demonstrating improved performance on the Pointing Game [34] benchmark with different CNN architectures. We perform ablation

experiments against baselines to investigate the reasons for improvement, as well as provide qualitative results from our technique, and failure cases.

References

1. Torchray (2019). <https://github.com/facebookresearch/TorchRay>
2. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>
3. Fong, R., Patrick, M., Vedaldi, A.: Understanding deep networks via extremal perturbations and smooth masks. In: ICCV, pp. 2950–2958 (2019)
4. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. In: ICCV, pp. 3429–3437 (2017)
5. Gardner, J., Pleiss, G., Weinberger, K.Q., Bindel, D., Wilson, A.G.: Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *NeurIPS* **31** (2018)
6. Gu, J., Yang, Y., Tresp, V.: Understanding individual decisions of cnns via contrastive backpropagation (2018). <https://doi.org/10.48550/ARXIV.1812.02100>
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
8. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. *NeurIPS* **28** (2015)
9. Jalwana, M.A.A.K., Akhtar, N., Bennamoun, M., Mian, A.: Cameras: Enhanced resolution and sanity preserving class activation mapping for image saliency (2021)
10. Jiang, P.T., Zhang, C.B., Hou, Q., Cheng, M.M., Wei, Y.: Layercam: exploring hierarchical class activation maps for localization. *IEEE Trans. Image Process.* **30**, 5875–5888 (2021)
11. Khorram, S., Lawson, T., Fuxin, L.: igos++ integrated gradient optimized saliency by bilateral perturbations. In: Proceedings of the Conference on Health, Inference, and Learning, pp. 174–182 (2021)
12. Lee, J.R., Kim, S., Park, I., Eo, T., Hwang, D.: Relevance-cam: Your model already knows where to look. In: CVPR, pp. 14944–14953 (2021)
13. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: CVPR, pp. 5188–5196 (2015)
14. Mokuwe, M., Burke, M., Bosman, A.S.: Black-box saliency map generation using bayesian optimisation. In: IJCNN, pp. 1–8. IEEE (2020)
15. Montavon, G., Binder, A., Lapuschkin, S., Samek, W., Müller, K.R.: Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209 (2019)
16. Petsiuk, V., Das, A., Saenko, K.: Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421* (2018)
17. Pleiss, G., Gardner, J., Weinberger, K., Wilson, A.G.: Constant-time predictive distributions for gaussian processes. In: ICML, pp. 4114–4123. PMLR (2018)
18. Qi, Z., Khorram, S., Li, F.: Visualizing deep networks by optimizing with integrated gradients. In: CVPR Workshops. vol. 2, pp. 1–4 (2019)
19. Ribeiro, M.T., Singh, S., Guestrin, C.: ” why should i trust you?” explaining the predictions of any classifier. In: KDD, pp. 1135–1144 (2016)
20. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: ICCV, pp. 618–626 (2017)

21. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: ICML, pp. 3145–3153. PMLR (2017)
22. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint [arXiv:1312.6034](https://arxiv.org/abs/1312.6034) (2013)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
24. Singh, A., Namboodiri, A.: Laplacian pyramids for deep feature inversion. In: ACPR, pp. 286–290. IEEE (2015)
25. Singh, A., Namboodiri, A.: Image attribution by generating images. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5990–5994. IEEE (2024)
26. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint [arXiv:1412.6806](https://arxiv.org/abs/1412.6806) (2014)
27. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint [arXiv:1312.6199](https://arxiv.org/abs/1312.6199) (2013)
28. Tursun, O., Denman, S., Sridharan, S., Fookes, C.: SESS: saliency enhancing with scaling and sliding. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII, pp. 318–333. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-19775-8_19
29. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: CVPR, pp. 9446–9454 (2018)
30. Wang, H., et al.: Score-cam: Score-weighted visual explanations for convolutional neural networks. In: CVPR Workshops, pp. 24–25 (2020)
31. Williams, C.K., Rasmussen, C.E.: Gaussian Processes For Machine Learning, vol. 2. MIT press Cambridge, MA (2006)
32. Yin, H., et al.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: CVPR, pp. 8715–8724 (2020)
33. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I, pp. 818–833. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
34. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. IJCV **126**(10), 1084–1102 (2018)
35. Zhang, Q., Rao, L., Yang, Y.: Group-cam: Group score-weighted visual explanations for deep convolutional networks. arXiv preprint [arXiv:2103.13859](https://arxiv.org/abs/2103.13859) (2021)



Split-DNN Computing for Video Analytics

Nagabhushan Esvara², Jaroslaw Sydir³, V. Srinivasa Somayazulu¹,
Parul Datta², Nilesh Ahuja³(✉), and Omesh Tickoo¹

¹ Intel Labs, Hillsboro, USA

² Bangalore, India

³ Santa Clara, USA

nilesh.ahuja@intel.com

Abstract. Optimization of Visual AI applications for next-generation networked and distributed edge scenarios is an important and challenging problem area given the computation, power and bandwidth resource constraints of client devices and edge servers. Dynamically adapting to variations in system resource availability and optimizing the trade-offs in accuracy vs. compression rate and computational complexity is important for system efficiency. An emerging paradigm for the deployment of complex Deep Neural Network (DNN) models for video analytics in these edge computing scenarios is split-DNN computing, where the DNN model is partitioned with one part executed on a client device and the other part on an edge server. Earlier work has largely addressed split-DNN computing in the context of image analytics. However, the application to video sequences presents significant challenges of computational complexity. In this paper, we propose a flexible and low-complexity approach to address these specific challenges for distributed DNN-based video analytics and semantics-preserving learned compression. We combine lightweight bottleneck encoder-decoder neural networks for compressing deep feature representations along with optical flow-based warping of these deep features. We demonstrate significant compression gains measured with a BD-Rate of -82.68% for object detection and -59.31% for segmentation when compared with the earlier image-based analytics and compression approaches, and even larger gains over conventional video compression. In addition, we enable dynamic optimization of split-DNN video analytics at the edge by providing lightweight training and inference approaches with simple solutions for fine-grained adaptation in the complexity-rate-accuracy space.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78122-3_21.

1 Introduction

The proliferation¹ of real-time visual AI inference at the edge is driving requirements for processing massive amounts of visual data with computationally heavy deep neural network (DNN) models, a significant challenge for power- and compute-constrained end devices. The rise of next generation networks with improved throughput, latency, and reliability is enabling offloading [3] of this visual AI computing by compressing video streams using video compression standards such as H.265/HEVC [13] for transmitting to a network edge server. While this can help address the limitations of processing at the client devices, this approach is sub-optimal since the AI compute must be fully offloaded to the edge server, precluding collaborative intelligence that exploits all available compute at both clients and edge servers. Additionally, conventional video compression is optimized for human visual perception, and the compression artifacts can seriously impact the accuracy of DNN-based video analytics. Recent advances in neural compression algorithms for either images or videos [1, 16, 23, 25] are also largely optimized for perceptual quality and add additional AI computational complexity in the edge compute scenarios since the video must be first reconstructed prior to analytics.

In parallel, there has been considerable research into optimal compression of images and videos for machine vision tasks. Prominent among this has been research into split-DNN computing [6, 7, 15, 18, 19], where the DNN trained for a specific task (or set of tasks) is split into a front-end (head) and back-end (tail) portions, and implemented in a distributed manner across a client device and an edge-server. Intermediate deep feature representations from the head portion are compressed and transmitted to the tail residing on an edge server where the remaining processing is completed. The compression of the features is learnt by jointly optimizing both for task-specific performance (e.g., classification, object detection, or image segmentation etc.) as well as compression efficiency. A key attribute of many of these approaches is the introduction of bottleneck layers at the split point of the DNN to reduce the dimension of the transmitted features [6, 7, 19] followed by network distillation to reduce the complexity and optimize performance. However, these approaches have largely focused on the image analytics pipeline, and do not exploit temporal correlations inherent in videos to drive further gains in the compression-accuracy tradeoff. Furthermore, although the proportion of compute between the client and edge can be adapted, the total end-to-end compute across the network remains the same.

1.1 Contribution

We summarize our contributions below:

- We propose a solution for distributed video analytics with split-DNN computing and end-to-end learned semantics-preserving video compression. To the

¹ Code to reproduce results at <https://github.com/IntelLabs/SPVC>.

best of our knowledge, this is the first work that exploits temporal coherence in the deep-feature space to optimize for accuracy-rate-complexity tradeoff in distributed video analytics rather than frame-by-frame image analytics. This is achieved by compressing features from a few selected key-frames using bottleneck modules (similar to [6]), and predicting non-key-frame features from the closest key-frame features via flow-based feature warping.

- We introduce key-frame interval as a parameter in split-DNN computing for additionally adapting client compute complexity and enabling an accuracy-rate-complexity trade-off. This allows system designers to have greater flexibility in managing and balancing workloads in compute-constrained edge scenarios, while maintaining task performance. This is in contrast to image analytics solutions such as [6, 7, 18] which allow for a accuracy-rate tradeoff, with limited ability to adapt the compute complexity beyond the DNN split between client and edge server.
- Finally, we evaluate our approach over two workloads, namely, (i) semantic segmentation and (ii) object detection, on true video datasets instead of image datasets, and we demonstrate how our approach outperforms image-based analytic approaches and conventional compression on such datasets.

In summary, our approach, which we call Semantic Video Compression with Feature Warping (SVC-FW), addresses the problem of compression for video-analytics (as opposed to images only), is well-suited for low-complexity client devices and also enables a flexible adaptation to trade-off rate, accuracy and client complexity.

2 Background

2.1 Semantic Image Compression with Bottleneck Units

Several split-DNN approaches for image analytics with feature compression introduce ‘bottleneck units’ at the split point - lightweight neural networks that reduce the dimension of the features prior to compression and encoding, and restore them to their original size after decoding. Most approaches employing bottleneck layers nevertheless require a retraining of the original DNN’s parameters (or at least the head portion) [7, 18] if a different split point or compression level is desired. A different approach was taken in [6] where the original DNN parameters were preserved unchanged across different splits. Instead, only the bottleneck modules were trained at each split point, with different parametrizations yielding different compression levels. Since only a small set of low complexity bottleneck modules need to be trained and loaded dynamically, such an approach promises to better address the constraints of real-world systems.

A procedure to design the bottleneck units in an optimized manner was also provided in [6], which we briefly summarize next. The procedure involves exploring the space of architectural hyper-parameters of a single bottleneck encoder layer such as number of output channels and stride of the convolutional kernel. The procedure can be generalized to include other design hyper-parameters

such as number of layers, topology of the layers, etc. A sample from this hyperparameter space is generated, and a bottleneck unit with these parameters is instantiated. The weights of this bottleneck unit are then trained without modifying the weights of the original network. The accuracy of the pipeline with the trained bottleneck is measured along with the average bit-rate required to transmit the compressed features. This process is repeated multiple times with an appropriate sampling procedure (random-search, grid-search, Bayesian approaches, etc.) to generate multiple bottleneck units at different rate-accuracy points. From this set, the Pareto optimal set of points are determined and these correspond to the set of trained bottleneck layers that yield the optimal rate-accuracy performance.

2.2 Deep Feature Prediction

Recognizing that high-level semantic information represented by deep features within a DNN evolve more slowly than the pixel level image appearances [14] motivated video compression and analytics approaches exploiting temporal coherence in the deep feature domain [10, 16, 17, 28]. In [17] for example, a deep video compression approach is based upon computing the latent features from a set of past frames and using them as input to a prediction network. The residual error between the actual and predicted latent features of the current frame are compressed and transmitted to the decoder. Adapting approaches such as this for efficient split-DNN edge video analytics and compression would be limited by the significant computational complexity of the prediction networks as well as the requirement to compute latent features for all or a large fraction of the input frames. In [10], learned motion estimation and deformable convolutions in the feature space are combined with feature prediction residual coding for deep video compression optimized for human visual perception metrics. However, the motion estimation in the feature space incurs both complexity and compression efficiency impacts, while the requirement to compute latent features for all frames is still a significant issue for computationally efficient edge analytics. Using motion vectors estimated in the pixel domain for temporal prediction of deep features in deep video compression offers a solution with lower complexity. In [16], optical flow estimated between current and previous frames is used to generate warped features that provide the context for compression of the latent representation of the current frame produced by an encoder/decoder network. However, all these and related approaches have focused on learned video compression and are not appropriate for efficient edge video analytics with split-DNNs. Sparse deep feature propagation as in [28] and related works employed optical flow computed in the pixel domain and combined this with relatively simple deep feature warping in order to reduce average computational complexity for DNN workloads.

In what follows, we outline our split-DNN video analytics approach that combines an efficient image analytics and semantics preserving compression solution on a sparse set of frames together with compressed learned optical flow for the remaining frames.

3 Our Approach

Our approach is based on exploiting the temporal correlations in the feature space of deep networks to achieve greater compression efficiency for distributed analytics. Similar to standard video-compression approaches (VVC, HEVC, etc.), a small set of video frames are selected as key-frames and the remaining are designated as non-key frames, as in [28]. The deep-feature representations from key-frames are compressed using lightweight bottleneck encoder module following the approach in [6]. These key-frame features serve as reference points relative to which the non-key frame features are calculated using the optical flow computed in the pixel space. For non-key frames, only the compressed optical flow is transmitted to the edge-server. At the server, a feature-warping module warps the relevant key-frame features using the flow information for a particular non-key frame to predict the deep features for that frame. This approach yields several benefits over existing image-based approaches –

1. Motion information requires far fewer bits to compress and transmit, enabling better compression than image only approaches.
2. The flow network is designed to be much more lightweight than the original backbone network. Hence, the computationally heavier backbone needs to be run only for key-frames and this reduces the overall computational complexity significantly.
3. Since the flow network is trained for the task (or tasks) of interest, the resultant motion field is optimized for those tasks. This helps improve the overall task accuracy.

3.1 Semantics-Preserving Video Compression with Flow Based Feature Warping

The overall flow diagram of our approach is shown in Fig. 1. A DNN model trained for a particular task typically comprises a backbone followed by a task network. The Deeplab-v3 model used in our experiments for video semantic segmentation comprises a Resnet-50 model (backbone) followed by an ASPP module (task). Similarly, the Faster-RCNN used for video object detection, comprises a Resnet-50 backbone followed by the region-proposal network (RPN) and prediction modules as task layers. We start by partitioning the backbone network into a ‘head’ portion that is deployed on the client side, and a ‘tail’ network deployed on the server. This is followed by the task network that produces the analytics task-specific outputs (e.g. for object detection, image segmentation, etc.). The flow of operations is as follows. At the client device,

- A sparse set of input video frames are selected as key-frames, with $N - 1$ non-key frames following every key frame, as in [28]. N is called the key-frame interval. The process of selecting key-frames and the key-frame interval can be adaptive, based upon the input context, the analytics task requirements, the network and compute resource availability, etc.

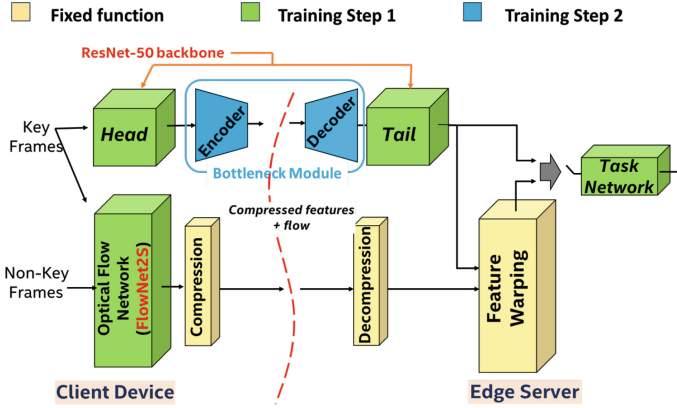


Fig. 1. The proposed distributed video analytics pipeline SVC-FW combining deep feature compression and compression of optical flow for feature warping.

- The key-frames are processed through the head network to extract deep feature representations as depicted in the upper branch of Fig. 1. These key-frame head network features are then compressed for transmission to the edge server using a lightweight bottleneck encoder module following the approach in [6]. A set of optimized lightweight bottleneck encoder-decoder modules is derived for different rate-accuracy points to enable efficient and flexible compression of the key-frame deep features.
- For each non-key frame, optical flow information is calculated between it and the nearest key-frame, compressed and transmitted to the edge server.

At the edge server,

- A bottleneck decoder is employed to decompress the key-frame features and feed them to the tail network for the next step in the DNN processing. The results are processed by the task-specific network, as well as stored for processing the relevant non-key frames.
- For each of the non-key frames, the optical flow is decompressed and then used to warp the relevant key-frame features in order to predict the deep features for that frame. In our work, bilinear interpolation is used as the warping function.
- Reconstructed deep features at the output of the tail network for both key-frames and non-key frames are processed through the task network to generate the DNN output analytics results (e.g. object bounding boxes, or segmentation maps, etc.).

For the non-key frame warping, there are two variants: (1) ‘forward’ warping, in which the optical flow computation is with respect to the most recent past key-frame, possessing the virtue of a low-delay system implementation (2) ‘bi-directional’ warping wherein the flow computation is with respect to the nearest

key-frame, past or future - which can result in superior rate-accuracy performance since the average distance of a non-key frame to the reference key-frame is reduced, though at the cost of increased delay.

3.2 Models and Training

To develop our proposed SVC-FW model, we adopt a 2-step training process as shown in Fig. 1. We start with the baseline analytics model that has been trained with the relevant video dataset using a task-appropriate end-to-end loss function. In Step 1, we train the parameters for the head, tail and task networks, as well as the flow network shown in green blocks in Fig. 1, which we collectively refer to as the Feature Warping with Flow, or FWF model. This FWF model is trained end-to-end by initializing the optical flow generator network using the weights from the pre-trained FlowNet2S model [11] and the rest of the network using the weights from the baseline model, and with the task-appropriate loss function.

The model weights from Step 1 are next used in Step 2 for training bottleneck modules for all DNN splits. First, a split version of the FWF model from Step 1 is generated at the desired DNN split point to provide the head and tail/task network parameters. Next, bottleneck encoder and decoder networks (shown in blue in Fig. 1) are included and trained. The head/tail/task network parameters as well as the flow network parameters are frozen and only the bottleneck encoder-decoder module parameters are trained starting with random weight initialization, and with the loss function as in [6].

$$L = L_t + \alpha L_r \quad (1)$$

with the difference that the task-specific loss term L_t is evaluated over both key-frames and non-key frames; L_r is the rate-loss term, and α controls the relative

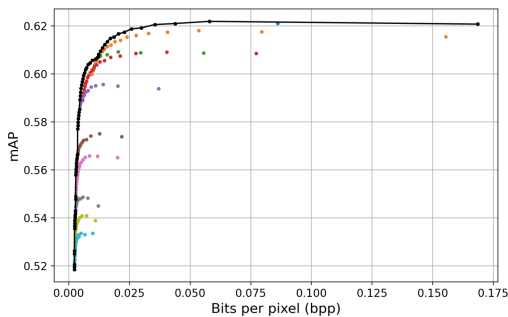


Fig. 2. Illustration of the hyper-parameter search procedure to determine the configurations providing optimal rate-accuracy performance. Each color represents a different optimal key-frame compression starting point with different points corresponding to increasing key-frame intervals as we move to the left.

weighting of the two terms, each setting generating models with different rate-accuracy performance. The combined FWF model together with the bottleneck modules for the key-frames is our SVC-FW model.

During inference, our SVC-FW model is initialized by loading the head network, flow network, and tail network weights from the FWF model in Step 1. The appropriate bottleneck encoder-decoder module(s) from Step 2 are selected together with the key-frame interval(s) based upon the rate-accuracy-complexity operating points desired. Because we perform warping at the end of the tail network, the head, flow, and tail networks are unchanged for all splits, and only the lightweight bottleneck encoder-decoder module(s) need to be loaded for each operating split point.

Bottleneck Design and Training. The procedure to design and train the bottleneck units for a desired split-point involves a search over a hyper-parameter space as described in [6] and summarized in Sect. 2.1, which includes (i) C_{feat} , the number of bottleneck encoder-decoder channels, (ii) S , the stride for the encoder convolutional layer, (iii) Q , the quantization parameter, and (iv) $\log_{10}\alpha$, where α is the rate loss weight term from Eq. 1. For video analytics, key-frame interval is an additional fifth parameter. Different from the image analytics case, therefore, the training is performed using *both key-frames and non-key frames* and the metric employed for search-space optimization is the average performance of SVC-FW over an entire key frame interval (i.e., average accuracy over a key frame and associated non-key frames) set to a fixed, maximum value of 20. Performing a hyperparameter search over this extended five dimensional parameter space would involve sampling from this space followed by training and evaluation of models to guide the Bayesian optimization search process. In order to reduce the complexity of the overall process, we follow a decoupled search process, where the four-dimensional hyperparameter space for the key-frame compression is first explored and the Pareto optimal configuration settings are derived. These are then used for the evaluation of SVC-FW across a range of key frame intervals.

In Fig. 2 we illustrate this process for an object detection task with a Faster R-CNN model with a split at the `res4` layer. The rate-accuracy points for different key-frame intervals N derived from the same key-frame compression configuration are shown as dots with the same color in this figure, with the rightmost point for each color corresponding to the $N = 1$ case. The different colors represent different key-frame compression configurations. Once all the points are accumulated, the Pareto frontier over all these operating points is derived, shown as the solid black line in the figure. This defines the profile or the configuration settings for the five hyper-parameters for generating the variable bit rate compression with optimized rate-accuracy performance.

3.3 System Design and Complexity

We first assess the computational complexity of our approach and then discuss how it offers greater flexibility for partitioning of a visual analytics DNN work-

load while simultaneously achieving superior rate-accuracy performance. Let N denote the key frame interval, i.e. a key-frame is followed by $N - 1$ non-key frames. Denoting the computational complexity of the flow network by M_{OF} , that of the backbone network (composed of the head plus tail networks) by M_b , and that of the task network by M_t , the time-averaged computational complexity over a key frame interval (set of $N - 1$ non-key frames and a key-frame) is calculated as

$$M_{avg} = \frac{M_b}{N} \left(1 + (N - 1) * \frac{M_{OF}}{M_b} \right) + M_t \quad (2)$$

By design, $M_{OF} \ll M_b$, and significant compute efficiency gains can be obtained with increasing N through employing a lightweight optical flow network when compared to the conventional DNN approach where the compute intensive backbone network processes every incoming video frame.

As shown in Fig. 1, regardless of the DNN split point, we perform the optical flow-based feature warping at the tail network output, which is usually the layer with smallest spatial resolution. Thus, the size of the compressed optical-flow data per frame is very small compared to the size of the compressed key-frame features. This leads to significant compression efficiency gains over the image based baseline, as the key-frame interval N increases. However, errors between the predicted non-key frame features and the “true” deep features generated in the image-based pipeline tend to increase with the increasing N . Our SVC-FW approach adapts the key frame interval in addition to the DNN split point and compression bottleneck module design parameters to enhance flexibility in enabling an optimal trade-off between the compression rate, task accuracy, and computational complexity.

4 Experiments

4.1 Setup

The performance of our approach is evaluated on two video analytics tasks: 1) object detection and 2) semantic segmentation. For the detection task, the Faster R-CNN model with a ResNet-50 backbone [21] was used. Additional modifications in the conv5 layer to produce denser features with a reduced dimension of 1024 channels as described in [28] were also employed. For segmentation, the Deeplabv3 model [4] with a ResNet-50 backbone was used. For both tasks, the Resnet50 backbone was pretrained on the ImageNet-1K dataset [22]. Also, for both tasks, FlowNet2S [11] is used for the optical flow network with the input frame downsized by 2×2 and the output flow resolution is $1/8$ of the original resolution. The flow output is selected from an appropriate stage in the FlowNet2S refinement network to match the desired feature resolution at the warping point. This avoids computation of the remaining up-convolutions in the flow network and bilinear interpolation is used to obtain optical flow at the input frame resolution. The complexity of the resulting FlowNet2S implementation for the input resolutions we considered in our experiments is quite small, at around

15 GMACs, which is $\approx 10\%$ of the backbone network complexity. This is used in the average complexity calculations per Eq. (2) as shown later on in Table 2 for two different DNN split points. The flow information itself is compressed with HEVC using FFmpeg [8] and libx265, with the `medium` preset setting, no B-frames, and with `crf=22`. The flow values are scaled to the range 0-255 and encoding is performed separately on X and Y channels. The task specific implementation and experiment details are given in the next two subsections.

Video Object Detection: For training and evaluation of the entire model, images from ImageNet VID (a true video dataset) were used together with ImageNet DET. When training the FWF model in Step 1, pairs of samples were drawn from the VID dataset, one for the key frame and one for the non-key frame. Samples from DET were created by using each sample as both the key and the non-key frames. A batch size of 8 was chosen and initial learning rate was 10^{-3} with a step to 10^{-4} at 60K iterations, for a total of 90K iterations. During training, the maximum key frame interval was set to $N = 10$. Next, the SVC-FW model was trained as described in Step 2 of Sect. 3.2, with the task loss L_t as described in [21], and the performance was evaluated using the VID validation set.

Video Segmentation: For video segmentation, the Deeplabv3 model [4] was fine-tuned on the Cityscapes dataset [5] to obtain the FWF model in Step 1 from Sect. 3.2. The training samples were generated by using the one frame per clip with fine-grained annotations as the non-key frame and a frame either before or after it - up to a pre-specified maximum key-frame interval separation - as the key-frame. The maximum key-frame interval during training was set to $N = 10$. The FWF model in Sect. 3.2 was trained with a batch size of 32, for 25 epochs as described in Sect. 3.2. For the SVC-FW model the batch size was set to 48 and training was performed for 12 epochs. In both cases, the initial learning rate was set to 10^{-4} and was reduced by a factor of 10 for the last few epochs. Evaluation was performed on the Cityscapes validation set by selecting the key-frame and non-keyframe pairs in the same way as described for the training set.

Evaluation Metrics. To evaluate and compare the effectiveness of our approach, we adopt the standard practice of deriving accuracy-vs-compression curve, which is a plot of a task-specific accuracy metric against the compression rate. For object-detection, the metric used is the mean average precision (mAP), while for video segmentation it is the mean intersection over union (mIoU) metric. The compression rate is represented by bits-per-pixel (bpp), which as the name indicates is the number of bits of information that needs to be transmitted divided by the total number of pixels. In the video setting, the number of bits is the average of the bits required to encode features for key-frames and those required to encode optical flow for non-key-frames. To quantify the compression efficiency improvement, we use a modification of the widely used BD metric [2], both BD-rate (which quantifies bitrate savings at equivalent quality levels) and BD-PSNR (which quantifies quality gains at equivalent bitrates). The original BD metric is

designed using peak signal to noise ratio (PSNR), which we modify by replacing the PSNR with an analytics-appropriate task metric - mAP or mIOU. We refer to the resulting task-specific metrics as BD-mAP or BD-mIOU. For BD-rate, a negative value implies bit-rate savings; hence the more negative the value, the better. On the other hand, for BD-accuracy metric, positive values imply better quality; hence the greater the value, the better.

Baselines. The performance of our proposed approach was compared with three baselines: two modern video-compression standards – HEVC [24] and VVC [12] (Versatile Video Coding) – and a recent state-of-the-art image analytics model [6]. For HEVC compression, we used FFmpeg v6.0.1 with libx265 [8], **medium** preset setting and **crf** rate control setting to generate HEVC compressed sequences at different rate-quality points and record the average compressed frame sizes. We then processed the decompressed sequences with the appropriate DNN model (Faster R-CNN or Deeplabv3 in our two cases) to compute the task-accuracy vs. bpp curves as explained earlier.

For VVC, the inputs were compressed using **VVenC** v1.9.1 [26] with the default **medium** preset setting. The compression levels were varied with the quantization parameter (QP) and their corresponding sizes were recorded. Similar to the HEVC baseline processing, the VVC encoded files were first decoded with **VVdeC** v2.1.3 [27] before running the appropriate DNN model upon the decoded frames.

The third baseline was the image based analytics model from [6], and evaluated the performance of the frame-by-frame distributed analytics and compression. This baseline serves to show the additional compression efficiency and computational complexity gains resulting from our video based SVC-FW approach compared with the earlier image based analytic/compression solutions.

4.2 Results

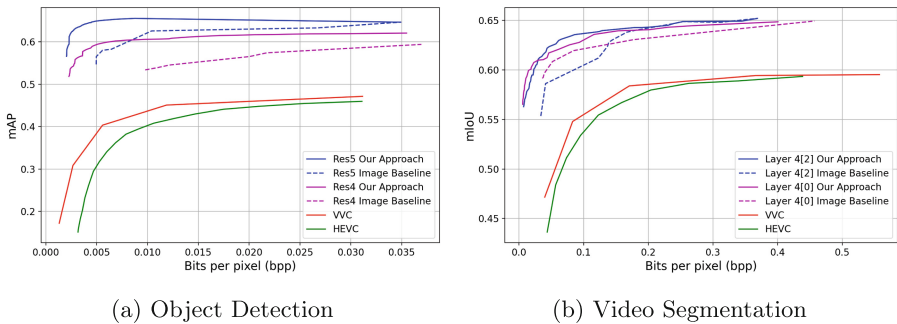


Fig. 3. Rate-Accuracy performance for the two tasks.

Comparing Forward and Bidirectional Warping. We first compare the forward warping and bi-directional warping approaches described in Sect. 3 on an example analytics task - object detection with the ResNet-50 backbone from the Detectron2 implementation split after the **res4** and **res5** blocks respectively. The results showed rate-accuracy gains for bi-directional warping over forward warping: for the **res5** split, BD-Rate = -18.81% ; and for **res4**, BD-Rate = -26.42% . We observed this superior behavior for bi-directional warping uniformly over different backbone network splits for the two different analytics tasks we studied. Hence, going forward we only report results with bi-directional warping.

Rate-Accuracy Performance. Next, we show the rate-accuracy performance of our approach for two analytic tasks and compare it with the baselines mentioned earlier for two different split points each as in Table 1. The rate-accuracy curves are shown in Fig. 3a for object detection and in Fig. 3b for segmentation. We see that our method clearly outperforms both the image baseline (frame-by-frame approach) as well as standards based compression (HEVC and VVC) for both tasks, something also seen from the improvements in BD-rate and BD-accuracy metrics shown in Table 1. It is also noteworthy that the performance of the conventional VVC and HEVC baselines is quite poor - we could not even evaluate BD-Rate between these results and our SVC-FW approach due to lack of sufficient overlap between the curves.

Table 1. BD-Rate and BD-Accuracy results showing gains for our approach vs. the image-based baseline and the VVC baseline. *See note in text.

Baseline	Object Detection						Segmentation					
	Frame-by-Frame			VVC			Frame-by-Frame			VVC		
	Split	BD rate	BD mAP	BD rate	BD mAP	Split	BD rate	BD mIoU	BD rate	BD mIoU		
res5	-72.69	0.066	-*	0.273	4[2]	-59.31	0.014	-92.59	0.131			
res4	-82.68	0.043	-*	0.175	4[0]	-40.29	0.007	-95.28	0.131			

Complexity. Table 2 shows the computational complexity for two different DNN split points for each of the two analytics tasks we considered. Following [6], the key-frame feature compression bottleneck modules are designed to incur very low added complexity in comparison with the total DNN network complexity. For this reason, the bottleneck module complexity is neglected in these calculations. The table shows reduction in total (client + edge server) complexity as well as client-only complexity as the key frame interval N increases, though there are diminishing returns with longer key-frame intervals.

Table 2. Average total and client-only computational complexity in GMACs for object detection and video segmentation tasks at different split points and key-frame intervals.

Key-Frame Interval		KF = 1		KF = 3		KF = 5		
Task	Split	layer	Total	Client	Total	Client	Total	Client
Segmentation	Split 1	4 [2]	335.56	204.67	209.19	78.30	183.92	53.03
	Split 2	4 [0]		129.79	259.11	53.34	243.82	38.05
Object Detection	Split 1	res5	179.53	126.76	104.70	77.90	89.74	36.97
	Split 2	res4		40.59	162.15	23.21	158.67	19.73

Flexibility in DNN Workload Partitioning. Neural compression approaches have thus far mostly focused on rate-distortion optimization, neglecting complexity. Recently, however, the topic of rate-distortion-complexity optimization has received greater attention [9, 20]. Our SVC-FW approach enables additional flexibility and finer-grained system operating points in navigating this space through the key-frame interval parameter choice, when compared with the image based baseline where the main design parameter is the DNN split point. An ‘operating-point’ refers to a point the rate-distortion-complexity space. To analyze the impact of key-frame interval on model performance, we proceed as follows: the key-frame interval is frozen at different settings, and the optimal compression bottleneck configurations are selected as described in Sect. 3.2. This results in an R-D performance curve for that fixed value of key-frame interval. The BD-accuracy metric for this curve relative to the VVC baseline is computed along with the complexity. The resultant complexity is divided by that of image analytics baseline [6] to get a normalized value (0.0 - 1.0).

In Fig. 4, following [9] we illustrate the operating points for the image-based baseline and for our approach with different key-frame intervals by plotting the BD-accuracy against normalized complexity (both total and client-only, using the numbers from Table 2) for the two analytics tasks. For object detection with the image-based baseline, the top left figure shows that we can obtain one operating point for each split point but the total (normalized) system complexity is unchanged at 1. With our approach, varying the key-frame interval generates additional system operating points with simultaneous improvements in BD-mAP and complexity (though diminishing and eventually worsening BD-mAP is to be expected) that better enable systems to optimize utilization of constrained resources. The top right figure shows the same BD-mAP results, except from the view of a compute-constrained client - here, the image-based baseline method offers two distinct operating points with a client complexity vs. BD-mAP trade-off, while our approach offers a much wider range of operating points to optimize the client system resource utilization. Similar results for the segmentation task are shown in the bottom row of Fig. 4.

In Fig. 5, we present another view of this for the case of object detection task, with the Faster R-CNN model’s ResNet50 backbone network split at res4

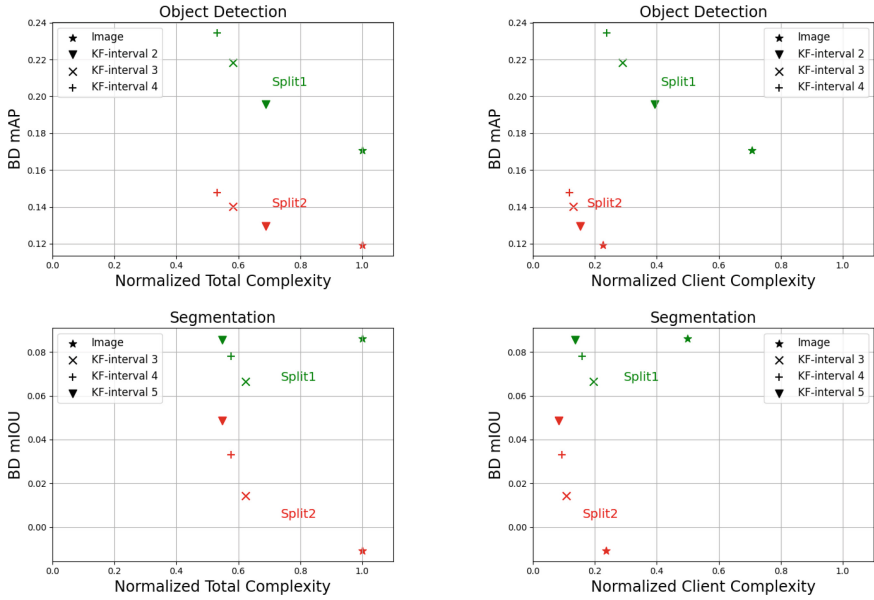


Fig. 4. Improvements in BD-mAP (vs. VVC baseline) for object-detection (top-row) and BD-mIoU for segmentation (bottom-row) vs Complexity (both total and client side) with varying key-frame intervals. Higher BD-mAP/mIoU is better. Increasing key-frame interval at moderate values not only reduces complexity, but also improves rate-accuracy performance.

layer. The lower plot for compression rate (bpp) is a section of the Pareto frontier plot from Fig. 2, with each color segment representing a different bottleneck compression level (i.e. a given key-frame compression model). The upper (complexity) plot shows that the corresponding normalized average total complexity decreases commensurately with the compression rate within each color segment representing a particular key-frame compression model. As the key-frame interval is increased, the optimal rate-accuracy performance is obtained by switching to a new key-frame compression configuration combined with a smaller key-frame interval. This plot shows the average complexity can be reduced by a factor of 5x (decrease in normalized complexity from 1.0 to 0.2) for a moderate tradeoff in accuracy.

5 Conclusions and Future Work

In this paper, we presented our split-DNN based distributed edge video analytics approach with lightweight semantics-preserving compression optimized for rate-complexity-accuracy tradeoffs and enabling flexible adaptation to dynamic edge network conditions. The results for different video analytics applications show significant compression efficiency gains with our approach compared to state of

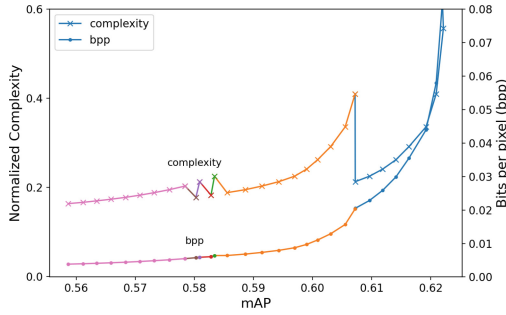


Fig. 5. Tradeoffs in object detection accuracy (mAP) vs (i) complexity (normalized w.r.t image analytics baseline [6]) in the upper plot and (ii) compression level (bpp) in the lower plot. Key-frame interval increases from right to left, and different colors represent different key-frame compression parameter settings.

the art image-based methods, as well as conventional state of the art video compression which is optimized for the human visual system. This is an area of great emerging interest, and developing approaches that apply across a wider range of different deep neural network architectures as well as other models such as transformers and multi-task models is an important area for further research. Jointly learning the compression for optical flow and the key frame features in an end-to-end manner, as well as employing learnable deep feature warping are promising areas to explore. Applying more advanced compression approaches such as arithmetic coding, improved loss functions, e.g., neural entropy estimation techniques etc. to replace the relatively simple choices we implemented here should also lead to further improvements. At the systems level, there is considerable scope to explore and develop approaches to key frame selection as well as key-frame interval adaptation in order to assure optimal end-to-end performance in dynamically varying conditions and with network errors.

References

1. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: International Conference on Learning Representations (ICLR) (2017)
2. Bjøntegaard, G.: Calculation of average PSNR differences between RD curves. Tech. rep., ITU-T Video Coding Experts Group (VCEG) (04 2001)
3. Chen, J., Ran, X.: Deep learning with edge computing: a review. In: Proceedings of the IEEE, pp. 1655–1674 (2019)
4. Chen, L.C., Papandreou, G., Shroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint [arXiv:1706.05587](https://arxiv.org/abs/1706.05587) (2017)
5. Cordts, M., et al.: The Cityscapes dataset for Semantic Urban Scene Understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
6. Datta, P., Ahuja, N., Somayazulu, V.S., Tickoo, O.: A low-complexity approach to rate-distortion optimized variable bit-rate compression for split DNN computing. In: 2022 International Conference on Pattern Recognition (ICPR) (2022)

7. Eshratifar, A., Esmaili, A., Pedram, M.: Bottlenet: a deep learning architecture for intelligent mobile cloud computing services. In: 2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), pp. 1–6. IEEE (2019)
8. FFMPEG: H.265/HEVC Video Encoding Guide
9. Gao, Y., Feng, R., Guo, Z., Chen, Z.: Exploring the rate-distortion-complexity optimization in neural image compression. arXiv preprint [arXiv:2305.07678](https://arxiv.org/abs/2305.07678) (2023)
10. Hu, Z., Xu, D., Lu, G., Jiang, W., Wang, W., Liu, S.: Fvc: an end-to-end framework towards deep video compression in feature space. *IEEE Trans. Pattern Anal. Mach. Intell.* (2023)
11. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: evolution of optical flow estimation with deep networks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017)
12. International Telecommunication Union: H.266: Verstatile video coding
13. ITU-R: ITU-R Rec. H.265 & ISO/IEC 23008-2: High efficiency video coding (2013)
14. Jayaraman, D., Grauman, K.: Slow and steady feature analysis: higher order temporal coherence in video. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
15. Kang, Y., et al.: Neurosurgeon: collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Comput. Architecture News* **45**(1), 615–629 (2017)
16. Li, J., Li, B., Lu, Y.: Deep contextual video compression. In: *35th Conference on Neural Information Processing Systems (NeurIPS)* (2021)
17. Liu, B., Chen, Y., Liu, S., Kim, H.S.: Deep learning in latent space for video prediction and compression. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2021)
18. Matsubara, Y., Callegaro, D., Baidya, S., Levorato, M., Singh, S.: Head network distillation: splitting distilled deep neural networks for resource-constrained edge computing systems. *IEEE access: practical innovations, open solutions* **8**, 212177–212193 (2020)
19. Matsubara, Y., Yang, R., Levorato, M., Mandt, S.: SC2 benchmark: supervised compression for split computing. *Trans. Mach. Learn. Res.* (accepted) (2023)
20. Minnen, D., Johnston, N.: Advancing the rate-distortion-computation frontier for neural image compression. In: *IEEE International Conference on Image Processing (ICIP)*, pp. 2940–2944. IEEE (2023)
21. Ren, S., He, K., Girshick, G., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2015)
22. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
23. Singh, S., Abu-el-Haija, S., Johnston, N., Balle, J., Shrivastava, A., Toderici, G.: End-to-end learning of compressible features. In: *IEEE International Conference on Image Processing (ICIP)* (2020)
24. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (hevc) standard. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1649–1668 (2012)
25. Torfason, R., Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Gool, L.V.: Towards image understanding from deep compression without decoding. In: *International Conference on Learning Representations (ICLR)* (2018)
26. Wieckowski, A., et al.: Vvenc: An open and optimized vvc encoder implementation. In: *Proc. IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 1–2 (2021)

27. Wieckowski, A., et al.: Towards a live software decoder implementation for the upcoming versatile video coding (vvc) codec. In: Proc. IEEE International Conference on Image Processing (ICIP), pp. 3124–3128 (2020)
28. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)



Task-Aware Local Descriptors Reconstruction Network for Few-Shot Fine-Grained Image Classification

Jianchang Tan , Xiangqian Ding , and Shusong Yu  

Department of Computer Science and Technology, Ocean University of China,
Qingdao 266100, China
tanjianchang@stu.ouc.edu.cn, {dingxq,yushusong}@ouc.edu.cn

Abstract. Few-shot fine-grained image classification involves using limited samples to classify images from novel subcategories within the same category. Recent research indicates that classifiers based on reconstruction for few-shot learning attain elevated accuracy levels due to their capacity to preserve greater detail in appearance. However, they reconstruct using a weighted sum of all local descriptors and consider the reconstruction error of all descriptors for classification. This may lead to the reconstruction and utilization of task-irrelevant descriptors for classification, potentially misleading the final outcome. This study presents, for the initial instance, a task-aware discriminative local descriptors reconstruction mechanism to address these issues, which can adaptively filter out task-irrelevant descriptors throughout the task, selecting highly discriminative descriptors for reconstruction. This design effectively aids the model in filtering out redundant information and exploring more nuanced and distinctive features throughout the task. Additionally, our unique detail-aware self-reconstruction module further refines feature discriminability. Extensive experimental results on fine-grained and generalized datasets consistently demonstrate that the proposed TARNet surpasses current state-of-the-art methods.

Keywords: Few-shot learning · Feature reconstruction · Task-aware

1 Introduction

As a particularly vital research avenue within computer vision, fine-grained image classification [5, 21, 34] boasts broad applications and has been the subject of thorough investigation. Its objective is to recognize finer sub-categories within the same general category. Specifically, the similarities between the sub-categories and the differences between instances at the intra-class level [30] make the fine-grained classification task difficult. Present deep learning techniques employ extensive annotated for effective training to tackle this challenge. However, annotating fine-grained images is prohibitively expensive [32], and image collection faces challenges posed by long-tail distributions [33, 35].

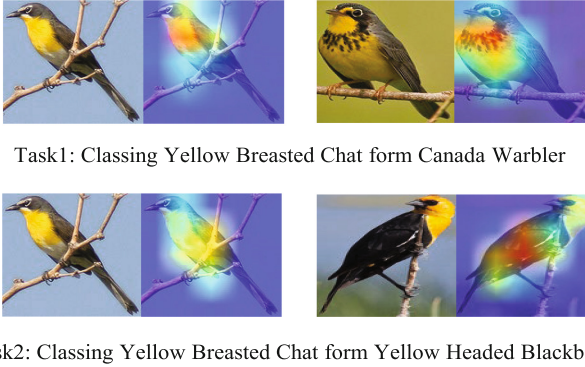


Fig. 1. Illustration of our motivation. For the same image of *Yellow Breasted Chat*, the most distinctive local parts change depending on the task.

To tackle the aforementioned problem, Few-shot approaches are extensively utilized for tasks involving fine-grained classification (FSFG) [10, 13, 36], designed to swiftly adjust to new tasks or domains using a scarce number of samples. Among them, few-shot methods based on feature reconstruction have attracted considerable interest due to their ability to preserve more spatial information and better capture shared features. FRN [23] is based on the principle that images of the same class can be better reconstructed due to their highly similar embeddings, and vice versa. Each local descriptor from the query set is reconstructed by a weighted aggregation across all spatial positions of the support features for each class, followed by classification based on the calculated reconstruction errors. BiFRN [24] introduces a novel bidirectional reconstruction mechanism, which not only employs the support set for reconstructing the query set but also utilizes the query set to reconstruct the support set. BiFRN aims to concurrently increase inter-class variations and decrease intra-class variations.

In the above few-shot methods based on reconstruction, whether reconstructing query images or the support images, local descriptors from all positions are reconstructed, including task-irrelevant descriptors (TID), such as redundant and background descriptors. Specifically, local descriptors that are effective for the current task may become distracting irrelevant descriptors in other tasks. For instance, when distinguishing between *Yellow Breasted Chat* and *Canada Warbler*, humans pay more attention to the color of the chest feathers. However, in identifying *Yellow Breasted Chat* and *Canada Warbler*, the feather color of the body parts is evidently more important, whereas the color of the chest feathers might interfere with the classification (see Fig. 1).

Inspired by the above insights, this study innovatively proposes a Task-Aware Local Descriptors Reconstruction Network (TARNet) that can selectively reconstruct highly discriminative local descriptors for the current task. To automate the identification of these discriminative local descriptors, we first introduce the task-aware discriminative local descriptor (TAD) selection module. Since highly

discriminative local descriptors will have a higher correlation with specific support classes, this module approximates the discriminative ability of each descriptor by analyzing its relationship with all support classes and selecting effective local descriptors using an adaptive threshold. We then propose the TAD Reconstruction module to specifically reconstruct the TAD and classify them by calculating the TAD reconstruction loss. In addition to the above modules, we also uncover a detail-aware self-reconstruction module that enhances the learning of fine-grained features and effectively collaborates with the prior component in identifying TAD.

In summary, the main facets of our contributions are threefold:

- We introduce a new task-aware local descriptors reconstruction network (TARNet) for FSFG.
- We propose a novel detail-aware self-reconstruction module to leverage both the spatial positional and channel information, which can aid in the semantic comprehension of the images.
- We design a new TAD reconstruction module to adaptively select highly discriminative local descriptors pertinent to the current task for reconstruction, enabling effective minimizing of the interference of TID on the classification task.

2 Related Work

2.1 Few-Shot Classification Through Metric-Based Approaches

Metric-based methods address the few-shot classification challenge by learning a distance metric, which defines a measure of similarity or dissimilarity between the support and query set. [7] presents a one-shot image recognition approach using Siamese Neural Networks, which leverages a CNN architecture to learn image embedding and trains the network to discriminate between pairs of images as either similar or dissimilar. QGN [14] inputs an additional query image to guide the Siamese Network’s learning. Prototypical Networks [16], which classify by learning prototype representations within a metric space, where the class prototype is the mean of its support set samples. Rather than employing a predetermined metric, RN [17] employs a relation module for deep distance metric learning, facilitating the comparison and classification of few-shot samples. To extend the Relation Network, Self-Attention Relation Network [4] integrates embedding, self-attention, and relation modules to enhance metric learning by effectively capturing non-local information and long-range dependencies. Some metric-based approaches yield enhanced results in classifying fine-grained images by focusing on extracting more unique features. For example, DN4 [11] employs deep local descriptors alongside an image-to-class distance for learning feature metrics. Bi-Similarity Network [12] combines two distinct similarity metrics to enable the model to learn more compact and discriminative feature representations.

The aforementioned metric-based approaches strive to learn a task-agnostic feature capable of generalizing to new categories using some specific distance metrics.

Contrary to the aforementioned approaches, we propose that the most discriminative features should vary for each task. Therefore, our model, TARNet, produces a TID mask matrix to explore the most distinctive features in the current task.

2.2 Few-Shot Classification Through Alignment-Based Approaches

In fine-grained few-shot methods, feature alignment approaches focus on spatially aligning similar objects to improve learned similarities between images. PARN [26] extracts features focused on semantic objects and overcomes the local connectivity issue of convolutional neural networks, enabling the model to compare relevant semantic features across different positions. SAML [3] employs a “collect-and-select” technique to align images with semantically related objects, concentrating on salient features. DeepEMD [31] employs the Earth Mover’s Distance as a metric to measure the structural distance between images and refine region matching. CTX [1] infers class membership by establishing a coarse spatial correspondence between query and labeled images, followed by calculating the distance between corresponding spatial features. OS2 [28] constructed an SQIE module to explore the mutual information between support and query slices. It utilized a collaborative attention mechanism to identify target co-occurring objects in the query slices based on the support slices. Unlike the aforementioned approaches that require new modules or large-scale trainable parameters, FRN [23] predicts image categories by reconstructing query feature maps within a latent space. Specifically, FRN avoids introducing a large number of learnable parameters by directly computing the regression from supporting features to query features. To address the issue of large within-class variances in fine-grained image classification, BiFRN [24, 25] incorporates a bidirectional reconfiguration mechanism, leveraging the supporting set to enhance inter-class variability and the query set to reduce intra-class variability. For the reconstruction of each local descriptor, reconstruction-based few-shot methods employ a weighted sum of local descriptors from all support samples in a class. However, in this process, TID from the support set may interfere with the reconstruction of TAD.

In contrast, in our TARNet, a novel TAD reconstruction module is devised to alleviate TID effects on TAD reconstruction, enhancing object semantic understanding.

3 Methodology

This section presents our task-aware local descriptors reconstruction network, detailing the detail-aware self-reconstruction module (Sect. 3.3), the TAD selection module (Sect. 3.4), and the TAD reconstruction and classification module (Sect. 3.5). An overview is provided in Fig. 2.

3.1 Problem Definition

In the typical setting of few-shot classification, there are three datasets: The Base dataset D_b along with its corresponding class set C_b , the validation dataset D_v with its class set C_v , and the novel dataset D_n with its class set C_n . Notably, their class spaces are distinct, with no overlapping elements. Typically, D_b and D_n are divided into individual tasks. In each task, C classes are randomly selected from these datasets. For each class, K labeled examples are included as support, and M unlabeled samples serve as query. Our goal is to learn transferable knowledge based on D_b . Then, utilizing D_v to ascertain whether the current model represents the best-performing few-shot classification model. Finally, the ultimate performance of the optimal model is usually assessed by calculating the mean accuracy across tasks sampled from D_n .

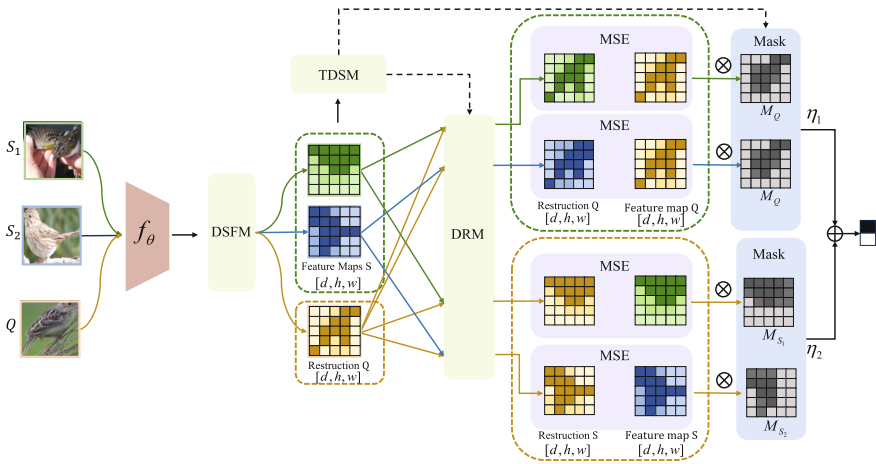


Fig. 2. The proposed task-aware local descriptors reconstruction network. DSFM refers to the detail-aware self-reconstruction module, TDSM refers to the task-aware discriminative local descriptor selection module and DRM refers to the task-aware discriminative reconstruction module.

3.2 The Framework of TARNet

Masking TID is essential in few-shot image classification because some local descriptors containing TAD are more relevant to the label than other descriptors [27]. Current reconstruction methods primarily reconstruct feature maps using all local descriptors and fail to reduce the interference of the information from TID, we propose a task-aware local descriptors reconstruction network.

Figure 2, we describe the framework of TARNet. We feed the query set Q and the support set S and to embedding module f_θ to obtain deep convolution

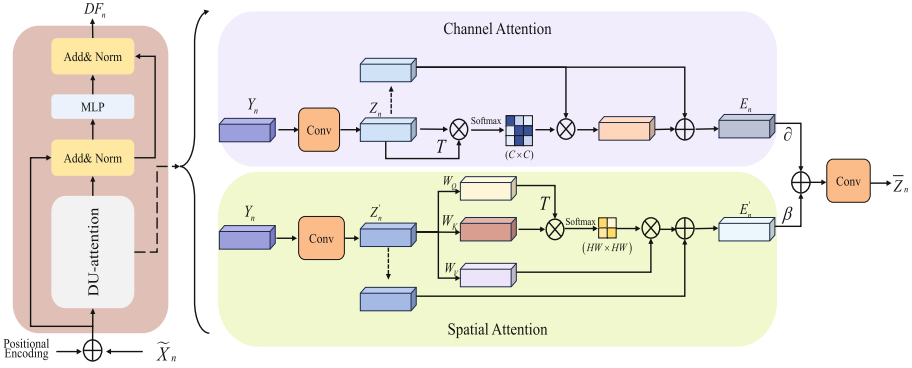


Fig. 3. Detail-aware self-reconstruction module.

image features F_q and F_s respectively. The subscripts q and s are used to represent embedding from Q and S . The detail-aware self-reconstruction module takes F_s and F_q as inputs to get detail-enhanced features DF_s and DF_q . Then, the TAD selection module takes DF_s and DF_q as inputs to obtain TID mask activation map M_s and M_q . These maps, along with DF_s and DF_q , are then inputted into the TAD reconstruction module, producing two sets of masked reconstructed support embedding and query embedding, denoted as MR_s and MR_q . The Euclidean metric distances between MR_s and support embedding are computed as the reconstruction error, as well as MR_q and query embedding. The metric score for classifying the query image is calculated by combining the two reconstruction errors through a weighted sum.

3.3 The Detail-Aware Self-reconstruction Module(DSFM)

DSFM can enhance foreground saliency and obtain more detailed information through the interplay of spatial and channel attention features to assist classification.

In a C -way K -shot classification task, we feed a given image from $C \times (K + M)$ samples, denoted as X_n , into the embedding module f_θ . This process yields a three-dimensional feature embedding array, which we then transform into an assembly of r ($r = h \times w$) d -dimensional local descriptors $\tilde{X}_n = [x_1, x_2, \dots, x_r] \in \mathbb{R}^{r \times d}$. We calculate the summation of \tilde{X}_n and the corresponding position encoding $pos \in \mathbb{R}^{r \times d}$ as the input, i.e., $Y_n = [x_1, x_2, \dots, x_r] + pos$, where pos employs sinusoidal position encoding.

We leverage the synergy between spatial and channel attention to extract more nuanced details, as illustrated in Fig. 3. For the spatial attention branch, we scale the number of channels by a convolution to λ ($0 \leq \lambda \leq 1$) for simplicity, denoted as $Z_n \in \mathbb{R}^{r \times d \times \lambda}$. Next, we input them into standard self-attention operation and generate the final spatially enhanced feature map E_n . The computational operation is depicted as follows:

$$E_n = \text{Softmax} \left(\frac{BK^\top}{\sqrt{d_k}} \right) V + Z_n \quad (1)$$

where B , K and V are obtained by Z_n multiplying W_ω^Q , W_ω^K and W_ω^V , respectively, with dimensions of $\mathbb{R}^{r \times d\lambda}$. Furthermore, W_ω^Q , W_ω^K and W_ω^V are three learnable weight parameters matrix with $d \times d$ size.

Similar to the spatial attention branch, in the channel attention branch, we also use convolution to obtain $Z'_n \in \mathbb{R}^{r \times d\lambda}$. Afterward, we feed Z'_n into a variant of self-attention to obtain the final channel-enhanced feature map E'_n .

$$E'_n = \text{Softmax} \left(\frac{(Z'_n)^\top Z'_n}{\sqrt{d_{Z'_n}}} \right) Z'_n + Z'_n \quad (2)$$

Next, we fuse E_n and E'_n with two learnable parameters α and β . To restore the channel count, we input them into a convolution layer, with the computational process described below:

$$\bar{Z}_n = \text{Conv}(\alpha E_n + \beta E'_n) \quad (3)$$

Finally, the detail-enhanced feature map DF_n is computed using Layer Normalization (LN) and Multilayer Perceptron (MLP).

$$DF_n = \text{LN}(\text{MLP}(\text{LN}(\bar{Z}_n + Y_n)) + \bar{Z}_n) \quad (4)$$

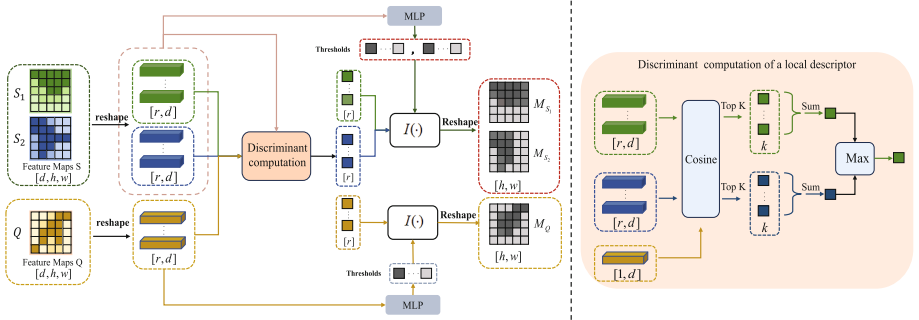


Fig. 4. Task-aware discriminative local descriptor selection module.

3.4 The TAD Selection Module (TDSM)

TDSM aims to adaptively select TAD and generate TID mask activation maps among the entire task to reduce the interference of the TID during reconstruction. The basic idea is that TAD plays a key role in classification. The complete

framework of TDSM in one image $X_n = [x_1, \dots, x_r] \in \mathbb{R}^{r \times d}$ is detailed as follows:

To reduce computing power, we compute the empirical mean of feature maps for each support class within each task, denoted as $P^c = [p_1^c, \dots, p_r^c] \in \mathbb{R}^{r \times d}$ in support class c . For each descriptor x_i , we identify the top- k closest standard support local descriptors $\mathcal{T}_{x_i} = \{\hat{p}_1^c, \dots, \hat{p}_k^c\}$ within class c by calculating their cosine distances to x_i from P^c . Following this, we ascertain the similarity of x_i to class c by summing up the cosine distances between the descriptor x_i and each \hat{p}_k^c :

$$\mathcal{D}_c^{x_i} = \sum_{\tilde{p}_k^c \in \mathcal{T}_{x_i}} \text{Sim}(x_i, \tilde{p}_k^c) \quad (5)$$

where $c \in \{1, \dots, N\}$ represents the support class and $\text{Sim}(\cdot)$ denotes cosine similarity in this work.

Then, we feed $\mathcal{D}_c^{x_i}$ to the softmax layer to normalize them and calculate the discriminative score of x_i :

$$\mathcal{R}^{x_i} = \max_c (\text{Softmax}(\mathcal{D}_c^{x_i})) \quad (6)$$

To distinguish between TAD and TID, we use a threshold \mathcal{V}_n^* to filter out local descriptors with low discriminative scores. Inspired by [2], we adopt a learnable module \mathcal{F}_ζ to generate \mathcal{V}_n^* . In this work, \mathcal{F}_ζ is implemented as MLP to adaptively predict the threshold \mathcal{V}_n^* specific to each local descriptor within the image. Formally, \mathcal{F}_ζ takes the current image descriptor set X_n as input and calculate threshold \mathcal{V}_n^* as follows:

$$\mathcal{V}_n^* = \sigma(\mathcal{F}_\zeta(X_n)) \quad (7)$$

where σ represents a sigmoid function. Next, we compute the values of the TID mask activation map $\mathcal{M}_n \in \mathbb{R}^{r \times 1}$ as follows:

$$\mathcal{M}_n = I(\mathcal{V}_n^*, \mathcal{R}^{X_n}) = \frac{1}{1 + \exp^{-\mu(\mathcal{R}^{X_n} - \mathcal{V}_n^*)}} \quad (8)$$

The above formula represents a modification of the sigmoid function. In theory, as μ reaches a sufficiently high value, the value of \mathcal{M}_n will approach one if $\mathcal{R}^{x_n} > \mathcal{V}_n^*$, otherwise, it will be near zero. This characteristic enables the mask attention map \mathcal{M}_n to effectively facilitate the filtration of TID.

Through module DSFM, we get that DF_c^s consist of $DF_{c,j}^s \in \mathbb{R}^{r \times d}$ in class c , where $j \in \{1, \dots, K\}$, and $DF_m^q \in \mathbb{R}^{r \times d}$, where $m \in \{1, \dots, C \times M\}$. Next, we feed $DF_{c,j}^s$ and DF_m^q into TDSM to produce TID mask activation maps $\mathcal{M}_{c,j}^s$ for the support set and \mathcal{M}_m^q for the query set, respectively.

3.5 The TAD Reconstruction Module (DRM)

Traditional reconstruction methods directly use all local descriptors to reconstruct feature maps, including descriptors that might be irrelevant to the task

at hand, such as background and redundant descriptors. In our work, we filter out these TID through a masking mechanism.

To filter out the TID during the reconstruction process, we adopt a cross-attention function and masking mechanism to generate masked reconstructed support features $MR_{c,m}^s \in \mathbb{R}^{kr \times d}$ in class c , where $m \in \{1, \dots, C \times M\}$ and masked reconstructed query features $MR_{c,m}^q \in \mathbb{R}^{r \times d}$, where $m \in \{1, \dots, C \times M\}$. Initially, DF_c^s , DF_m^q , and DF_m^q are each multiplied by W_ϕ^Q , W_ϕ^K , and W_ϕ^V , respectively, obtaining S_c^Q , Q_m^K and Q_m^V , where W_ϕ^Q , W_ϕ^K and $W_\phi^V \in \mathbb{R}^{d \times d}$. Similarly, DF_m^q , DF_c^s , and DF_c^s are each multiplied by W_ϕ^Q , W_ϕ^K and W_ϕ^V , respectively, obtaining Q_m^Q , S_c^K , S_c^V .

Next, We determine $MR_{c,m}^q$ using the support embedding S_c^V in the c -th class and calculate $MR_{c,m}^s$ in the c -th class from the m -th query embedding Q_m^V , employing the following two equations.

$$MR_{c,m}^s = MAtt(S_c^Q, Q_m^K, Q_m^V) = \left(\text{Softmax} \left(\frac{S_c^Q (Q_m^K)^\top}{\sqrt{d_{Q_m^K}}} \right) \cdot \text{re}_q(\mathcal{M}_m^q) \right) Q_m^V \quad (9)$$

$$MR_{c,m}^q = MAtt(Q_m^Q, S_c^K, S_c^V) = \left(\text{Softmax} \left(\frac{Q_m^Q (S_c^K)^\top}{\sqrt{d_{S_c^K}}} \right) \cdot \text{re}_s(\mathcal{M}_c^s) \right) S_c^V \quad (10)$$

where Symbol “ \cdot ” denotes element-wise multiplication. $\text{re}_s(\cdot)$ refers to the operation of copying \mathcal{M}_c^s r times to generate $RM_c^s \in \mathbb{R}^{kr \times r}$. Likewise, $\text{re}_q(\cdot)$ refers to the operation of copying \mathcal{M}_m^q kr times to generate $RM_m^q \in \mathbb{R}^{r \times kr}$.

After DRM, we compute the reconstruction error between original feature maps and reconstructed feature maps by using the Euclidean metric. Since the label is only related to the foreground, the error of the TID must also be filtered out. The final reconstruction error is derived from the weighted sum of the discrepancies between local descriptors of the original images and their corresponding reconstructed images. The computing process is shown as follows:

$$e_{c,m} = \tau(\eta_1(\|Q_m^V - MR_{c,m}^q\| \times \mathcal{M}_m^q) + \eta_2(\|S_c^V - MR_{c,m}^s\| \times \mathcal{M}_c^s)) \quad (11)$$

where η_1 and η_2 are learnable weight parameters associated with each reconstruction error, respectively. Symbol τ is a learnable temperature factor. Symbol “ \times ” is spatial-wise multiplication. We then normalize $e_{c,m}$ to get $\hat{e}_{c,m}$ as follows:

$$\hat{e}_{c,m} = \frac{\exp(e_{c,m})}{\sum_{i=1}^C \exp(e_{c,m})} \quad (12)$$

In one C-way K-shot task, the total loss is computed using cross-entropy loss as follows:

$$\mathcal{L} = -\frac{1}{C \times M} \sum_{m=1}^{C \times M} \sum_{c=1}^C y_{c,m} \log(\hat{e}_{c,m}) \quad (13)$$

where $y_{c,m}$ equals 1 when c and l_m are equal, otherwise 0, and l_m is the label of the query image Q_m .

4 Experimental Results and Analysis

4.1 Experimental Setup

Dataset. Fine-grained datasets: CUB [20] contains 200 bird species with a total of 6033 images. Dogs [6] includes 20 dog breeds, totaling 20580 images. Cars [8] comprises 196 vehicle classes with 16185 images. meta-iNat [18, 22] is a wildlife species benchmark with 1135 categories, each having between 50 to 1000 images. tiered meta-iNat [22] is a more challenging version of meta-iNat, where its 354 test categories do not overlap with the 781 training categories. Coarse-grained datasets: mini-ImageNet [19] is a subset of ImageNet consisting of 100 classes, each with 600 images. tiered-ImageNet [15] is a larger subset of ImageNet with 351-97-160 categories for training, validation, and testing, respectively.

We divided each dataset into training, validation, and test sets. The proportions of D_{train} , D_{val} , and D_{test} are the same as in references [22, 24] and all images were resized to 84×84 .

Implementation Details. Our study conducted experiments using two widely adopted backbone architectures: Conv-4 and Resnet-12. The design of these architectures is entirely consistent with the designs in [23, 29]. During the training phase, both Conv-4 and ResNet-12 models are trained using SGD with Nesterov momentum of 0.9 for 1200 epochs. The initial learning rate is set to 0.1, with a weight decay of $5e-4$, and the learning rate is reduced by a factor of 10 every 400 epochs. For Conv-4, we use 30-way 5-shot episodes for training and 30-way 5-shot episodes for testing. The nearest neighbor k is set to 2. Specifically, for ResNet-12, due to memory constraints, we use 15-way 5-shot episodes for its training. The nearest neighbor k is set to 5. In both settings, we use 15 query images per class. We adopt standard data augmentation techniques, including random horizontal flip, color jitter, and center crop, to enhance training stability. During testing, we randomly create 10,000 episodes in D_v to compute the final results, ensuring reliability with a 95% confidence interval.

Table 1. Comparison to prior works on meta-iNat and tiered meta-iNat with Conv-4 backbones. All 95% confidence intervals are below 0.25. The highest results are highlighted in **bold font**

Method	meta-iNat		tiered meta-iNat	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
ProtoNet [16]	55.34	76.43	34.34	57.13
CTX [1]	60.03	78.80	36.83	60.84
FRN [23]	62.42	80.45	43.91	63.36
BiFRN [24]	65.85	83.28	47.56	67.55
Ours	66.63	84.61	48.31	69.75

4.2 Results

Fine-Grained Few-Shot Classification. To ascertain the efficacy of our approach in FSFG, we benchmark against established few-shot image classification techniques (PARN [26], DeepEMD [31], CTX [1], FRN [23], TDM [9], BiFRN [24]). The datasets utilized by these methods are also employed in the research presented in this work.

Table 2. Comparison to prior works on CUB, Stanford Cars and Dogs. Average accuracy(in %) is reported. Results with Conv-4 backbone appear in the top block and ResNet-12 in the bottom. The highest results are highlighted in **bold font**

Method	CUB		Dogs		Cars	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
ProtoNet [16]	64.82 ± 0.23	85.74 ± 0.14	46.66 ± 0.21	70.77 ± 0.16	50.88 ± 0.23	74.89 ± 0.18
Relation [17]	63.94 ± 0.92	77.87 ± 0.64	47.35 ± 0.88	66.20 ± 0.74	46.04 ± 0.91	68.52 ± 0.78
DN4 [11]	57.45 ± 0.89	84.41 ± 0.58	39.08 ± 0.76	69.81 ± 0.69	34.12 ± 0.68	87.47 ± 0.47
PARN [26]	74.43 ± 0.95	83.11 ± 0.67	55.86 ± 0.97	68.06 ± 0.72	66.01 ± 0.94	73.74 ± 0.70
DeepEMD [31]	64.08 ± 0.50	80.55 ± 0.71	46.73 ± 0.49	65.74 ± 0.63	61.63 ± 0.27	72.95 ± 0.38
BSNet [12]	62.84 ± 0.95	85.39 ± 0.56	43.42 ± 0.86	71.90 ± 0.68	40.89 ± 0.77	86.88 ± 0.50
CTX [1]	72.61 ± 0.21	86.23 ± 0.14	57.86 ± 0.21	73.59 ± 0.16	66.35 ± 0.21	82.25 ± 0.14
FRN [23]	74.90 ± 0.21	89.39 ± 0.12	60.41 ± 0.21	79.26 ± 0.15	67.48 ± 0.22	87.97 ± 0.11
TDM [9]	72.01 ± 0.22	89.05 ± 0.12	51.57 ± 0.23	75.25 ± 0.16	65.67 ± 0.22	86.44 ± 0.12
BiFRN [24]	79.08 ± 0.20	92.22 ± 0.10	65.23 ± 0.22	81.87 ± 0.14	76.32 ± 0.20	92.36 ± 0.11
Ours	82.37 ± 0.19	93.58 ± 0.10	68.60 ± 0.22	83.82 ± 0.13	80.86 ± 0.18	94.67 ± 0.07
ProtoNet [16]	81.02 ± 0.20	91.93 ± 0.11	73.81 ± 0.21	87.39 ± 0.12	85.46 ± 0.19	95.08 ± 0.08
CTX [1]	80.39 ± 0.20	91.01 ± 0.11	73.22 ± 0.22	85.90 ± 0.13	85.03 ± 0.19	92.63 ± 0.11
DeepEMD [31]	75.59 ± 0.30	88.23 ± 0.18	70.38 ± 0.30	85.24 ± 0.18	80.62 ± 0.26	92.63 ± 0.13
FRN [23]	84.30 ± 0.18	93.34 ± 0.10	76.76 ± 0.21	88.74 ± 0.12	88.01 ± 0.17	95.75 ± 0.07
TDM [9]	85.15 ± 0.18	93.99 ± 0.09	78.02 ± 0.20	89.85 ± 0.11	88.92 ± 0.16	96.88 ± 0.06
BiFRN [24]	85.44 ± 0.18	94.73 ± 0.09	77.19 ± 0.21	88.34 ± 0.12	90.20 ± 0.15	97.60 ± 0.05
Ours	86.15 ± 0.17	94.91 ± 0.08	78.11 ± 0.20	89.96 ± 0.11	89.98 ± 0.15	97.26 ± 0.06

Tables 1 and 2 showcase the classification results for 5-way few-shot tasks on fine-grained datasets. Our TARNet achieves the highest accuracy across all five fine-grained datasets when utilizing the Conv-4 architecture. This underscores the efficiency and superiority of TARNet. Meanwhile, TARNet is 0.22% and 0.34% lower than BiFRN on the 5-way 1-shot and 5-shot tasks of Cars when the ResNet-12 is adopted, and outperforms BiFRN by 0.18% to 1.62% in other settings. This indicates that TARNet consistently delivers competitive performance across various datasets and experimental configurations.

Table 3. Comparison to prior on mini-ImageNet and tiered-ImageNet.

Method	Backbone	mini-ImageNet		tiered-ImageNet	
		5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
ProtoNet [16]	Conv4	48.98 \pm 0.20	70.92 \pm 0.16	49.58 \pm 0.22	70.31 \pm 0.18
BiFRN [24]		54.00 \pm 0.20	71.79 \pm 0.16	57.74 \pm 0.22	76.28 \pm 0.17
Ours		54.79 \pm 0.20	73.48 \pm 0.15	58.60 \pm 0.22	77.82 \pm 0.17
ProtoNet [16]	Resnet12	58.80 \pm 0.20	76.68 \pm 0.15	64.77 \pm 0.23	81.58 \pm 0.17
BiFRN [24]		59.29 \pm 0.20	76.00 \pm 0.15	65.71 \pm 0.23	80.80 \pm 0.18
Ours		60.05 \pm 0.20	76.12 \pm 0.16	66.93 \pm 0.23	81.08 \pm 0.18

General Few-Shot Classification. We also conducted further evaluations of TARNet’s performance on the mini-ImageNet and tiered-ImageNet datasets. From Table 3, it can be seen that our method achieves the best or most competitive performance on the 5-way 1-shot and 5-shot tasks of mini-ImageNet and tiered-ImageNet, indicating that our method has broad applicability.

In summary, the above results demonstrate that TARNet is effective in few-shot classification tasks on both fine-grained and coarse-grained datasets.

4.3 Ablation Study

This section first examines the impact of DSFM and DRM on TARNet, as well as the influence of spatial and channel attention on TDSM. Next, we examine the influence of feature map size on the classification performance of TARNet on the CUB and Cars datasets under the Conv-4 backbone. Finally, we visualized the reconstruction errors for the 5-way, 5-shot classification tasks.

Table 4. Ablation studies using only DSFM module or DRM module.

DSFM	DRM	Backbone	CUB		Dogs		Cars	
			5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
\times	\times	conv4	74.95	90.17	61.56	80.25	71.41	89.03
\checkmark	\times		80.83	92.45	67.05	81.67	79.19	92.88
\times	\checkmark		78.55	92.71	62.09	81.09	77.81	93.70
\checkmark	\checkmark		82.37	93.58	68.60	83.82	80.86	94.67
\times	\times	Resnet12	85.02	94.50	76.63	89.16	89.14	97.15
\checkmark	\times		86.03	94.65	77.74	89.35	89.81	97.30
\times	\checkmark		85.37	94.53	76.29	87.74	89.64	97.24
\checkmark	\checkmark		86.15	94.91	78.11	89.56	89.98	97.26

The Impact of the DSFM and DRM. To assess the efficacy of our method, we systematically strip down its components. Specifically, we perform experiments wherein we remove the DSFM module (denoted as DRM), then the DRM module (denoted as DSFM), and finally, both. The results, presented in Table 4, clearly demonstrate enhanced performance when both DSFM and DRM modules are employed concurrently (DSFM+DRM). This indicates that the DSFM and DRM modules are not only essential but also complement each other to improve the overall functionality.

Table 5. Ablation of channel and spatial attention in the DSTM module.

Channel	Spatial	Backborn	CUB		Dogs		Cars	
			5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
\times	\times	Conv4	78.55	92.71	62.09	81.09	77.81	93.70
\checkmark	\times		81.95	93.35	66.00	83.33	81.15	94.66
\times	\checkmark		82.17	93.08	67.72	84.20	80.96	94.81
\checkmark	\checkmark		82.37	93.58	68.60	83.82	80.86	94.67
\times	\times	Resnet12	85.37	94.53	76.29	87.74	89.64	97.24
\checkmark	\times		86.40	94.84	77.31	88.95	90.07	97.01
\times	\checkmark		85.33	94.41	78.32	89.44	89.01	96.99
\checkmark	\checkmark		86.15	94.91	78.11	89.56	89.98	97.26

The Impact of the Channel Attention and Spatial Attention in DSFM Module. To verify the effectiveness of channel attention and spatial attention in the DSFM module, we conducted ablation experiments targeting these two components. The results are shown in Table 5. The results show that using either channel attention or spatial attention alone improves classification performance to some extent, indicating that both attention mechanisms are effective. However, in most cases, the combination of both achieves optimal or near-optimal results. Therefore, we generally choose the DSFM module that integrates both channel and spatial attention.

Table 6. Ablation on feature map size of TARNet.

Feature Map Size	Methods	CUB		Cars	
		5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
5×5	BiFRN	73.61 ± 0.21	89.40 ± 0.12	70.01 ± 0.21	86.13 ± 0.12
	Ours	75.87 ± 0.21	90.46 ± 0.12	73.79 ± 0.21	88.78 ± 0.11
10×10	BiFRN	74.17 ± 0.21	90.52 ± 0.12	70.98 ± 0.21	88.63 ± 0.11
	Ours	76.64 ± 0.21	91.73 ± 0.12	75.90 ± 0.20	91.71 ± 0.09

The Impact of Feature Map Size. Table 1 demonstrates that Conv4 yields more pronounced improvements than Resnet12. We believe this is because the feature maps generated by Resnet12 have a larger receptive field, which may filter out some important foreground information along with TID. To test this, we eliminated the pooling layer from the last convolutional layer in Conv4, enlarging the feature map from 5×5 to 10×10 , thus reducing the receptive field. Table 6 shows that the classification accuracy of BiFRN increased by an average of 1.29% when using 10×10 feature maps compared to 5×5 maps, while the TARNet model improved by 1.77%. This suggests that the smaller the receptive field, the higher the improvement in the performance of TARNet in the same backbone.

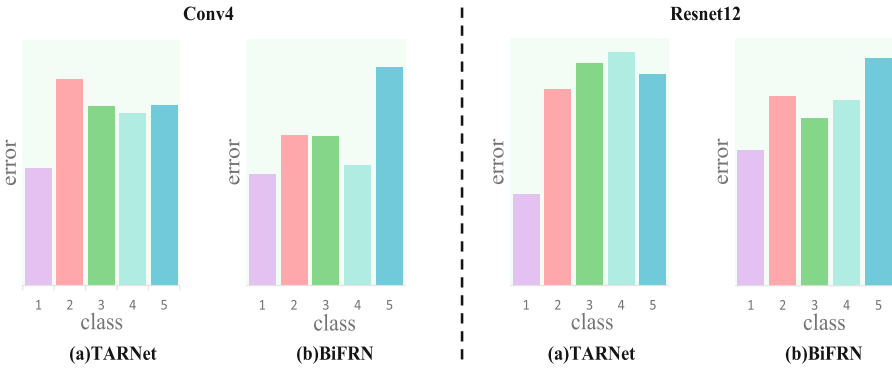


Fig. 5. Reconstruction errors predicted by BiFRN and our TARNet under the 5-way 5-shot setting on the CUB-200 dataset. In every bar chart, the vertical axis represents the five classes, while the horizontal axis reflects the reconstruction error.

The Quality of the Selected Query Descriptors in TARNet. We visualized the reconstructed errors generated by BiFRN and our TARNet under the Conv-4 and ResNet-12 backbones on a 5-way 5-shot classification task from CUB-200. Specifically, We designate the query as belonging to the first category and calculate the reconstruction errors between the query and the five classes. The results of the visualization are shown in Fig. 5. It is evident that both models accurately classified the query, as the reconstruction errors computed by both models of the first class were the smallest among all settings. However, Our model yields a greater disparity in the reconstruction error between the confused and target classes. This indicates that TARNet has found more discriminative features and only reconstructed them.

4.4 Model Complexity Analysis

Table 7. Comparison of the efficiency of some publicly available few-shot methods.

Method	Backborn	Params. (M)	FLOPs (G)	Time (S)
ProtoNet [16]	Conv4	0.113	0.601	0.004
BiFRN [24]		0.152	0.608	0.034
Ours		0.211	0.616	0.081
ProtoNet [16]	Resnet12	12.424	21.161	0.044
BiFRN [24]		16.132	21.726	0.258
Ours		21.847	22.435	0.374

To comprehensively evaluate performance, we compared the efficiency of TARNet with several publicly available few-shot learning methods. The results of model parameters (Params.), floating-point operations (FLOPs), and inference time (Times) are shown in Table 7. From the comparisons in Table 7 and Table 1 to Table 3, it can be seen that although TARNet slightly increases the storage requirements, it still achieves better few-shot fine-grained image classification performance while maintaining competitive computational and inference efficiency. Especially when using Conv4 as the backbone, where a slight increase in computational cost and storage leads to significant performance improvements, we consider this investment worthwhile.

5 Conclusion

In this work, we proposed a task-aware local descriptors reconstruction network for FSFG. Our primary innovation is a task-aware discriminative local descriptors reconstruction module, i.e. We eliminate background and redundant local descriptors, reconstructing only task-aware discriminative local descriptors. Relative to current methods based on reconstruction, our proposed method can better focus on the most crucial details of the current task, disregarding distractions or irrelevant parts. Rigorous testing indicates that our network consistently achieves strong performance across three fine-grained image datasets, often rivaling or even outperforming the existing state-of-the-art methods.

Acknowledgements. This work was supported by the Science and Technology Project of Qingdao (No.23-2-8-smjk-20-nsh).

References

1. Doersch, C., Gupta, A., Zisserman, A.: Crosstransformers: spatially-aware few-shot transfer. *Adv. Neural. Inf. Process. Syst.* **33**, 21981–21993 (2020)
2. Dong, C., Li, W., Huo, J., et al.: Learning task-aware local representations for few-shot learning. In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 716–722 (2021)
3. Hao, F., He, F., Cheng, J., et al.: Collect and select: semantic alignment metric learning for few-shot learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8460–8469 (2019)
4. Hui, B., Zhu, P., Hu, Q., Wang, Q.: Self-attention relation network for few-shot learning. In: *2019 IEEE international conference on Multimedia & Expo Workshops (ICMEW)*, pp. 198–203. *IEEE* (2019)
5. Ke, X., Cai, Y., Chen, B., et al.: Granularity-aware distillation and structure modeling region proposal network for fine-grained image classification. *Pattern Recogn.* **137**, 109305 (2023)
6. Khosla, A., Jayadevaprakash, N., Yao, B., et al.: Novel dataset for fine-grained image categorization: Stanford dogs. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop on Fine-grained Visual Categorization (FGVC)*, vol. 2. Citeseer (2011)
7. Koch, G., Zemel, R., Salakhutdinov, R., et al.: Siamese neural networks for one-shot image recognition. In: *International Conference on Machine Learning Deep Learning Workshop*, vol. 2. Lille (2015)
8. Krause, J., Stark, M., Deng, J., et al.: 3d object representations for fine-grained categorization. In: *Proceedings of the IEEE International Conference on Computer Vision workshops*, pp. 554–561 (2013)
9. Lee, S., Moon, W., Heo, J.P.: Task discrepancy maximization for fine-grained few-shot classification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5331–5340 (2022)
10. Li, P., Zhao, G., Xu, X.: Coarse-to-fine few-shot classification with deep metric learning. *Inf. Sci.* **610**, 592–604 (2022)
11. Li, W., Wang, L., Xu, J., et al.: Revisiting local descriptor based image-to-class measure for few-shot learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7260–7268 (2019)
12. Li, X., Wu, J., Sun, Z., et al.: Bsnet: Bi-similarity network for few-shot fine-grained image classification. *IEEE Trans. Image Process.* **30**, 1318–1331 (2020)
13. Li, X., Yang, X., Ma, Z., et al.: Deep metric learning for few-shot image classification: a review of recent developments. *Pattern Recogn.*, 109381 (2023)
14. Munjal, B., Flaborea, A., et al.: Query-guided networks for few-shot fine-grained classification and person search. *Pattern Recogn.* **133**, 109049 (2023)
15. Ren, M., Triantafyllou, E., Ravi, S., et al.: Meta-learning for semi-supervised few-shot classification. *arXiv preprint [arXiv:1803.00676](https://arxiv.org/abs/1803.00676)* (2018)
16. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. *Adv. Neural Inform. Process. Syst.* **30** (2017)
17. Sung, F., Yang, Y., Zhang, L., et al.: Learning to compare: relation network for few-shot learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208 (2018)
18. Van Horn, G., Mac Aodha, O., Song, Y., et al.: The inaturalist species classification and detection dataset. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 8769–8778 (2018)

19. Vinyals, O., Blundell, C., Lillicrap, T., et al.: Matching networks for one shot learning. *Adv. Neural Inform. Process. Syst.* **29** (2016)
20. Wah, C., Branson, S., Welinder, P., et al.: The caltech-ucsd birds-200-2011 dataset (2011)
21. Wei, X.S., Song, Y.Z., Mac Aodha, O., et al.: Fine-grained image analysis with deep learning: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(12), 8927–8948 (2021)
22. Wertheimer, D., Hariharan, B.: Few-shot learning with localization in realistic settings. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6558–6567 (2019)
23. Wertheimer, D., Tang, L., Hariharan, B.: Few-shot classification with feature map reconstruction networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8012–8021 (2021)
24. Wu, J., Chang, D., Sain, A., et al.: Bi-directional feature reconstruction network for fine-grained few-shot image classification. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 2821–2829 (2023)
25. Wu, J., Chang, D., Sain, A., et al.: Bi-directional ensemble feature reconstruction network for few-shot fine-grained classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1–16 (2024)
26. Wu, Z., Li, Y., Guo, L., Jia, K.: Parn: position-aware relation networks for few-shot learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6659–6667 (2019)
27. Yan, L., Li, F., Zheng, X., et al.: Few-shot learning via task-aware discriminant local descriptors network. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 2887–2894 (2023)
28. Yang, Y., Wang, B., Zhang, D., et al.: Self-supervised interactive embedding for one-shot organ segmentation. *IEEE Trans. Biomed. Eng.* (2023)
29. Ye, H.J., Hu, H., Zhan, D.C., et al.: Few-shot learning via embedding adaptation with set-to-set functions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8808–8817 (2020)
30. Zha, Z., Tang, H., Sun, Y., et al.: Boosting few-shot fine-grained recognition with background suppression and foreground alignment. *IEEE Trans. Circ. Syst. Video Technol.* (2023)
31. Zhang, C., Cai, Y., Lin, G., et al.: Deepemd: few-shot image classification with differentiable earth mover’s distance and structured classifiers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12203–12213 (2020)
32. Zhang, C., Yao, Y., Xu, X., et al.: Extracting useful knowledge from noisy web images via data purification for fine-grained recognition. In: *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 4063–4072 (2021)
33. Zhang, S., Li, Z., Yan, S., et al.: Distribution alignment: a unified framework for long-tail visual recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2361–2370 (2021)
34. Zhao, P., Li, Y., Tang, B., et al.: Feature relocation network for fine-grained image classification. *Neural Netw.* **161**, 306–317 (2023)
35. Zhou, Y., Hu, Q., Wang, Y.: Deep super-class learning for long-tail distributed image classification. *Pattern Recogn.* **80**, 118–128 (2018)
36. Zhu, Y., Liu, C., Jiang, S.: Multi-attention meta learning for few-shot fine-grained image recognition. In: *Proceedings of the Conference on International Joint Conferences on Artificial Intelligence*, pp. 1090–1096 (2020)



TRIGS: Trojan Identification from Gradient-Based Signatures

Mohamed Hussein¹✉, Sudharshan Subramaniam Janakiraman¹,
and Wael AbdAlmageed²

¹ Information Sciences Institute, University of Southern California, Arlington,
VA 22203, USA

mehussein@isi.edu, sudharshan.sj@seekout.com

² Electrical and Computer Engineering Department, Clemson University, Riggs Hall,
Clemson, SC 29634, USA

wabdalm@clemson.edu

Abstract. Training machine learning models can be very expensive or even unaffordable. This may be, for example, due to data limitations, or computational power limitations. Therefore, it is a common practice to rely on open-source pre-trained models whenever possible. However, this practice is alarming from a security perspective. Pre-trained models can be infected with Trojan attacks, in which the attacker embeds a trigger in the model such that the model's behavior can be controlled by the attacker when the trigger is present in the input. In this paper, we present a novel method for detecting Trojan models. Our method creates a signature for a model based on activation optimization. A classifier is then trained to detect a Trojan model given its signature. We call our method TRIGS for TRoJan Identification from Gradient-based Signatures. TRIGS achieves state-of-the-art performance on two public datasets of convolutional models. Additionally, we introduce a new challenging dataset of ImageNet models based on the vision transformer architecture. TRIGS delivers the best performance on the new dataset, surpassing the baseline methods by a large margin. Our experiments also show that TRIGS requires only a small amount of clean samples to achieve good performance, and works reasonably well even if the defender does not have prior knowledge about the attacker's model architecture. Our data (<https://github.com/vimal-isi-edu/tat>) and code (<https://github.com/vimal-isi-edu/trigs>) are publicly available.

Keywords: Poisoning · Backdoor Attacks · Trojan Models · Defense Methods

1 Introduction

Machine learning has made great progress since the introduction of deep learning. However, the training of deep models remains more of an art than science. It

S. S. Janakiraman—Contributed to this research during his time at USC.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15303, pp. 356–371, 2025.
https://doi.org/10.1007/978-3-031-78122-3_23

requires a lot of trial and error and parameter fine-tuning. All this incurs a significant computational cost and energy footprint. More importantly, high-performing models are trained on huge amounts of data, a process that can only be afforded by a few organizations. As a result, researchers and practitioners use open-source pre-trained models when they are available.

Despite the ubiquity of using open-source pre-trained models, this practice poses a security threat. Delegating the training process to a third party allows the training party to embed a trigger pattern in the training data. In such a case, the trained model behaves normally in the absence of the trigger but can produce a certain output, determined by the attacker, when the trigger is present. This is known as Trojan or backdoor attacks on machine learning models.

Trojan attacks are hard to detect in a trained model because the model behaves normally on benign inputs. Without knowledge of the trigger, it is impossible to reproduce the model’s malicious behavior. Consequently, many proposed methods for Trojan model detection employ reverse engineering to reconstruct possible triggers used to train a given model. The candidate triggers are usually then filtered using heuristics about the trigger size [3], norm [31], or the resulting attack success rate [15]. The reverse engineering process can be time-consuming, especially, if it involves attempting all possible combinations of source and target classes for trigger reconstruction [27]. Furthermore, the deployed heuristics for anomaly detection are susceptible to detecting a trigger when none exists [23].

In this paper, we introduce a novel method for the detection of Trojan models. Our method does not attempt to reconstruct the trigger, nor does it apply heuristics about the nature of the trigger. Instead, we use a purely data-driven approach to detect the presence of a trigger from its fingerprint in the model’s signature. The main ingredient of our method is the construction of such a signature for a model, which is accomplished using an activation optimization process that results in a fixed number of activation maps for a given classification model. The signature can be further reduced in size via a feature extraction step that uses pixel-wise statistics. A classifier is then used to detect whether a model is Trojan or not based on the signature or its features. We call our method TROjan Identification from Gradient-based Signatures (TRIGS). The process is illustrated in Fig. 1. TRIGS is agnostic to the nature of the probe models’ architecture. In fact, it works well on very different architectures, as we shall discuss later.

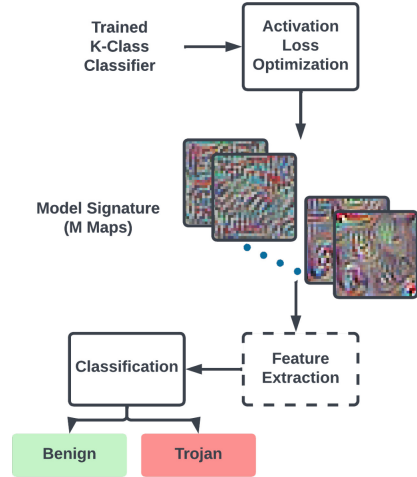


Fig. 1. TRIGS Framework.

Most of the proposed methods for Trojan model detection in the literature are evaluated on non-public model sets of vastly varying sizes. The few publicly available datasets for image-classification models are limited in the number of classes they support. Also, the vast majority of the model architectures are convolutional. In this paper, we introduce a new dataset of vision transformer (ViT) models [5] trained on ImageNet. Our dataset will be the largest public dataset in terms of the number of classes (1000) supported by its classification models. It is also the only dataset that focuses on the ViT architecture, which has recently become a popular backbone for many computer vision tasks [1, 32]. On our collected data and two public datasets, TRIGS delivers state-of-the-art performance.

2 Related Work

2.1 Activation Maximization

Activation maximization, also known as model inversion or feature visualization, was first introduced in [6] to visualize the internal nodes of a neural network. The method employed gradient descent with L2 regularization to visualize internal units of Stacked Denoising Auto-encoders and Deep Belief Networks. In [24], the same technique was applied to convolutional networks. The authors also showed that this gradient-based approach is a generalization of the deconvolution-based approach in [35], which was proposed for the same purpose. In [34], Gaussian blur and pixel clipping were added as additional regularization techniques to produce smoother visualizations. Alternatively to Gaussian blur, in [18], random jittering and minimization of the total variation were introduced as extra regularization techniques. More feature visualization techniques are discussed in [20].

2.2 Trojan Attacks

Trojan attacks on deep learning models were first introduced by [9], in which mislabeled examples stamped with a trigger were used to train a Trojan model. In [16], a method was presented for creating Trojan attacks without access to training data by using model inversion [19]. In [8, 22], methods for clean label poisoning attacks were introduced. These methods target the misclassification of a specific test example. Interestingly, in [21, 26], clean label attacks were carried out in such a way that a trigger can be used in the testing phase while being completely hidden during training.

2.3 Defenses Against Trojan Attacks

Defenses against Trojan attacks include the detection of poisoned samples in a training dataset [2] and making model training robust against poisoned samples

[30]. In both types of defenses, it is assumed that the defender has control over the training process. Other types of defenses include modifying a known Trojan model to bypass the trigger [14] and detecting if a trained model is Trojan or not [13]. Our focus in this section is on the latter type of defense, which is the topic of this paper. We argue that Trojan model detection is an indispensable capability because it is the first step towards removing the effect of the trigger if it is present.

DeepInspect [3] is an algorithm for detecting models with backdoors assuming that the defender has access only to the trained model and no access to clean data samples. To achieve this goal, the method uses model inversion [7] to construct a training dataset for the model. Using the constructed training data, a generator is employed to create perturbation patterns (triggers) such that the model produces a given target class with the poisoned samples. Then, anomaly detection is applied to determine if any of the generated triggers is a real trigger used to train the model. Similarly, in [27, 31], anomaly detection is applied to detect the real trigger (if any) among a set of generated triggers. The idea was extended in [4] to the black-box case, where the model is only accessible through its query responses. However, in this case, the triggers are generated by reverse engineering with real clean samples. In [15, 23], potentially compromised neurons are first identified. Based on the identified compromised neurons, possible triggers are generated and only those that consistently subvert the model’s predictions to a certain target class are admitted. The methods require at least one clean sample of each class. The case when the clean samples available to the defender are limited or non-existent was handled in [29]. The method uses the similarity between two embeddings per image, one with a universal perturbation pattern and one with a local perturbation pattern, as an indicator of the presence of a Trojan. Similar to [33], Universal Litmus Patterns (ULPs) [13] were introduced as probes to a classification model, the output of which can distinguish benign from Trojan models. Complex attack scenarios, in which the trigger pattern is not limited to be patch-shaped, were the focus of [17]. More recently [28], a detection method was introduced based on the observation that Trojan models have an anomalously large logit margin for the target class. Our proposed method, TRIGs, works both in the white-box and black-box settings and with a limited access to clean data. TRIGs also works well with both CNN and ViT architectures. The closest defense to TRIGs is the One-Pixel Signature (OPS) defense [11], in which a model signature is used to train a binary classifier to distinguish Trojan from benign models. However, to work in the black-box scenario, OPS uses brute force search to construct the signatures instead of using gradient descent optimization as in TRIGs. Also, the signature size in OPS is proportional to the number of classes, which can be very large, while TRIGs can leverage pixel statistics to significantly reduce the signature size regardless of the number of classes.

2.4 Datasets for Trojan Attack Defense

Most of the work done on Trojan attack defenses used private datasets, usually containing a small number of models. To our knowledge, the only work with models publicly released is the universal litmus patterns work [13], where the models for the CIFAR10 and Tiny ImageNet classification tasks have been released. More recently, under IARPA’s TrojAI program¹, a software package [12] and multiple datasets have been released for different computer vision and NLP tasks. Our focus in this paper is on the image classification task in natural images, as opposed to synthetic images used in the TrojAI data collections. The datasets released so far for image classification have been limited in the number of classes supported (maximum is 200 classes in the Tiny ImageNet classification task). Furthermore, there has been no sufficient focus on the vision transformer architecture [5] despite its rising popularity. Therefore, we create a new dataset based on vision transformer models trained on the ImageNet dataset (1000 classes).

3 Approach

3.1 Threat Model

The attacker is assumed to train a K -class classifier and provide it to the victim such that the classifier works normally on clean inputs, but once a trigger is attached to an input, the classifier produces a certain class (the target class) of the attacker’s choice. The trigger is assumed to be small in size with respect to the input so that the attacker can deploy the attack in the physical world. The attacker achieves their goal by poisoning a fraction of the training dataset, which is done by adding the trigger to the poisoned fraction from all classes and giving them the target label as the ground truth label during training. Alternatively, the attacker can release a poisoned dataset to the public such that the victim can train the classifier on their end. In this case, the attacker can choose to use a clean-label poisoning mechanism that still allows the attacker to deploy the attack in the physical world.

The defender, who can be a third party different from the victim, has access to the trained model’s weights and hence can use gradient descent to create a signature for the model without the need for any data samples. The defender also can train a binary classifier (*a detector*) that can tell from the signature whether the model is Trojan or not. The detector is trained on signatures from a set of benign and Trojan models for the target K -class classification task. To train the detector, the defender needs access to pre-trained benign and Trojan models, which can be obtained from trusted sources, such as NIST’s TrojAI data, or can be created by the defender by training a small number of *shadow models* on a small set of clean data.

A similar threat model in the black box setting was used in [11, 13, 33]. We show that our approach still works in the black-box setting. However, it

¹ <https://www.iarpa.gov/research-programs/trojai>.

is important to note that targeting the white box case is still practical due to the wide-spread use of pretrained model weights downloaded from the web. In such cases, when the model weights are available, it is imperative to leverage them to enhance the detectability of Trojan models.

3.2 Intuition

Due to the way the attack is installed, the Trojan model develops a strong association between the trigger pattern and the target class. Such a strong association is expected to be evident upon model inversion. Namely, if we attempt to synthesize an image that maximizes or minimizes the activation associated with the target class, the trigger pattern is expected to have a fingerprint in such an image. Not only that, but the trigger’s fingerprint is expected to appear even if we are maximizing or minimizing the activations of other classes. For example, if our objective is to minimize the activation of a class other than the target one, the easiest way could be just to add the trigger to an image. Similarly, if the objective is to maximize such an activation instead, the model would make sure that it does not have any trace of the trigger. Therefore, whether we are maximizing or minimizing the activation of any class, the trigger can have a fingerprint on the resulting image.

3.3 Framework

Figure 1 illustrates the proposed framework, which generalizes the intuition outlined above. Given a trained K -class classifier, a signature is created by finding images that optimize M loss functions, which are computed based on the logits of the K classes. Therefore, M is a function of K . This results in M such images, which collectively constitute the signature for the model. A classifier is then trained to determine from the model’s signature whether it is Trojan or not, after an optional feature extraction step.

3.4 Activation Optimization

Let $f(x)$ be a K -class classification model. That is, $f : \mathbf{R}^{C \times H \times W} \rightarrow \mathbf{R}^K$, such that the input to the function f is a C -channel $H \times W$ image, and the output is a vector of K logits corresponding to the K classes. The i^{th} activation optimization map of the signature is defined as

$$a_i = \arg \min_x L_i(f(x)) , \quad (1)$$

where L_i is a loss function defined over the logits corresponding to an input x . Then the signature of the model is defined as

$$\mathcal{S} = [a_1 | a_2 | \dots | a_{M-1} | a_M] , \quad (2)$$

where $|$ is the channel-wise image concatenation operator.

In the current realization of our framework, we use $M \leq 2K$ loss functions, where $M = K$ when we use logit minimization or maximization as our loss functions, and $M = 2K$ when we combine logit maximization and minimization together. Let $f_j(x)$ be the j^{th} element of the output of f . In the case of combining minimization and maximization, the i^{th} loss function is defined as

$$L_i(f(x)) = \begin{cases} f_i(x) & i \in \mathbf{Z}^+, i \leq K \\ -f_{i-K}(x) & i \in \mathbf{Z}^+, K < i \leq 2K \end{cases} . \quad (3)$$

For the rest of the paper, unless otherwise specified, we will use the variant of the signature with $M = 2K$.

Regularization. The activation optimization process can be implemented using gradient descent starting from a random image. However, a number of regularizations are important to make the resulting images as natural as possible. Otherwise, we may end up having images that contain no useful patterns. In particular, we applied the following regularization techniques during activation optimization.

L₂ Regularization. This is the most common regularization technique used in model inversion. It works by adding the L_2 norm of the resulting image as a term in the loss. That is

$$R_{L_2}(x) = \|x\|_2 . \quad (4)$$

Total Variation Regularization. The total variation regularization [18] is used to enhance the smoothness of the generated image by minimizing the local gradients at every pixel. In particular, we minimize the L_1 norm of the local gradient in each channel as follows.

$$R_{TV}(x) = \sum_{ijk} |x(i, j, k) - x(i, j - 1, k)| + |x(i, j, k) - x(i - 1, j, k)| , \quad (5)$$

where $x(i, j, k)$ is the pixel value at location (i, j) in the k^{th} channel of x .

Adding the main loss and the regularization terms together, the i^{th} activation optimization map is obtained by

$$a_i = \arg \min_x L_i(f(x)) + \lambda_{L_2} R_{L_2}(x) + \lambda_{TV} R_{TV}(x) , \quad (6)$$

where λ_{L_2} and λ_{TV} are loss term weight parameters to be finetuned.

3.5 Feature Extraction

The size of our constructed model signature grows linearly with the number of classes. When the number of classes is large, training a classifier on the resulting signature may not be practical. To address this issue, we propose a feature

extraction step, which converts the signature into a fixed number of channels regardless of the number of classes. The idea is to use pixel-wise statistics over the signature channels. Consider the signature \mathcal{S} composed of M activation optimization maps, as shown in Eq. 2. Suppose that each activation optimization map contains c channels (typically $c = 3$). Then, \mathcal{S} has $N = cM$ channels in total. Consider the pixel at (i, j) in the N channels of \mathcal{S} . Let $s_{ij} = [s_{ij1} s_{ij2} \dots s_{ijN}]$ be a vector containing the values of the N channels at the (i, j) pixel location. Let $g : \mathbf{R}^N \rightarrow \mathbf{R}^P$ such that $u_{ij} = g(s_{ij})$ be a vector of P statistics computed over the values of s_{ij} . The compilation of the pixel statistics vectors constitute a P -channel feature map \mathcal{U} whose size is independent of the number of maps M in the raw signature \mathcal{S} . Specifically, we set $P = 11$, where the 11 statistics are as follows: minimum, maximum, sample mean, sample standard deviation, 0.25 quantile, median, 0.75 quantile, and four histogram bins.

As discussed in Sect. 3.4, our current realization uses a combination of activation minimization and activation maximization maps. That is $\mathcal{S} = [\mathcal{S}_{\min} | \mathcal{S}_{\max}]$, where \mathcal{S}_{\min} and \mathcal{S}_{\max} are the portions of \mathcal{S} that correspond to the activation minimization maps and the activation maximization maps, respectively. In addition to the pixel statistics feature map \mathcal{U} , whose values are computed over all channels of \mathcal{S} , we also construct \mathcal{U}_{\min} and \mathcal{U}_{\max} , which are pixel statistics feature maps computed over the channels of \mathcal{S}_{\min} and \mathcal{S}_{\max} , respectively. Therefore, our final pixel statistics feature map is $[\mathcal{U}_{\min} | \mathcal{U}_{\max} | \mathcal{U}]$ with a total of 33 channels.

4 Experimental Evaluation

4.1 Evaluation Data

To evaluate our method, we use two public datasets introduced in [13] and create our own dataset. All Trojan models in all the datasets are created via the BadNets method [9].

Public Datasets. One of the two public datasets is for models trained on the CIFAR10 dataset. The models are based on a modified version of the VGG architecture [25]. The other public dataset is for models trained on the Tiny ImageNet dataset. The models for the latter dataset are based on a shallow version of the ResNet18 architecture [10], which we will refer to as ResNet10. In both datasets, 20 different trigger patterns were used such that 10 of them appear only in the training models, and the other 10 appear only in the testing models. Each dataset has 1000 and 2000 models, respectively, for training, and 200 models for testing, all split equally between benign and Trojan models.

Our Dataset. Our own created dataset contains 1,200 models, with 600 benign and 600 Trojan. From each class, we use 500 models for training and the remaining 100 for testing. All models in our dataset are created from a pre-trained ViT-B-16 architecture [5] available with the `torchvision` package. Specifically, we used the weight version named `ViT_B_16_Weights.IMAGENET1K_V1`. Each model was then trained for one epoch on 90% of the ImageNet training set using the

AdamW optimizer with a learning rate of 10^{-5} and a batch size of 64. For each Trojan model, a random target class was chosen, and a randomly generated trigger was created and placed at a random location in 1% of the training data. A trigger was generated by first randomly sampling a 5×5 3-channel tensor and then resizing it to 32×32 using bicubic interpolation. The performance of the original ViT-B-16 model on the ImageNet validation data was 81%. After training for one epoch, the accuracy of our benign models dropped to around 79% (which could be due to overfitting), and the accuracy of the poisoned models on clean data was between 78% and 79%. Therefore, our Trojan models preserved performance on clean data. On the other hand, with the addition of triggers, the performance of Trojan models dropped to almost 0% in all victim classes, which means that the trigger was effective in poisoning the model.

4.2 Implementation Details

Signatures were created using the Adam optimizer with 200 iterations. A learning rate of 10 was used with the CIFAR10 dataset while a learning rate of 0.1 was used with the Tiny ImageNet and the ImageNet datasets. For the CIFAR10 dataset, it was important to standardize the final image so that it has pixel values with 0.5 mean and 0.25 standard deviation.

L_2 regularization was implemented by setting the weight decay argument of the optimizer to 10^{-5} . The weight for the total variation regularization was set to 10^{-3} for the CIFAR10 and ImageNet datasets, and was set to 10^{-2} for the Tiny ImageNet dataset.

The detection classifier model (ResNeXt-50 ($32 \times 4d$)), was trained using the Adam optimizer with a learning rate of 10^{-4} , and with 100 epochs. 90% of the training samples were used for training and the remaining 10% were used for validation.

4.3 Evaluation Results

Sample signatures for one Trojan model and one benign model from the CIFAR10 Trojan dataset are shown in Fig. 2. In this dataset, the trigger was placed at the corners of the image. You can see the footprint of the trigger clearly at the corners of the Trojan model’s signature, particularly in the activation minimization maps (top two rows).

In the remainder of this section, we focus on comparing different variants of our methods to prior research. Table 1 shows the detection accuracy and the area under the receiver operating characteristics curves (AUC) for detecting Trojan attacks using our method and three baseline methods on the three datasets. As explained in Sect. 3.4, we applied three variants of our method using activation minimization, activation maximization, and both, in which case we concatenated the signature channels coming from the former two optimizations. We also used the pixel statistics channels as explained in Sect. 3.5. In Table 1, for all experiments on CIFAR10 and Tiny ImageNet and for the statistics experiment on ImageNet, we present the average and the standard deviation of the metrics

over ten independent training sessions for each classifier. For activation maximization, minimization, and their combination on ImageNet, we only trained three models for each due to the heavy computational cost.

The three baseline methods, which are ULP [13], k-Arm optimization [23], and MM-BD [28], were chosen based on the availability of their code and its adaptability to new datasets. For ULP, we used the publicly available code for CIFAR10 and Tiny ImageNet. We applied the best configuration in the paper for each dataset, which was ten litmus patterns for CIFAR10 and five for Tiny ImageNet. We created ten sets of litmus patterns for each dataset. In Table 1, we report the mean and standard deviation of the AUC and accuracy scores over the litmus pattern sets. It is worth noting that we could not reproduce or even come close to the results reported in the ULP paper despite using the code released by the authors. For the ImageNet dataset, we could not get the method to work due to excessive computational cost and lack of convergence.

For k-Arm optimization, we adapted the publicly released implementation and evaluated it on the three datasets. We used the Trigger Size output for each model as the score based on which we computed the AUC. Again, we evaluated the method ten times for each probe model with different random seeds. We report the mean and standard deviation of the resulting metrics in Table 1.

For the MM-BD method, we adapted the publicly available code to work with our datasets. We found that the default number of steps used in the paper (300) was too small for the models to converge. For a fair comparison, in all our experiments, we let the optimization run until convergence. We used ten different runs for each model in the CIFAR10 and the Tiny ImageNet datasets. However, due to the excessive computational time, we only used one run for the ImageNet dataset.

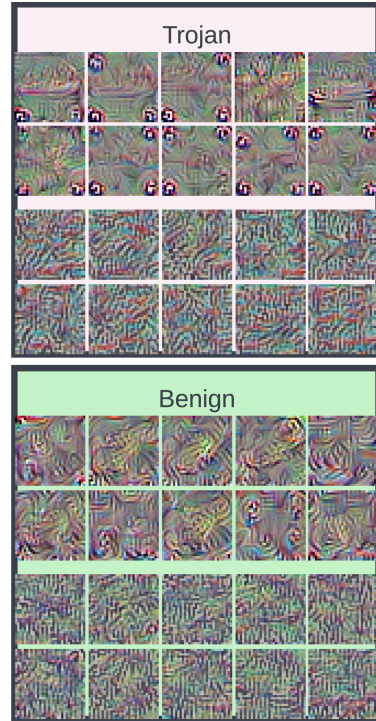


Fig. 2. Sample signatures from the CIFAR10 Trojan dataset. Each signature has 20 images corresponding to the 10 classes of the dataset. The top two rows of each signature are for the activation minimization maps while the bottom two rows are for the activation maximization maps. Note how the trigger has a clear fingerprint in the minimization maps for the signature of the Trojan model.

In Figs. 3a to 3c, box plots are used to present the AUC scores for all the runs. From the results in Table 1 and Figs. 3a to 3c, we can observe that, in each dataset, at least one of our four variants surpasses or matches the baseline performance, regardless of whether the probe model is CNN or ViT-based. Moreover, when our method surpasses the baseline methods, the margin is statistically significant.

It is interesting to observe that the pixel-wise statistics variant is the only variant that consistently outperforms or matches all baseline methods. It is also the best in the case of CIFAR10 and Tiny ImageNet, achieving the highest mean score and the lowest standard deviation. However, for ImageNet, the variant that combines both types of activation optimization maps achieves the best performance. It is also interesting to notice that the activation minimization variant consistently performs better than the activation maximization one. This result is surprising given that all prior work on model inversion focused on activation maximization (or alternatively minimizing the classification loss, e.g. the cross-entropy loss). Here, for the first time, we find a good use for activation minimization-based model inversion.

Out of the three baseline methods, the only serious contender is the MM-BD method. In fact, this method achieves a perfect AUC score on the Tiny ImageNet dataset (though its accuracy is not the best). However, similar to the other two baseline methods, MM-BD struggles on the ImageNet dataset. We believe this struggle is due to the ViT architecture, in which the main assumption of the MM-BD method (the presence of an anomalously large logit margin for the target class in a Trojan model) may not hold.

Table 1. Comparative performance results.

		CIFAR10		Tiny ImageNet		ImageNet	
		AUC	Acc.	AUC	Acc.	AUC	Acc.
ULP		0.64 (0.060)	0.61 (0.048)	0.74 (0.075)	0.71 (0.066)	–	–
k-Arm		0.68 (0.028)	0.51 (0.000)	0.65 (0.120)	0.54 (0.024)	0.51 (0.67)	0.5 (0.000)
MM-BD		0.90 (0.012)	0.79 (0.029)	1.00 (0.000)	0.97 (0.009)	0.59	0.51
TRIGS	Both	0.95 (0.022)	0.90 (0.038)	0.98 (0.010)	0.93 (0.014)	0.94 (0.015)	0.87 (0.033)
	Max	0.60 (0.067)	0.57 (0.054)	0.93 (0.016)	0.83 (0.047)	0.73 (0.013)	0.66 (0.020)
	Min	0.96 (0.011)	0.92 (0.019)	0.96 (0.015)	0.92 (0.013)	0.82 (0.108)	0.75 (0.083)
	Stats	0.99 (0.003)	0.96 (0.011)	1.00 (0.001)	0.99 (0.010)	0.84 (0.046)	0.76 (0.050)

4.4 Sensitivity to Chosen Statistics

Since the pixel-wise statistics variant is the most efficient and provides the best performance on average, we study its sensitivity to varying the number of used statistics. We focus on the ImageNet dataset because the other two datasets are almost saturated. The results in Table 1 are for 11 statistics (Sect. 3.5). We experimented with adding four more quantiles at 0.125, 0.375, 0.625, 0.875. We

also experimented with different numbers of histogram bins. Figure 3d shows the results of these experiments. The bars represent the mean AUC over 10 training sessions and the error lines represent the range of values. There is no clear advantage of adding more quantiles. However, more histogram bins slightly enhance the performance at the cost of higher memory and computational costs.

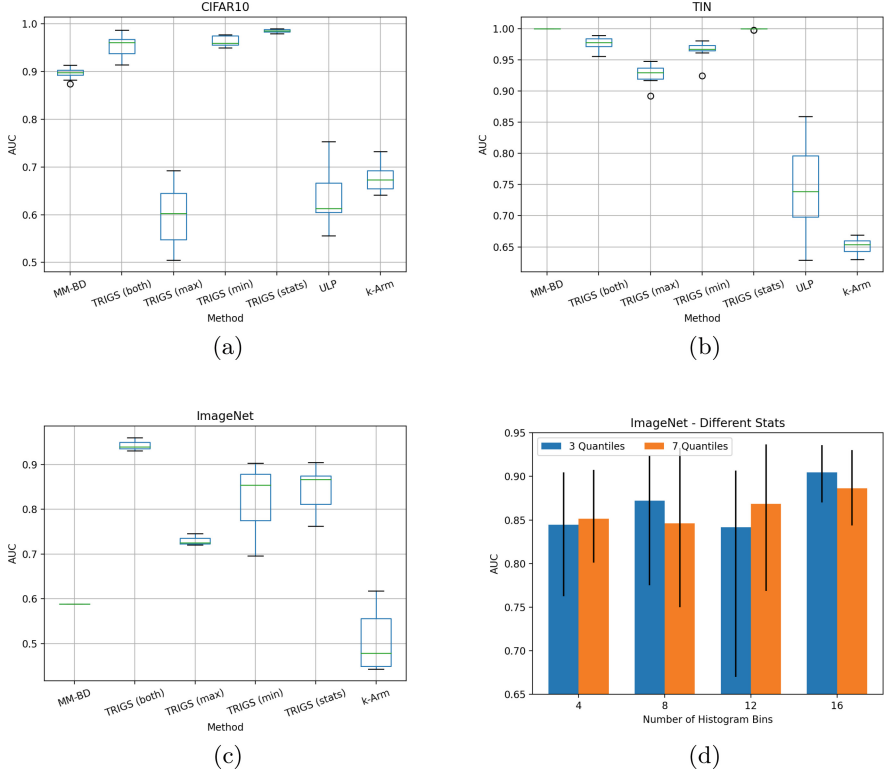


Fig. 3. (a-c) AUC Box plots for the main three evaluation datasets. (d) Average AUC with different number of histogram bins and quantiles for the ImageNet dataset. The error lines show the range of values.

4.5 Stronger Threat Models

In this section, we study the effect of having a stronger threat model. In particular, we study three aspects of the threat model: (1) the data available to the defender for training shadow models is different and much smaller than the data available to the attacker, (2) the defender uses a different architecture to train the shadow models, and (3) the defender uses a small number of shadow models.

To conduct these experiments, we created another set of models trained on the Tiny ImageNet dataset. To mimic the effect of having different and smaller

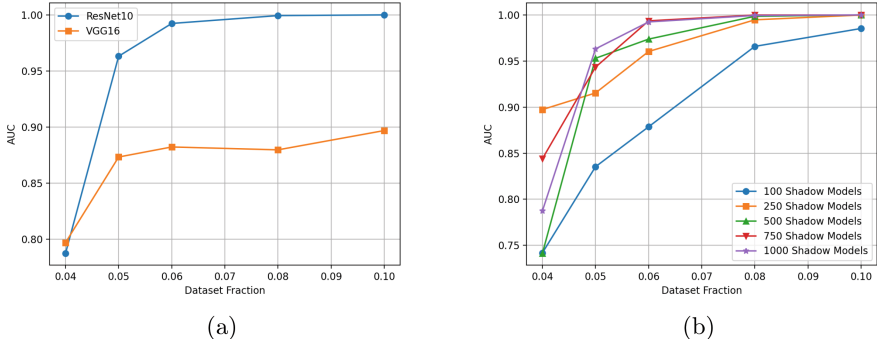


Fig. 4. Average AUC values vs (a) different fractions of the Tiny ImageNet dataset used to train the ResNet10 shadow models, and (b) same as (a) but across different numbers of shadow models used to train the detector.

data available to the defender, we split the dataset into two disjoint sets: a large set consisting of 50% of the original training data used only by the attacker (i.e. the testing models.) and 4–10% of the data used only by the defender to train the shadow models. All shadow models were trained on the ResNet10 architecture adopted in [13]. For each percentage of the data used to train the shadow models, we trained 1000 of them, split as 500 benign and 500 Trojan models. For the testing models (representing the attacker’s trained models), we trained 200 models using the ResNet10 architecture and 200 models using the VGG16 architecture. For each architecture, half of the models were benign and the other half were Trojan. Each model, whether used for training or testing the detector, was trained on a unique random trigger created in a similar way to what we used for the ImageNet-ViT dataset, but using a trigger size of 8×8 . Triggers are placed in random locations in 2% of the training data in the case of the testing models, and in 5% of the training data in the case of the training models. The reason for having different poisoning fractions is that as the size of the training data reduces, we found that a higher poisoning fraction is needed to achieve a high attack success rate (typically $\sim 98\%$).

Figure 4a shows the average AUC plots for these experiments. Each point is an average of 10 different runs. For these experiments, we used the pixel-wise statistics variant of our method. As can be observed from the plots, as low as 6% of the dataset is enough for excellent performance if the architecture of the shadow models matches with that of the probe models. When the architectures are different, despite the drop in performance, it is still higher than the baseline methods, ranging from around 0.8 to 0.9 AUC.

In another experiment, we study the effect of reducing the number of shadow models used to train the detector. We originally trained 1000 models for each fraction of the Tiny ImageNet dataset. We evaluate the performance when only 100, 250, 500, or 750 models are used to train the detector. The results are shown in Fig. 4b. In these results, the ResNet10 architecture is used for the

testing models. Each point in the plot is an average of 10 different runs. The performance does not degrade much if we reduce the number of shadow models down to 250, especially if we use at least 8% of the Tiny ImageNet dataset for training the shadow models. However, going further down to 100 models can hurt the performance.

5 Conclusion

In this paper, we present a new method for detecting Trojan models named TRIGS for TRojan Identification from Gradient-based Signatures. TRIGS applies a data-driven approach, where a signature of a trained model is constructed using activation optimization, and a classifier detects whether the model is Trojan or not based on the signature. On two public datasets as well as our own created challenging dataset, TRIGS achieves state-of-the-art performance, in most cases surpassing baseline methods by large margins. TRIGS works well regardless of whether the probe model architecture is convolutional or a vision transformer. It also works very well when the defender only has access to a small amount of clean samples. Our dataset will be the first public dataset for Trojan detection that is composed only of models based on the vision transformer architecture and trained on a 1000-class classification task (those of the ImageNet dataset).

Acknowledgement. This research is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), under cooperative agreement number HR00112020009. The views and conclusions contained herein should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
2. Chen, B., et al.: Detecting backdoor attacks on deep neural networks by activation clustering. In: AAAI's Workshop on Artificial Intelligence Safety (2019)
3. Chen, H., Fu, C., Zhao, J., Koushanfar, F.: DeepInspect: a black-box trojan detection and mitigation framework for deep neural networks. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (2019). <https://doi.org/10.24963/ijcai.2019/647>
4. Dong, Y., et al.: Black-box detection of backdoor attacks with limited information and data. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021). <https://doi.org/10.1109/ICCV48922.2021.01617>

5. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
6. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing Higher-Layer Features of a Deep Network. Technical Report, Univeristé de Montréal (2009)
7. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (2015). <https://doi.org/10.1145/2810103.2813677>
8. Geiping, J., et al.: Witches' Brew: industrial scale data poisoning via gradient matching. In: International Conference on Learning Representations (2022)
9. Gu, T., Dolan-Gavitt, B., Garg, S.: BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain (2019)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). <https://doi.org/10.1109/CVPR.2016.90>
11. Huang, S., Peng, W., Jia, Z., Tu, Z.: One-pixel signature: characterizing CNN models for backdoor detection. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12372, pp. 326–341. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58583-9_20
12. Karra, K., Ashcraft, C., Fendley, N.: The TrojAI Software Framework: An Open-Source tool for Embedding Trojans into Deep Learning Models (2020). <https://doi.org/10.48550/arXiv.2003.07233>
13. Kolouri, S., Saha, A., Pirsiavash, H., Hoffmann, H.: Universal litmus patterns: revealing backdoor attacks in CNNs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
14. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Neural attention distillation: erasing backdoor triggers from deep neural networks. In: International Conference on Learning Representations (2021)
15. Liu, Y., Lee, W.C., Tao, G., Ma, S., Aafer, Y., Zhang, X.: ABS: scanning neural networks for back-doors by artificial brain stimulation. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (2019). <https://doi.org/10.1145/3319535.3363216>
16. Liu, Y., et al.: Trojaning attack on neural networks. In: Proceedings 2018 Network and Distributed System Security Symposium (2018). <https://doi.org/10.14722/ndss.2018.23291>
17. Liu, Y., Shen, G., Tao, G., Wang, Z., Ma, S., Zhang, X.: Complex backdoor detection by symmetric feature differencing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
18. Mahendran, A., Vedaldi, A.: visualizing deep convolutional neural networks using natural pre-images. *Inter. J. Comput. Vis.* **120**(3) (2016). <https://doi.org/10.1007/s11263-016-0911-8>
19. Nguyen, A., Yosinski, J., Clune, J.: Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks (2016). <https://doi.org/10.48550/arXiv.1602.03616>
20. Nguyen, A., Yosinski, J., Clune, J.: Understanding neural networks via feature visualization: a survey. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R. (eds.) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. LNCS (LNAI), vol. 11700, pp. 55–76. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28954-6_4

21. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34 (2020). <https://doi.org/10.1609/aaai.v34i07.6871>
22. Shafahi, A., et al.: Poison frogs! targeted clean-label poisoning attacks on neural networks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
23. Shen, G., et al.: Backdoor scanning for deep neural networks through K-Arm optimization. In: Proceedings of the 38th International Conference on Machine Learning (2021)
24. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps (2014). <https://doi.org/10.48550/arXiv.1312.6034>
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
26. Souri, H., Fowl, L., Chellappa, R., Goldblum, M., Goldstein, T.: Sleeper agent: scalable hidden trigger backdoors for neural networks trained from scratch. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
27. Wang, B., et al.: Neural cleanse: identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy (SP) (2019). <https://doi.org/10.1109/SP.2019.00031>
28. Wang, H., Xiang, Z., Miller, D.J., Kesidis, G.: MM-BD: post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In: 2024 IEEE Symposium on Security and Privacy (SP) (2023). <https://doi.org/10.1109/SP54263.2024.00015>
29. Wang, R., Zhang, G., Liu, S., Chen, P.-Y., Xiong, J., Wang, M.: Practical detection of trojan neural networks: data-limited and data-free cases. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12368, pp. 222–238. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_14
30. Weber, M., Xu, X., Karlas, B., Zhang, C., Li, B.: RAB: provable robustness against backdoor attacks. In: 2023 IEEE Symposium on Security and Privacy (SP) (2023). <https://doi.org/10.1109/SP46215.2023.00037>
31. Xiang, Z., Miller, D.J., Kesidis, G.: Revealing backdoors, post-training, in dnn classifiers via novel inference on optimized perturbations inducing group misclassification. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2020). <https://doi.org/10.1109/ICASSP40776.2020.9054581>
32. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: simple and efficient design for semantic segmentation with transformers. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
33. Xu, X., Wang, Q., Li, H., Borisov, N., Gunter, C.A., Li, B.: Detecting AI trojans using meta neural analysis. In: 2021 IEEE Symposium on Security and Privacy (SP) (2021). <https://doi.org/10.1109/SP40001.2021.00034>
34. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H.: Understanding Neural Networks Through Deep Visualization (2015)
35. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53



Multifaceted Anchor Nodes Attack on Graph Neural Networks: A Budget-Efficient Approach

Huanzhang Zhu¹, Shaoxin Li¹, and Lingyang Chu¹

McMaster University, Hamilton, ON, Canada
{zhuh98, li2018, chu19}@mcmaster.ca

Abstract. Structural adversarial attack methods that attack a graph neural network by perturbing the edges of the input graph are well-known for their strong effectiveness. However, most existing structural attacks focus on achieving high attack performance, but they ignore the high cost of budget to control (i.e., buy out or hijacking) the nodes (i.e., user accounts in a social network) when executing the attacks in real-world networks. The classic anchor nodes attacks are more budget-efficient because they only control a small set of anchor nodes to conduct all the attacks. However, their attack effectiveness is also limited by the restriction of using one set of anchor nodes. In this paper, we develop a strong and budget-efficient multifaceted anchor nodes attack on graph neural networks. The key idea is to simultaneously train multiple sets of anchor nodes and an assignment network, such that the assignment network can select the best set of anchor nodes to conduct each new attack successfully. This significantly improves the attack effectiveness while keeping the budget of controlled nodes small. Extensive experiments on five real-world datasets demonstrate the outstanding performance of our method. Our code and Appendix is available at <https://github.com/zhz0108/mfan/>.

Keywords: Adversarial attack · Graph neural network

1 Introduction

Modern graph neural networks (GNN) are widely employed in many real-world application scenarios, such as social network analysis [4, 24], recommendation systems [6, 13] and drug discovery [9]. However, GNNs are known to be vulnerable to adversarial attacks [11, 18, 28, 30, 33, 39], which may potentially cause severe negative impacts on the society. For instance, fraudulent users on social networks, such as spam bots, phishing accounts and scam accounts, may establish seemingly genuine friendships to evade account validation checks. Hoax articles on Wikipedia can effectively disguise themselves by carefully modifying their links [20]. A money laundering account may also evade detection by conducting seemingly normal transactions with other legitimate accounts.

To improve security and robustness of GNNs, many works have been proposed to discover potential security loopholes in trained GNN models by developing strong adversarial attacks. Among the existing adversarial attacks on GNNs [5, 8, 11, 16, 18, 22, 23, 28, 30, 32, 33, 36–39], the structural attack methods [5, 7, 12, 16, 22, 23, 32, 34, 36–38], which attack a GNN by perturbing (i.e., modifying) the edges of the input graph, are well-known for their strong effectiveness. However, most existing structural attack methods [5, 12, 16, 22, 23, 32, 36, 38] focus on improving the attack effectiveness, but they ignore the huge cost of budget to control nodes in order to perturb edges. For example, in order to perturb an edge in a social network, the adversary needs to take control of at least one node connected to the edge, which often requires either paying a user to buy out his/her account or paying a hacker to hijack the account. If an adversary perturbs many edges, it needs to control a large number of nodes, which not only costs a big budget to realize the attack but also increases the risk of being detected.

How to conduct a strong and budget-efficient attack against GNNs is a novel problem that has not been systematically studied in the literature. As discussed later in Sect. 2, both the targeted structural attacks [5, 7, 12] and the global structural attacks [1, 16, 23, 32, 38] have to control many nodes in order to achieve a good attack performance. The huge budget required to control the nodes significantly reduces their cost-effectiveness. In comparison, the anchor nodes attacks [36, 37] are more budget-efficient because they only need to control a small set of anchor nodes to attack all the target nodes. However, due to the limited attack effectiveness of a single set of anchor nodes, existing anchor nodes attacks [36, 37] often cannot achieve good attack performance.

In this paper, we develop a budget-efficient Multi-Faceted Anchor Nodes (MFAN) attack against GNNs performing node classification tasks. The key idea is to train multiple sets of anchor nodes together with an assignment network, where each set of anchor nodes successfully attacks a different set of target nodes, and the assignment network accurately selects the best set of anchor nodes to attack each new target node. In this way, MFAN achieves outstanding attack performance by conducting “divide and conquer”, which successfully attacks the union of the nodes that are successfully attacked by each set of anchor nodes. MFAN is also budget-efficient because it only controls a small number of anchor nodes. Specifically, we make the following contributions. First, we propose a novel adversarial attack task that aims to achieve strong attack effectiveness while costing a low budget of controlled nodes. Second, we successfully tackle the task by developing the strong and budget-efficient MFAN attack. Last, we conduct extensive experiments on five real-world datasets to demonstrate the outstanding performance of MFAN.

2 Related Work

Multifaceted anchor nodes attack on graph neural networks is a novel problem that has not been systematically tackled before. Since our work only modifies the

edge structure of the input graph, it is substantially different from the existing works that perform attacks by modifying node features [2, 8, 29, 40] or injecting poisonous nodes [11, 18, 28, 30, 33, 39]. Instead, it is more related to the structural attacks [1, 5, 7, 12, 16, 22, 34–38] that reduce the classification performance of a victim graph neural network model by perturbing (i.e., adding or removing) the edges of the input graph. We discuss the relationship between our work and the related existing works in detail as follows.

Targeted Structural Attacks. [5, 7, 12]. A targeted structural attack causes the victim model to fail its classification on a single node by perturbing a small set of edges in the graph. RL-S2V [12] employs hierarchical reinforcement learning to generate edge perturbations, which significantly reduces the prediction accuracy of the victim model. FGA [7] leverages iterative gradient information of pairwise nodes from a trained graph convolutional neural network to generate edge perturbations. GF-Attack [5] generates edge perturbations by employing graph embedding learning with a corresponding graph filter. The above targeted structural attacks generate a different perturbation of edges for each specific target node to attack. Since perturbing edges requires control of the nodes connected to the edges, launching a new attack by perturbing a new set of edges requires control of a new set of nodes. Since controlling each new node costs a proportion of the budget, the targeted structural attacks are not budget-efficient. When the budget is limited, these attacks cannot afford to control enough new nodes to attack many target nodes.

Global Structural Attacks. [1, 16, 22, 23, 32, 34, 38]. A global structural attack aims to reduce the overall classification accuracy of a victim model on all the nodes of the graph by perturbing a large set of edges once and for all. Meta-Self [41] uses meta-learning to perform attacks by treating the adjacency matrix of the graph as a hyper-parameter. [35] proposes two perturbation methods using first-order attack generation. [16] launches attacks on large-scale graphs by using projected randomized block coordinate descent (BCD) and greedy randomized BCD to sample the set of edges to perturb. There are also works that perform effective global structural attacks by eliminating gradient bias [23], using Eigen decomposition [1], and utilizing a certified robustness-inspired framework [32]. The above global structural attacks often require perturbing a large number of edges in order to achieve a good attack performance. These methods are not budget-efficient, because perturbing a large number of edges requires control of many nodes, which demands a substantial budget.

Anchor Nodes Attacks. [36, 37]. An anchor nodes attack is a special type of structural attack. It first finds a constant set of nodes, named anchor nodes, then fails the classification of the victim model on each target node by flipping the edges between the anchor nodes and the target node. Here, flipping an edge means adding a non-existing edge or deleting an existing edge. As the first work in this line, GUA [37] identifies a set of anchor nodes by a minimum perturbation iteratively. The follow-up work GUAP [36] generates a set of new nodes and uses them as anchor nodes to launch attacks. Both GUA and GUAP are budget-

efficient because they only require control of a small constant set of anchor nodes in order to attack all the target nodes. However, their attack effectiveness is largely restricted by the limitation of using only a single set of anchor nodes for all the attacks. The proposed multifaceted anchor nodes attack differs from GUA and GUAP because it smoothly incorporates multiple sets of anchor nodes with a well-trained assignment network to significantly improve attack effectiveness while keeping a low budget of controlled nodes.

3 Preliminary: Graph Neural Network

Denote by $G = (V, E, A, X)$ an unweighted graph, where V is the set of nodes, E is the set of edges, and $N = |V|$ is the number of nodes in G . The adjacency matrix $A \in \{0, 1\}^{N \times N}$ describes the edge structure of G . Each node in G is associated with a d -dimensional feature vector; we represent the feature vectors of all the nodes by a feature matrix $X \in \mathbb{R}^{N \times d}$, where the i -th row of X corresponds to the feature vector of the i -th node v_i in G . Each node in graph G is associated with a class label in one of the C total number of classes. A graph neural network, denoted by $f(X, A)$, takes the feature matrix X and the adjacency matrix A of a graph G as the input and predicts the class labels for the nodes in G .

Following the literature [7, 35–37, 40, 41], we target a classic graph neural network named graph convolutional network (GCN) [19] as the victim model to attack. A typical GCN consists of one or more hidden graph convolution layers followed by a softmax layer to produce the final prediction. The hidden graph convolution layer is defined as

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}), \quad (1)$$

where l is the number of convolution layers, $\sigma(\cdot)$ is the activation function ReLU [15], and $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is a normalized adjacency matrix with $\tilde{A} = A + I$ and diagonal degree matrix $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. After adding the final softmax layer, a typical GCN with one hidden graph convolution layer is represented by

$$f(X, A) = \text{softmax}(\hat{A} \text{ReLU}(\hat{A}XW^{(0)}) W^{(1)}), \quad (2)$$

where $W^{(0)}$ and $W^{(1)}$ are the model parameters of the GCN. We write $f(X, A)$ in short as f when the context is clear. The output of the GCN f , denoted by

$$Z = f(X, A), \quad (3)$$

is a matrix with N rows and C columns, where the i -th row and j -th column entry, denoted by Z_{ij} , is the probability of the i -th node being predicted as the j -th class, $1 \leq j \leq C$. We also write the i -th row of Z as $f(X, A)_i$.

4 Task Definition and Problem Formulation

4.1 The Task of Multifaceted Anchor Nodes Attack

In the task of multifaceted anchor nodes (MFAN) attack, we aim to attack a victim GCN f trained on an unweighted graph G . We perform the attack by perturbing (i.e., modifying) the edges in G to generate a perturbed graph G' , such that the label of a target node v_i predicted by f on G' is different from the label of v_i predicted by f on G .

Following the routine of [36,37], the perturbation on G is defined as a set of edge modifications (i.e., adding or removing edges) induced by a set of nodes in G , named **anchor nodes**. Denote by $Q \subset V$ a set of anchor nodes, a **perturbation** induced by Q to attack f 's prediction on a target node v_i is to flip the edges between each node in Q and v_i . Here, **flip** means adding a non-existing edge or deleting an existing edge. We refer to "change f 's prediction on a target node v_i " as "attack the target node v_i " in short.

In our work, we aim to train K sets of anchor nodes, denoted by a collection $\mathcal{Q} = \{Q_1, \dots, Q_K\}$, where each set of anchor nodes $Q_k \in \mathcal{Q}$ specializes in attacking a large subset of target nodes in V . Together with the training of \mathcal{Q} , we also train an **assignment network**, denoted by g_θ , which is used to select the best-suited set of anchor nodes from \mathcal{Q} to attack a target node v_i . We define the MFAN task as follows.

Definition 1. *Given an integer budget $\xi > 0$ on the number of controlled nodes for each perturbation and a victim GCN f trained on an unweighted graph G , the task of **multifaceted anchor nodes (MFAN) attack** is to train an attack model, composed by K sets of anchor nodes $\mathcal{Q} = \{Q_1, \dots, Q_K\}$ and an assignment network g_θ , such that*

1. *the size of each set of anchor nodes is not larger than ξ ;*
2. *each target node $v_i \in V$ is attacked by the perturbation induced by the set of anchor nodes $Q_k \in \mathcal{Q}$ that is selected by the assignment network g_θ ; and*
3. *our attack model can successfully attack most of the nodes in G .*

Compared to the classic anchor nodes attacks [36,37], the MFAN attack is multifaceted because it uses more than one set of anchor nodes. The assignment network g_θ takes G and the target node v_i as the input to select the best-suited set of anchor nodes that has the largest chance to successfully attack v_i . Since different sets of anchor nodes are specialized in attacking different subsets of target nodes, the proposed MFAN attack is essentially performing "divide and conquer" to successfully attack the union of target nodes that are attacked by each set of anchor nodes. This significantly boosts the attack effectiveness of MFAN. The budget ξ limits the number of controlled nodes for each set of anchor nodes. Since the same sets of anchor nodes are used to attack all the target nodes in G , the number of anchor nodes to control is very small, which makes MFAN extremely budget-efficient.

We use the same white-box setting in [36,37] that an adversary has full access to the structure and parameters of the victim model f . However, as demonstrated

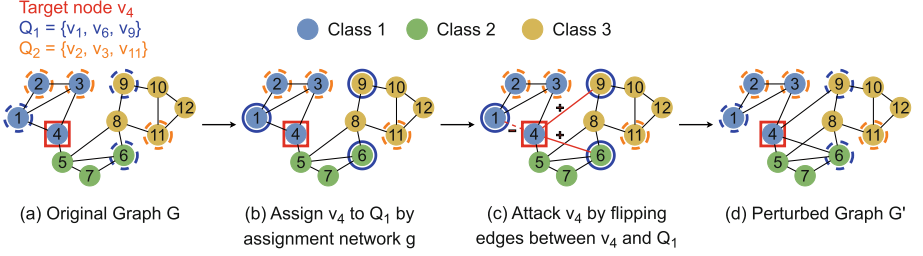


Fig. 1. An example of MFAN attack. To attack the target node v_4 in the original graph G , the assignment network selects the anchor nodes set $Q_1 = \{v_1, v_6, v_9\}$. Then it flips the edges between v_4 and Q_1 , generating the perturbed graph G' .

later in Sect. 6.2, the attack model trained by MFAN on a white-box victim model f can be straightforwardly transferred to successfully attack other black-box models.

4.2 Modelling the Assignment Network and the Perturbation

We model the **assignment network** as a GCN, denoted by $g_\theta(X, A)$, that predicts the probabilities of selecting each of the K sets of anchor nodes in \mathcal{Q} to attack each target node v_i in G . The output of $g_\theta(X, A)$ is an N -by- K matrix, where the entry in the i -th row and the k -th column, denoted by $g_\theta(X, A)_{ik}$, is the probability of selecting the set of anchor nodes $Q_k \in \mathcal{Q}$ to attack the target node v_i in G .

Given a set of anchor nodes $Q_k \in \mathcal{Q}$, MFAN attacks the target node v_i by conducting a perturbation on G , which flips the edges between v_i and each anchor node in Q_k . To mathematically model this perturbation, we first represent the set of anchor nodes in Q_k by a perturbation vector $\mathbf{p}_k \in \{0, 1\}^N$. The i -th element of \mathbf{p}_k being equal to 1 means the i -th node v_i in G is an anchor node in Q_k . In this way, the sets of anchor nodes in \mathcal{Q} are modelled by the set of perturbation vectors $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_K\}$. Then, we follow [36, 37] to model the **perturbation** induced by a perturbation vector $\mathbf{p}_k \in \mathcal{P}$ to attack v_i as

$$\rho(v_i, \mathbf{p}_k) = (\mathbf{1} - P) \circ A + P \circ (\mathbf{1}_0 - A), \quad (4)$$

where ρ is the perturbation function, \circ means element-wise multiplication, $\mathbf{1}$ is an N -by- N matrix with all entries equal to one, $\mathbf{1}_0$ is an N -by- N matrix of all ones except the diagonal entries being zeros, and P is an N -by- N matrix with i -th row and i -th column replaced by \mathbf{p}_k and the other entries equal to zero.

According to [36, 37], the output of $\rho(v_i, \mathbf{p}_k)$, denoted by $A' = \rho(v_i, \mathbf{p}_k)$, is the **perturbed adjacency matrix** of the **perturbed graph G'** , where the edges in G between v_i and each anchor node in Q_k are flipped. Figure 1 shows an example of the attacking procedure of MFAN.

4.3 Formulating the Problem

We formulate the loss of the proposed MFAN attack as

$$\mathcal{L}(\mathcal{P}, \theta) = - \sum_{i=1}^N \sum_{k=1}^K g_{\theta}(X, A)_{ik} \cdot \text{CE}(f(X, A')_i, f(X, A)_i), \quad (5)$$

where $A' = \rho(v_i, \mathbf{p}_k)$ is the perturbed adjacency matrix induced by the perturbation vector $\mathbf{p}_k \in \mathcal{P}$, f is the victim model, $f(X, A')_i$ and $f(X, A)_i$ represent the predicted class distributions of v_i on the perturbed graph G' and original graph G , respectively, and $\text{CE}(\cdot, \cdot)$ is the cross entropy loss.

In Eq. (5), the cross entropy term measures the difference between $f(X, A')_i$ and $f(X, A)_i$ when we use \mathbf{p}_k to attack the target node v_i . Since $g_{\theta}(X, A)_{ik}$ is the probability of selecting \mathbf{p}_k to attack v_i , the summation term $\sum_{k=1}^K$ is computing the expected difference between $f(X, A')_i$ and $f(X, A)_i$. According to the task definition in Definition 1, our goal is to successfully attack most of the nodes in G , which can be achieved by maximizing the expected difference between $f(X, A')_i$ and $f(X, A)_i$ on all the nodes in G . As a result, we formulate the MFAN attack as the following optimization problem.

$$\min_{\mathcal{P}, \theta} \mathcal{L}(\mathcal{P}, \theta) \quad \text{s.t.} \quad \forall \mathbf{p}_k \in \mathcal{P}, \|\mathbf{p}_k\|_1 \leq \xi, \mathbf{p}_k \in \{0, 1\}^N, \quad (6)$$

where the constraint $\|\mathbf{p}_k\|_1 \leq \xi$ requires the number of anchor nodes in each set $Q_k \in \mathcal{Q}$ to be no larger than the budget ξ .

5 Solving the Formulated Problem

The original optimization problem in Eq. (6) is a constrained integer programming problem, which is NP-hard and cannot be straightly solved by gradient-based methods. A typical solution often involves two steps.

Step-1: we relax each integer-valued constraint $\mathbf{p}_k \in \{0, 1\}^N$, $\mathbf{p}_k \in \mathcal{P}$, to a real-valued constraint $\mathbf{p}_k \in [0, 1]^N$; this converts the original optimization problem in Eq. (6) to

$$\min_{\mathcal{P}, \theta} \mathcal{L}(\mathcal{P}, \theta) \quad \text{s.t.} \quad \forall \mathbf{p}_k \in \mathcal{P}, \|\mathbf{p}_k\|_1 \leq \xi, \mathbf{p}_k \in [0, 1]^N. \quad (7)$$

Step-2: following the standard penalty method [3, 14, 27], we incorporate each constraint $\|\mathbf{p}_k\|_1 \leq \xi$ as a penalty term $\max(\|\mathbf{p}_k\|_1 - \xi, 0)$ in the loss function. This converts the problem in Eq. (7) to

$$\min_{\mathcal{P}, \theta} \mathcal{L}(\mathcal{P}, \theta) + \lambda \sum_{\mathbf{p}_k \in \mathcal{P}} \max(\|\mathbf{p}_k\|_1 - \xi, 0) \quad \text{s.t.} \quad \forall \mathbf{p}_k \in \mathcal{P}, \mathbf{p}_k \in [0, 1]^N. \quad (8)$$

Algorithm 1: Training \mathcal{P} and θ

Input : ξ, f, G , and a training set of nodes $V_T \subset V$.
Output: \mathcal{P} and θ

- 1 Randomly initialize $\mathcal{P} \leftarrow \mathcal{P}^{(0)}$ and $\theta \leftarrow \theta^{(0)}$; and set $T = 1, \lambda = 0.1$ and $max_epoch = 120$.
- 2 **while** $epoch < max_epoch$ **do**
- 3 **for** each mini-batch in V_T **do**
- 4 **for** each $\mathbf{p}_k \in \mathcal{P}$ **do**
- 5 $\mathbf{p}_k \leftarrow \mathbf{p}_k - \eta_1 \nabla_{\mathbf{p}_k} \mathcal{L}^*(\mathcal{P}, \theta)$
- 6 $\mathbf{p}_k \leftarrow \text{clip}(\mathbf{p}_k, 0, 1)$
- 7 **end**
- 8 $\theta \leftarrow \theta - \eta_2 \nabla_{\theta} \mathcal{L}^*(\mathcal{P}, \theta)$
- 9 **end**
- 10 Update $\lambda \leftarrow \lambda \times 5$ for every 20 epochs when $epoch \leq \frac{1}{2} max_epoch$.
- 11 Update $T \leftarrow \frac{T}{2}$ for every 5 epochs when $epoch > \frac{1}{2} max_epoch$.
- 12 **end**
- 13 $\mathcal{P} \leftarrow \text{quantize}(\mathcal{P}, \xi)$
- 14 **return** \mathcal{P} and θ

Algorithm 2: Attacking a target node v_i in G

Input : G, v_i, \mathcal{P} , and g_{θ}
Output: A perturbed adjacency matrix A'

- 1 $k^* \leftarrow \arg \max_k g_{\theta}(X, A)_{ik}$ (Selecting the most suitable set of anchor nodes)
- 2 $A' \leftarrow \rho(v_i, \mathbf{p}_{k^*})$ (Conduct perturbation by the selected set of anchor nodes)
- 3 **return** A'

Solving the optimization problem in Eq. (8) often cannot find a good collection of K sets of anchor nodes, because, due to the relaxed constraint $\mathbf{p}_k \in [0, 1]^N$, the entries in each solution $\mathbf{p}_k \in \mathcal{P}$ can be a real value that is far from 0 or 1. Directly quantizing (i.e., binarizing) the entries in \mathbf{p}_k to 0 or 1 causes a large quantization error, thus reduces the quality of the final solution.

To tackle this issue, we propose a **simulated annealing trick** to force the real-valued entries in each solution $\mathbf{p}_k \in [0, 1]^N$ to be close to 0 or 1. This effectively reduces the quantization error when quantizing a solution \mathbf{p}_k to a binary vector in $\{0, 1\}^N$, thus improving the quality of the final solution. Specifically, we develop a **weighted penalty term** to rewrite Eq. (8) as:

$$\min_{\mathcal{P}, \theta} \mathcal{L}(\mathcal{P}, \theta) + \lambda \sum_{\mathbf{p}_k \in \mathcal{P}} \max(\|\mathbf{w}_k \circ \mathbf{p}_k\|_1 - \mathbf{w}_k^{min} \xi, 0) \quad \text{s.t. } \forall \mathbf{p}_k \in \mathcal{P}, \mathbf{p}_k \in [0, 1]^N, \quad (9)$$

where \mathbf{w}_k is an N -dimensional weight vector computed from \mathbf{p}_k , \mathbf{w}_k^{min} is the minimum value of all the entries in \mathbf{w}_k , and \circ is element-wise product. The h -th

entry of \mathbf{w}_k is computed by

$$\mathbf{w}_k^h = \sigma(\mathbf{p}_k^h) = \frac{1}{1 + e^{(\mathbf{p}_k^h - \delta)/T}}, \tag{10}$$

where \mathbf{p}_k^h is the h -th entry in \mathbf{p}_k , δ is the mean of the ξ -th and $(\xi + 1)$ -th largest entry in \mathbf{p}_k , and T is a hyperparameter controlling the ‘‘temperature’’ of the annealing process. A smaller value of T renders the annealing function σ closer to a step function that assigns weights close to 1 to the top- ξ largest entries in \mathbf{p}_k and assigns weights close to 0 to the other entries. Therefore, by gradually decreasing the value of T , we can force the top- ξ largest entries in \mathbf{p}_k closer to 1 and also force the other entries closer to 0. This makes the solution of \mathbf{p}_k close to a binary vector, which reduces the quantization error.

Equation (9) is the **formal form** of our optimization problem, which can be solved by standard proximal gradient descent [25]. A feasible solution to Eq. (9) is also a feasible solution to Eq. (7) due to the following reasons. First, when minimizing the objective function in Eq. (9) following [3, 14, 27], λ is gradually increased to push the penalty terms to zero. This ensures each feasible solution $\mathbf{p}_k \in \mathcal{P}$ to Eq. (9) satisfies the constraint $\|\mathbf{w}_k \circ \mathbf{p}_k\|_1 \leq \mathbf{w}_k^{min} \xi$. Second, as shown by Theorem 1, since $\|\mathbf{w}_k \circ \mathbf{p}_k\|_1 \leq \mathbf{w}_k^{min} \xi$, we have $\|\mathbf{p}_k\|_1 \leq \xi$, which implies \mathbf{p}_k is a feasible solution to Equation (7).

Theorem 1. *For any $\mathbf{p}_k \in [0, 1]^N$, if $\|\mathbf{w}_k \circ \mathbf{p}_k\|_1 \leq \mathbf{w}_k^{min} \xi$, then $\|\mathbf{p}_k\|_1 \leq \xi$.*

Proof. Since $\|\mathbf{w}_k \circ \mathbf{p}_k\|_1 \leq \mathbf{w}_k^{min} \xi$, we have $\xi \geq \|\frac{\mathbf{w}_k}{\mathbf{w}_k^{min}} \circ \mathbf{p}_k\|_1 \geq \|\mathbf{p}_k\|_1$.

Since Eq. (7) is a relaxed version of the original optimization problem in Eq. (6), we can obtain a **final solution** to Eq. (6) by quantizing each feasible solution $\mathbf{p}_k \in \mathcal{P}$ to a binary vector in $\{0, 1\}^N$. We ensure that the final solution quantized from \mathbf{p}_k satisfies the budget of controlled nodes by quantizing the top- ξ largest entries in \mathbf{p}_k to be 1 and the other entries to 0. Since the simulated annealing trick forces the top- ξ largest entries in \mathbf{p}_k closer to 1 and the other entries closer to 0, the quantization error of \mathbf{p}_k is small.

Now, we introduce how to train \mathcal{P} and θ by solving the optimization problem in Eq. (9). Denote by $\mathcal{L}^*(\mathcal{P}, \theta)$ the objective function in Eq. (9), Algorithm 1 summarizes the details to train \mathcal{P} and θ . Line 5 and line 8 are conducting typical gradient steps to update \mathbf{p}_k and θ , respectively, where η_1 and η_2 are learning rates. In line 6, the function $\text{clip}(\mathbf{p}_k, 0, 1)$ is conducting a proximal projection of standard proximal gradient descent [25] to clip each entry in \mathbf{p}_k that is out of the range $[0, 1]$ back to its closest value in $[0, 1]$. Line 10 gradually increases the weight λ of the penalty term. Line 11 conducts the simulated annealing to make the annealing function σ closer to a step function by gradually reducing the value of T . Line 13 quantizes \mathbf{p}_k to a final solution.

After training \mathcal{P} and θ , we can use \mathcal{P} and assignment network g_θ to conduct multifaceted anchor node attacks. Algorithm 2 summarizes the details to attack a target node v_i in G . In line 1, we use g_θ to select the most suitable set of anchor nodes to attack v_i , denoted by \mathbf{p}_{k^*} . In line 2, we use the selected \mathbf{p}_{k^*} to attack

v_i by flipping the edges between v_i and each of the anchor nodes represented by \mathbf{p}_{k^*} . Since g_θ is trained together with \mathcal{P} , it generalizes well to select the most suitable set of anchor nodes, which significantly boosts the success rate of each attack.

Table 1. Model configuration of the assignment network.

Layer	Type	Input Dim.	Weight Dim.	Output Dim.	Activation
1	Graph Convolution	$N \times d$	$d \times 16$	$N \times 16$	ReLU
2	Graph Convolution	$N \times 16$	$16 \times K$	$N \times K$	Softmax

Table 2. Statistics of datasets.

Dataset Statistics	Cora	Citeseer	Facebook	Wiki	Pubmed
#Nodes	2,708	3,327	4,039	2,405	19,717
#Edges	5,278	4,676	88,234	17,981	44,324
#Features	1,433	3,703	1,283	4,973	500
#Classes	7	6	193	17	3

6 Experiments

In this section, we conduct extensive experiments to compare our method with six baseline methods performing adversarial graph structural attacks, such as GUA [37]¹, GUAP [36]², PGD [35], DICE [34], Meta-Self [41] and FGA [7]³. We use the publicly available source code of the baseline methods and we use their default parameter settings in our experiments. Our code is available at the following link⁴.

Experiment Setting. We use the publicly available source code for each of the baseline methods and we use their default parameter settings in our experiments. For our method, we adopt the GCN described in Equation (2) to implement the assignment network g_θ . The model configuration of g_θ is shown in Table 1, where the graph convolution layer performs mean aggregation. The “Input Dim.,” “Weight Dim.” and “Output Dim.” are the dimensions for $H^{(l)}$,

¹ Code: <https://github.com/chisam0217/Graph-Universal-Attack>.

² Code: <https://anonymous.4open.science/r/ffd4fad9-367f-4a2a-bc65-1a7fe23d9d7f/>.

³ Code for PGD, DICE, Meta-Self and FGA: <https://github.com/DSE-MSU/DeepRobust>.

⁴ Code for MFAN: <https://github.com/zhz0108/mfan>.

$W^{(l)}$ and $H^{(l+1)}$ in each layer l , as defined in Equation (1). Following the literature [7, 35–37, 40, 41], we use the classic graph convolutional neural network (GCN) [19] introduced in Sect. 3 as the default victim model to attack. We also use the GAT [31] and Node2Vec [17] as the black-box victim models for transfer attacks. We set $\xi = 5$, $K = 2$ and use the initial value of $\lambda = 0.1$ by default if not specified otherwise. For the training algorithm (Algorithm 1) of our method, we use a batch size of 32, $max_epoch = 120$ and $\eta_1 = 0.01$. The learning rate η_2 is set to 0.2 on the Facebook dataset and 0.005 on the other datasets. The effect of K on the attack effectiveness of our method is analyzed in Appendix A. All experiments are conducted on a server with an NVIDIA RTX 3090 GPU, 64 GB RAM and an Intell(R) Core(TM) i9-10900K CPU.

Datasets. The experiments are performed on five commonly used node classification benchmark datasets listed in Table 2. Cora [26], Citeseer [26] and Pubmed [26] are scientific publication networks, Facebook [21] is a social network, and Wiki [10] is a network of web pages with their hyperlinks as edges. For the largest dataset Pubmed, we sample a subgraph with 2,000 nodes to do the training and use the rest of the nodes as target nodes for testing. The training is performed on the sampled subgraph instead of the complete graph of Pubmed. For the other datasets, we follow the setting of [5, 7, 28, 29, 40], where the training is performed on the entire graph with 20% of the nodes used for training and the rest 80% of the nodes used as target nodes for testing. FGA cannot finish attacking all the testing target nodes on Pubmed within 72 h, thus we cannot report its corresponding results.

Evaluation Metrics. We evaluate the attack performance on misclassification by *foolingratio* (FR), that is,

$$FR = \frac{\# \text{ of misclassified nodes in test set}}{\# \text{ of nodes in test set}} \quad (11)$$

Due to the various attack formats of different baseline methods, we design two measures on the *budget of controlled nodes* to comprehensively evaluate their performance.

The first type of budget, named *budgetpertargetnode* (BPT) and denoted by ξ , is the number of controlled nodes used in attacking a single target node. For GUA and GUAP, since they use the same set of anchor nodes to attack each target node, ξ is the number of anchor nodes. For our method, since we only use one set of anchor nodes to attack each target node, ξ is exactly the budget in Definition 1. For the other baseline methods, ξ is the average number of nodes connected by a perturbed edge to a target node either before or after the attack.

The second type of budget, named *budgetforalltargetnodes* (BFA) and denoted by δ , is the total number of all controlled nodes used to attack all the target nodes in the testing dataset. For GUA and GUAP, $\delta = \xi$. For our method, $\delta = K * \xi$. For FGA, δ is the size of the union of nodes connected to perturbed edges in each attack. For the other baseline methods, δ is the total number of nodes connected to a perturbed edge either before or after the attack.

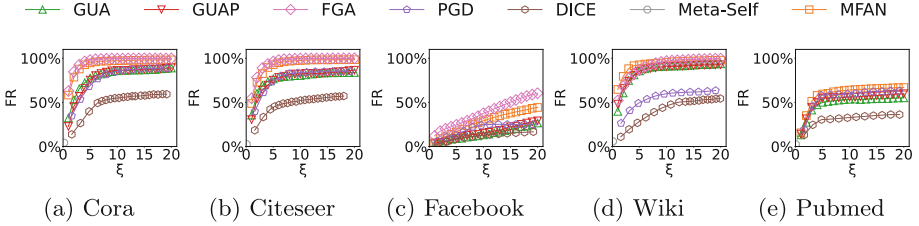


Fig. 2. Fooling ratio (FR) v.s. budget per target node (BPT, ξ).

For each attack method, we also measure its *training time* by the time cost to train an attack model, and we measure its *attacking time* by the time cost of using a trained attack model to conduct an attack on a target node. We discuss the attacking time in Appendix B, and we analyze the training time complexity and the empirical training time in Appendix C.

6.1 Fooling Ratio Under Different Budgets of Controlled Nodes

In this section, we analyze the FR of all compared methods under different budgets of controlled nodes.

Figure 2 shows how the FR of each method changes when using different BPT (i.e., ξ). The FR of all methods increases when ξ increases because a larger BPT allows each attack to perturb more edges, which improves the chance of success. FGA achieves the best FR on Cora, Citeseer and Facebook when using the same BPT as the other methods. Because FGA specifically trains a unique set of controlled nodes to attack each new target node. This significantly improves the successfulness of each attack, however, as illustrated later, it also requires a significantly larger BFA because attacking each new target node requires controlling a new set of nodes. The other baseline methods cannot achieve a comparable FR to FGA because they only use a single set of controlled nodes to conduct each attack, and the constant set of controlled nodes is not specifically trained for each new target node. Interestingly, the proposed MFAN does not specifically train the anchor nodes for each new target node either, but the FR of MFAN

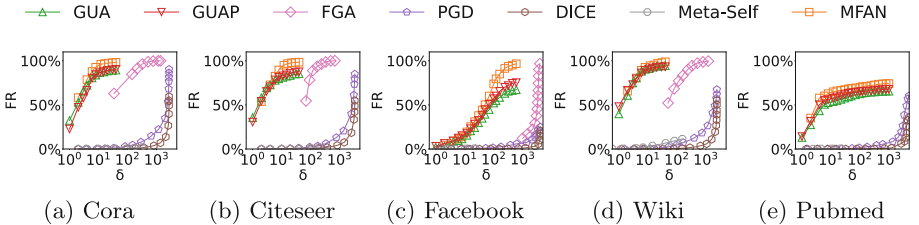


Fig. 3. Fooling ratio (FR) v.s. budget for all target nodes (BFA, δ).

is the closest to FGA on all the datasets. This demonstrates the outstanding performance of MFAN.

While FGA champions the performance when using BPT as the budget of controlled nodes, its FR is dramatically inferior to the anchor nodes attacks (i.e., GUA, GUAP and MFAN) when using BFA as the budget. Figure 3 shows how the FR of each method changes when using different BFA (i.e., δ). We can see that the minimum BFA of FGA, which is achieved by setting $\xi = 1$ for FGA, is much larger than the BFA of GUA, GUAP and MFAN. This is because FGA specifically trains a unique set of controlled nodes to attack each new target node, thus it requires control of a lot of nodes to attack the thousands of target nodes in the testing dataset. The global structural attacks (i.e., DICE, PGD and Meta-Self) also cost a significantly larger BFA than the anchor nodes attacks, because they perturb a large number of edges once and for all the attacks. On the contrary, the anchor nodes attacks are extremely efficient in BFA, because they only use a small constant set(s) of anchor nodes to attack all the target nodes. We can also see that MFAN champions the performance on all datasets due to the effective multifaceted attack that successfully implements the principle of “divide and conquer”.

In summary, when the total number of controlled nodes is limited by the resource (i.e., money, hackers, etc.) to control nodes, MFAN achieves the best FR performance by leveraging the power of “divide and conquer”. Moreover, the attacks conducted by MFAN are also much more stealthy than the other non-anchor nodes attacks due to its small BFA.

6.2 Effectiveness of Transfer Attack

In this section, we evaluate the effectiveness of the transfer attacks by all the compared methods. For each method, we first train its attack model on the white-box victim model f , which is the GCN mentioned in the experiment setting. Then, we apply the trained attack model to attack two other black-box victim models, such as GAT [31] and Node2Vec [17]. The black-box victim models are trained on the same dataset as the white-box victim model.

Table 3 shows the FR of transfer attacks on the five datasets, respectively. The best FR is in bold and the runner-up is underlined. FGA could not produce meaningful results when using small BFA (i.e., δ), thus we cannot report its corresponding results. Meta-Self requires too much memory to run when using the values of ξ in the tables, thus we cannot report its corresponding performance due to out-of-memory issues.

We can see that MFAN achieves the best performance in most cases, which shows the outstanding performance of MFAN in transfer-attacking black-box victim models. We believe such transfer attack performance is due to the following reasons. First, each set of anchor nodes works well in transfer attacks. Since each set of anchor nodes is trained to successfully attack a large group of target nodes, the set of anchor nodes tends to exploit the common defect patterns of many target nodes. Since the defect patterns are commonly carried by many target nodes, they can be learned by a new GNN model trained on the same

Table 3. Fooling ratio of non-transfer attacks and transfer attacks.

Dataset	Model	GCN (white-box, non-transfer)				GAT (black-box, transfer)				Node2Vec (black-box, transfer)			
Cora	Budget	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$
	GUA	77.37%	85.75%	88.83%	89.96%	81.39%	86.19%	88.74%	90.03%	76.92%	84.85%	89.33%	88.60%
	GUAP	75.32%	86.20%	89.25%	90.25%	65.56%	85.78%	86.54%	91.89%	71.64%	85.86%	87.04%	87.11%
	PGD	71.88%	85.33%	0.74%	2.07%	74.72%	83.86%	0.93%	1.80%	72.04%	70.89%	15.53%	15.77%
	DICE	44.30%	54.99%	0.05%	0.10%	51.03%	51.77%	0.11%	0.35%	79.88%	80.24%	15.83%	16.21%
	Meta-Self	-	-	0.02%	0.06%	-	-	0.59%	1.17%	-	-	15.84%	16.04%
	FGA	95.00%	98.63%	-	-	85.44%	92.95%	-	-	62.81%	82.14%	-	-
	MFAN	93.79%	94.30%	94.30%	96.11%	84.67%	93.04%	93.04%	94.59%	85.14%	92.17%	92.17%	96.01%
Citeseer	Budget	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$
	GUA	76.63%	80.35%	83.83%	84.49%	74.63%	81.51%	84.10%	85.12%	86.41%	86.56%	86.70%	86.94%
	GUAP	75.17%	82.20%	86.45%	87.65%	66.87%	79.69%	83.19%	89.24%	76.89%	81.40%	83.18%	84.13%
	PGD	75.67%	83.80%	0.16%	0.34%	77.92%	80.42%	0.64%	1.33%	79.58%	79.45%	37.14%	37.45%
	DICE	45.30%	48.96%	0.03%	0.05%	46.73%	48.90%	0.16%	0.33%	79.06%	79.79%	37.80%	37.89%
	Meta-Self	-	-	0.04%	0.06%	-	-	0.44%	0.72%	-	-	37.22%	38.05%
	FGA	96.27%	98.70%	-	-	84.45%	92.72%	-	-	73.93%	86.29%	-	-
	MFAN	92.37%	97.31%	97.31%	98.45%	90.21%	95.78%	95.78%	97.86%	91.79%	93.09%	93.09%	94.65%
Facebook	Budget	$\xi = 10$	$\xi = 20$	$\delta = 200$	$\delta = 400$	$\xi = 10$	$\xi = 20$	$\delta = 200$	$\delta = 400$	$\xi = 10$	$\xi = 20$	$\delta = 200$	$\delta = 400$
	GUA	13.12%	26.60%	60.44%	66.02%	13.45%	16.63%	50.78%	58.23%	10.24%	15.47%	54.19%	61.77%
	GUAP	15.12%	29.10%	68.69%	73.67%	16.59%	26.68%	64.35%	70.23%	14.96%	20.44%	59.40%	65.23%
	PGD	24.12%	24.96%	3.24%	5.47%	31.16%	33.67%	2.74%	5.45%	16.85%	19.43%	10.25%	10.68%
	DICE	12.99%	16.90%	0.26%	0.66%	15.24%	19.05%	0.74%	1.21%	19.05%	25.93%	10.07%	10.11%
	Meta-Self	-	-	1.33%	2.16%	-	-	1.98%	4.16%	-	-	10.10%	10.33%
	FGA	37.68%	49.40%	-	-	30.33%	49.14%	-	-	21.23%	34.87%	-	-
	MFAN	27.36%	39.53%	88.37%	94.94%	33.49%	38.50%	76.54%	85.53%	38.50%	58.53%	83.39%	87.52%
Wiki	Budget	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$
	GUA	85.58%	90.85%	93.68%	94.35%	42.70%	45.90%	50.06%	56.72%	30.81%	44.99%	54.47%	74.18%
	GUAP	85.40%	90.65%	92.73%	93.11%	39.85%	48.76%	50.81%	66.59%	21.87%	43.08%	51.52%	67.44%
	PGD	52.01%	60.01%	0.81%	2.06%	56.37%	67.92%	0.68%	1.11%	42.55%	46.15%	14.74%	15.20%
	DICE	30.78%	44.94%	0.13%	0.35%	39.58%	63.42%	0.26%	0.31%	46.63%	62.37%	14.64%	15.45%
	Meta-Self	-	-	4.66%	6.69%	-	-	0.70%	1.27%	-	-	15.10%	15.71%
	FGA	86.74%	96.84%	-	-	52.81%	63.24%	-	-	44.23%	72.45%	-	-
	MFAN	92.55%	95.52%	95.52%	97.82%	62.93%	67.07%	67.07%	73.31%	47.78%	54.97%	54.97%	82.70%
Pubmed	Budget	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$	$\xi = 5$	$\xi = 10$	$\delta = 20$	$\delta = 40$
	GUA	50.04%	53.50%	55.11%	59.42%	47.33%	48.54%	51.03%	55.45%	41.00%	43.27%	44.36%	48.31%
	GUAP	54.97%	56.13%	59.88%	63.14%	49.98%	56.97%	60.08%	65.30%	45.19%	47.33%	50.87%	54.33%
	PGD	58.02%	59.47%	0.20%	0.33%	58.06%	63.44%	1.02%	1.99%	47.59%	52.98%	20.14%	24.77%
	DICE	30.46%	32.55%	0.03%	0.05%	36.68%	38.40%	0.73%	0.92%	32.32%	37.65%	18.00%	18.78%
	Meta-Self	-	-	0.26%	0.38%	-	-	0.80%	1.14%	-	-	19.56%	22.17%
	FGA	-	-	-	-	-	-	-	-	-	-	-	-
	MFAN	59.48%	64.04%	64.04%	66.88%	60.09%	62.95%	62.95%	65.63%	49.64%	51.34%	51.34%	53.34%

dataset. In this case, the anchor nodes are still effective to transfer-attack the new model. Second, the assignment network also works well in transfer attacks. Since the input graph does not change, the output of the assignment network will not change. Therefore, the same set of anchor nodes will still be selected to attack the target node when performing transfer attacks. If the same defect pattern of the target node is learned by a new GNN model, the selected set of anchor nodes can successfully transfer-attack the target node.

Table 4. FR of standard MFAN (ST) and ablated MFAN (AB).

Dataset	Model	GCN (white-box, not transfer)			GAT (black-box, transfer)			Node2Vec (black-box, transfer)		
		$\xi = 5$	$\xi = 10$	$\xi = 20$	$\xi = 5$	$\xi = 10$	$\xi = 20$	$\xi = 5$	$\xi = 10$	$\xi = 20$
Cora	MFAN (AB)	78.32%	83.43%	83.46%	75.15%	82.41%	83.60%	50.96%	69.34%	77.98%
	MFAN (ST)	93.79%	94.30%	96.11%	84.67%	93.04%	94.59%	85.14%	92.17%	96.01%
Citeseer	MFAN (AB)	77.41%	80.92%	81.70%	75.86%	80.40%	80.61%	80.16%	81.61%	82.81%
	MFAN (ST)	92.37%	97.31%	98.45%	90.21%	95.78%	97.86%	91.79%	93.09%	94.65%
Facebook	MFAN (AB)	13.33%	21.51%	29.46%	15.05%	27.81%	32.21%	12.98%	23.64%	29.76%
	MFAN (ST)	16.25%	27.36%	39.53%	18.22%	33.49%	38.50%	20.45%	38.50%	58.53%
Wiki	MFAN (AB)	75.77%	85.55%	90.48%	57.99%	63.72%	70.95%	44.49%	50.28%	78.05%
	MFAN (ST)	92.55%	95.52%	97.82%	62.93%	67.07%	73.31%	47.78%	54.97%	82.70%
Pubmed	MFAN (AB)	46.67%	48.26%	52.99%	40.26%	42.97%	45.94%	40.47%	44.70%	47.66%
	MFAN (ST)	59.48%	64.04%	66.88%	60.09%	62.95%	65.63%	49.64%	51.34%	53.34%

6.3 The Effects of g_θ , λ and Simulated Annealing

In this subsection, we discuss the effects of the assignment network g_θ , the hyperparameter λ and the simulated annealing trick.

The Effect of the Assignment Network g_θ . To investigate the effect of g_θ , we compare the FR of the standard MFAN using the assignment network and an ablated MFAN that selects a set of anchor nodes uniformly at random. Both methods use the same sets of anchor nodes trained by the standard MFAN. As shown in Table 4, the FR of the standard MFAN is much better than the ablated MFAN, which shows the effectiveness of the assignment network.

The Effect of λ . We analyze the effect of λ by comparing the performance of MFAN when using different *growth rates*, denoted by v , which is the multiplying factor on λ when increasing its value in line 10 of Algorithm 1. As shown in Fig. 4, the loss curves when training MFAN are comparable when using different growth rates. This means the training of MFAN is stable with respect to the growth rate of λ . If we zoom in Fig. 4(c), we can see the SPT with a larger v drops slightly faster than the SPT with a smaller v . This is because a larger v increases λ at a faster speed, which pushes the SPT faster towards zero. However, since the effect of v on the training of MFAN is not significant, it does not affect the FR of MFAN very much. As a result, we can see in Table 5 that the FR of MFAN are comparable when using different values of v .

The Effect of the Simulated Annealing Trick. To show the effect of the simulated annealing trick, we analyze the FR and the quantization error of the perturbation vectors produced by two versions of MFAN. One version is the standard MFAN that solves Eq. (9), where the simulated annealing trick is applied; the other version is an ablated MFAN that solves Eq. (8), which does not apply the simulated annealing trick. The quantization error (QE) is measured by

$$QE = \sum_{\mathbf{p}_k \in \mathcal{P}} \|\mathbf{p}_k - \phi(\mathbf{p}_k, \xi)\|_1, \tag{12}$$

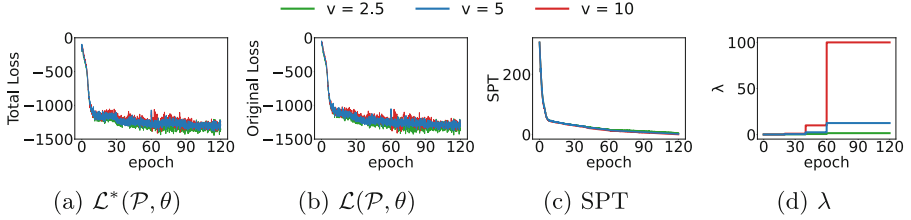


Fig. 4. The curves of the total loss $\mathcal{L}^*(\mathcal{P}, \theta)$, the original loss $\mathcal{L}(\mathcal{P}, \theta)$, the sum of penalty terms (SPT) $\sum_{\mathbf{p}_k \in \mathcal{P}} \max(\|\mathbf{w}_k \circ \mathbf{p}_k\|_1 - \mathbf{w}_k^{\min} \xi, 0)$ and λ on Cora dataset. Each subfigure shows three curves using different growth rates.

Table 5. FR of MFAN when using different growth rates $v \in \{2.5, 5.0, 10.0\}$.

Dataset	$\xi = 5$			$\xi = 10$			$\xi = 20$		
	$v = 2.5$	$v = 5.0$	$v = 10.0$	$v = 2.5$	$v = 5.0$	$v = 10.0$	$v = 2.5$	$v = 5.0$	$v = 10.0$
Cora	93.69%	93.79%	93.97%	94.27%	94.30%	94.19%	95.85%	96.11%	95.85%
Citeseer	91.79%	92.37%	92.34%	97.24%	97.31%	97.33%	98.45%	98.45%	98.38%
Facebook	15.01%	15.55%	15.14%	25.85%	25.97%	25.66%	38.29%	38.84%	38.84%
Wiki	91.26%	92.55%	92.62%	94.61%	95.52%	95.33%	97.79%	97.82%	97.62%
Pubmed	59.48%	59.48%	58.98%	63.97%	64.04%	63.77%	66.79%	66.88%	66.31%

Table 6. Effect of simulated annealing on fooling ratio and quantization error.

Dataset	Model	Fooling ratio (FR) \uparrow			Quantization error (QE) \downarrow		
		$\xi = 5$	$\xi = 10$	$\xi = 20$	$\xi = 5$	$\xi = 10$	$\xi = 20$
Cora	Ablated MFAN (w/o simulated annealing)	91.44%	92.53%	94.99%	8.68	16.01	28.42
	Standard MFAN (with simulated annealing)	93.79%	94.30%	96.11%	1.37	2.98	4.07
Citeseer	Ablated MFAN (w/o simulated annealing)	90.76%	96.02%	96.88%	16.77	29.10	47.77
	Standard MFAN (with simulated annealing)	92.37%	97.31%	98.45%	1.57	3.12	7.72
Facebook	Ablated MFAN (w/o simulated annealing)	14.88%	25.47%	38.12%	51.21	104.45	172.56
	Standard MFAN (with simulated annealing)	16.25%	27.36%	39.53%	15.30	27.29	34.84
Wiki	Ablated MFAN (w/o simulated annealing)	90.69%	94.94%	96.75%	8.24	12.51	22.00
	Standard MFAN (with simulated annealing)	92.55%	95.52%	97.82%	2.20	2.61	4.61
Pubmed	Ablated MFAN (w/o simulated annealing)	57.66%	60.98%	63.55%	9.84	20.14	27.10
	Standard MFAN (with simulated annealing)	59.48%	64.04%	66.88%	2.85	3.43	5.01

where $\phi(\mathbf{p}_k, \xi)$ outputs the quantized \mathbf{p}_k with the top- ξ largest entries being 1 and other entries being 0. As shown in Table 6, the standard MFAN achieves a smaller QE and a larger FR than the ablated MFAN. This demonstrates the effectiveness of the simulated annealing trick in reducing the quantization error and improving attack performance.

7 Conclusion

In this paper, we proposed and tackled a novel problem named multifaceted anchor nodes attack. The key idea is to simultaneously train multiple sets of anchor nodes together with an assignment network. Each set of anchor nodes is specialized in successfully attacking a different set of target nodes, and the assignment network accurately selects the best suitable set of anchor nodes to attack a new target node. In this way, we implement the mechanism of “divide and conquer” to successfully attack the union of the nodes that are attacked by each set of anchor nodes. Since the same sets of anchors are used to attack all the target nodes, our method is extremely budget-efficient, which only requires controlling a very small number of nodes to achieve outstanding attack performance.

Acknowledgements. This work was supported in part by the NSERC Discovery Grant program (RGPIN-2022-04977) and in part by the start-up funding of McMaster University. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

1. Bojchevski, A., Günnemann, S.: Adversarial attacks on node embeddings via graph poisoning. In: ICML, pp. 695–704 (2019)
2. Bose, A.J., Cianflone, A., Hamilton, W.L.: Generalizable adversarial attacks with latent variable perturbation modelling. [arXiv:1905.10864](https://arxiv.org/abs/1905.10864) (2019)
3. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)
4. Cao, Q., Shen, H., Gao, J., Wei, B., Cheng, X.: Popularity prediction on social platforms with coupled graph neural networks. In: WSDM, pp. 70–78 (2020)
5. Chang, H., et al.: A restricted black-box adversarial framework towards attacking graph embedding models. In: AAAI, pp. 3389–3396 (2020)
6. Chang, J., et al.: Sequential recommendation with graph neural networks. In: ACM SIGIR, pp. 378–387 (2021)
7. Chen, J., Wu, Y., Xu, X., Chen, Y., Zheng, H., Xuan, Q.: Fast gradient attack on network embedding. [arXiv:1809.02797](https://arxiv.org/abs/1809.02797) (2018)
8. Chen, Y., Ye, Z., Zhao, H., Wang, Y., et al.: Feature-based graph backdoor attack in the node classification task. *Inter. J. Intell. Syst.* (2023)
9. Cheung, M., Moura, J.M.: Graph neural networks for covid-19 drug discovery. In: IEEE International Conference on Big Data, pp. 5646–5648 (2020)
10. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: EMNLP-CoNLL, pp. 708–716 (2007)
11. Dai, E., Lin, M., Zhang, X., Wang, S.: Unnoticeable backdoor attacks on graph neural networks. In: WWW, pp. 2263–2273 (2023)
12. Dai, H., et al.: Adversarial attack on graph structured data. In: ICML, pp. 1115–1124 (2018)
13. Fan, W., et al.: Graph neural networks for social recommendation. In: WWW, pp. 417–426 (2019)

14. Fiacco, A.V., McCormick, G.P.: *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM (1990)
15. Fukushima, K.: Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Trans. Syst. Sci. Cybernet.* **5**(4), 322–333 (1969)
16. Geisler, S., Schmidt, T., Şirin, H., Zügner, D., Bojchevski, A., Günnemann, S.: Robustness of graph neural networks at scale. In: *NeurIPS*, pp. 7637–7649 (2021)
17. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: *ACM SIGKDD*, pp. 855–864 (2016)
18. Ju, M., Fan, Y., Zhang, C., Ye, Y.: Let graph be the go board: gradient-free node injection attack for graph neural networks via reinforcement learning. In: *AAAI*, pp. 4383–4390 (2023)
19. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: *ICLR* (2017)
20. Kumar, S., West, R., Leskovec, J.: Disinformation on the web: impact, characteristics, and detection of wikipedia hoaxes. In: *WWW*, pp. 591–602 (2016)
21. Leskovec, J., Mcauley, J.: Learning to discover social circles in ego networks. In: *NeurIPS*, pp. 539–547 (2012)
22. Li, K., Liu, Y., Ao, X., He, Q.: Revisiting graph adversarial attack and defense from a data distribution perspective. In: *ICLR* (2023)
23. Liu, Z., Luo, Y., Wu, L., Liu, Z., Li, S.Z.: Towards reasonable budget allocation in untargeted graph structure attacks via gradient debias. In: *NeurIPS*, pp. 27966–27977 (2022)
24. Min, S., Gao, Z., Peng, J., Wang, L., Qin, K., Fang, B.: Stgsn—a spatial-temporal graph neural network framework for time-evolving social networks. *Knowl.-Based Syst.* **214**, 106746 (2021)
25. Rockafellar, R.T.: *Convex analysis*. Princeton University Press, Princeton, N. J., Princeton Mathematical Series (1970)
26. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. *AI Mag.* **29**(3), 93–93 (2008)
27. Smith, A.E., Coit, D.W., Baeck, T., Fogel, D., Michalewicz, Z.: Penalty functions. *Handbook of evolutionary computation* (1997)
28. Sun, Y., Wang, S., Tang, X., Hsieh, T.Y., Honavar, V.: Non-target-specific node injection attacks on graph neural networks: a hierarchical reinforcement learning approach. In: *WWW*, pp. 673–683 (2020)
29. Takahashi, T.: Indirect adversarial attacks via poisoning neighbors for graph convolutional networks. In: *IEEE International Conference on Big Data*, pp. 1395–1400 (2019)
30. Tao, S., Cao, Q., Shen, H., Huang, J., Wu, Y., Cheng, X.: Single node injection attack against graph neural networks. In: *CIKM*, pp. 1794–1803 (2021)
31. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: *ICLR* (2018)
32. Wang, B., Pang, M., Dong, Y.: Turning strengths into weaknesses: a certified robustness inspired attack framework against graph neural networks. In: *CVPR*, pp. 16394–16403 (2023)
33. Wang, X., Cheng, M., Eaton, J., Hsieh, C.J., Wu, S.F.: Fake node attacks on graph convolutional networks. *J. Comput. Cognitive Eng.* **1**(4), 165–173 (2022)
34. Waniek, M., Michalak, T.P., Wooldridge, M.J., Rahwan, T.: Hiding individuals and communities in a social network. *Nat. Hum. Behav.* **2**(2), 139–147 (2018)
35. Xu, K., et al.: Topology attack and defense for graph neural networks: an optimization perspective. In: *IJCAI*, pp. 3961–3967 (2019)

36. Zang, X., Chen, J., Yuan, B.: Guap: Graph universal attack through adversarial patching. [arXiv:2301.01731](https://arxiv.org/abs/2301.01731) (2023)
37. Zang, X., Xie, Y., Chen, J., Yuan, B.: Graph universal adversarial attacks: a few bad actors ruin graph learning models. In: IJCAI, p. 3328–3334 (2020)
38. Zhang, B., Dong, Y., Chen, C., Zhu, Y., Luo, M., Li, J.: Adversarial attacks on fairness of graph neural networks. In: ICLR (2024)
39. Zou, X., et al.: Tdgia: effective injection attacks on graph neural networks. In: ACM SIGKDD, pp. 2461–2471 (2021)
40. Zügner, D., Akbarnejad, A., Günnemann, S.: Adversarial attacks on neural networks for graph data. In: ACM SIGKDD, pp. 2847–2856 (2018)
41. Zügner, D., Günnemann, S.: Adversarial attacks on graph neural networks via meta learning. In: ICLR (2019)



Causal Attentive Group Recommendation

Liancheng Xu, Xiaoqi Wu, Xiaoxiang Wang, and Xinhua Wang^(✉)

School of Information Science and Engineering, Shandong Normal University,
Jinan 250358, China

lxcxu@sdu.edu.cn, {2021021010,2021317083}@stu.sdu.edu.cn,
17860546560@163.com

Abstract. The demand for group recommendations in the field of recommendation systems is steadily increasing. In group recommendation, how to accurately aggregate the preferences of group members to infer group decisions has become the core issue. Currently, various deep learning methods are applied to group recommendation problems. Among them, attention based methods dynamically aggregate group member preferences by distinguishing the importance of different members, which greatly improves group recommendation performance. However, attention mechanism methods cannot avoid the negative impact of potential confounding factors. That is to say, the correlation between group members and the learned candidate projects cannot accurately reflect the impact of group members on the group recommendation results, leading to false correlation. This affects the accuracy of group representation learning. To tackle this challenge, the paper introduces a model named Causal Attentive Group Recommendation(CAGR). This model incorporates causal inference within an attention network to tailor the group representation, effectively addressing the problem of capturing erroneous correlations. Building upon the potential outcome framework, CAGR utilizes the concept of individual treatment effects (ITE) to quantify the causal relationship between each group member and the outcome. Our objective is to capture the authentic influence of group members on the desired outcome. To integrate causal insights into the group representation learning process, we introduce regularization that aligns the distance between the ITE of group members and the conventional attention weights, correct the importance of group members, and obtain a more accurate causal correlation between group and group members. Comprehensive experiments conducted on two authentic datasets validate the superiority of our proposed model in the realm of group recommendation.

Keywords: Group recommendation · Causal inference · Attention mechanism

1 Introduction

In the context of the information age, due to the surge in group activities, users with shared interests and goals now form various online groups[1]. Traditional

recommendation systems tailored for individuals are no longer sufficient to cater to the preferences of these collectives, thereby fostering the emergence of the notion of group recommendation[2][3]. Group recommendation entails aggregating the preferences of all group members to deliver tailored recommendations that align with the group’s contentment[4]. Group recommendation sets itself apart from traditional individual recommendation systems by involving decision-making, rendering the process considerably intricate. Because each member of the group plays a distinct role and contributes differently, leading to varying impacts on the final decision.

With the application of deep learning in personal recommendation, significant advancements have been made [5]. Researchers are leveraging various components of deep neural networks to simulate the group’s decision-making process [6]. In current research, some neural attention methods [2,7,8] are demonstrating superior performance. Attention-based approaches offer the ability to discern the importance of the item’s features, enabling efficient aggregation of group members’ preferences in the context of group recommendation. In group recommendation, the relevance captured by the attention mechanism is derived from calculating the co-occurrence frequency of each member with the candidate items recommended to the group [2]. For instance, in a group decision regarding travel destinations, a group member frequently journeys to China. Consequently, this user would carry a significant weight in the destination decision process regarding China. However, this approach comes with certain limitations. Firstly, although numerous innovative neural network architectures have been introduced for group representation learning (GRL), they still struggle to adequately capture the collective item preferences of a group. This limitation arises from potential confounding factors, and the correlation captured by the attention mechanism doesn’t accurately represent the genuine impact of group members on the target outcome, that is, spurious correlation [9]. For example, in the group decision-making process of choosing a tourist destination mentioned above. If a member frequently visits China for work reasons or only passes through China, such as flight transfers, then the member’s understanding of China is very limited, but the attention mechanism model will also assign higher weights to that group member. It is obvious that simple co-occurrence frequencies cannot identify this issue and lead to biased recommendation results. Therefore, it is crucial to identify causal relationships and assign appropriate weights to each member in the decision-making process.

To tackle the aforementioned challenges, this paper presents a novel model known as the CAGR, which effectively identifies causal relationships from observational data by incorporating a potential outcome framework and calculates weights accordingly. Causal inference has been used in recommendation systems [10] to tackle potential confounding factors that can affect recommendation performance. Inspired by individual recommendation, causal inference can also help in identifying causal relationships within groups. By leveraging the causal inference framework, this paper enhances the attention mechanism in the group recommendation model with the goal of better learning the significance of group

members' weights for improved group recommendation outcomes. Specifically, we first obtain the attention weights of group members through an attention mechanism network to describe their importance. Then, based on the potential outcome framework, the concept of individual treatment effectiveness (ITE) [11] is used to accurately measure the causal relationship between group members and expected outcomes. ITE involves evaluating causal relationships (known as interventions) by calculating the differences in outcomes when relevant group members are actively retained and deleted. This generates an ITE vector, where each component reflects the causal impact of the member on the target, which is another important variable. Finally, in order to incorporate causal reasoning into group recommendations, we use various distance functions to align the obtained group ITE vectors with attention weights, minimizing the distance between them. This corrects the importance of group members and obtains an accurate correlation between group members and recommendation results. In summary, the main contributions of this study are as follows:

- This paper pioneers the integration of causal inference into group recommendation. By utilizing the potential outcome framework, the group recommendation task undergoes reconstruction, effectively eliminating potential confounding factors and revealing the causal relationships within group recommendation.
- The Individual Treatment Effect (ITE) is employed to infuse causality into the conventional attention mechanism, and the group members are aggregated to realize group representation learning and improve group recommendation performance.
- We propose the CAGR and extensively evaluate its effectiveness through experiments on two real datasets, showing significant improvement over previous methods.

This paper describes related work in Sect. 2, introduces the potential outcome framework in Sect. 3, elaborates on the CAGR model in Sect. 4, in Sect. 5, the experimental results that validate the effectiveness of the model are presented, and the paper concludes in Sect. 6.

2 Related Work

In this section, we delve into two relevant aspects of our study: group recommendation and causal recommendation.

2.1 Group Recommendation

Group recommendation methods can be classified into two main categories: memory-based and model-based approaches. The first method aggregate group members' preferences without accounting for their interactions. For example, the average strategy (AVG) calculates the group's overall preference score by averaging the individual preference scores of its members [12]. The maximum satisfaction strategy (MS) ranks group members according to their scores

and then calculates the average score of the top n members to represent the group preference[13]. However, this approach ignores other group members. Similar strategies include the least misery strategy (LM) and others. Model-based approaches emphasize modeling the decision-making process and internal interactions among group members. These models can be classified into two groups: conventional approaches and deep learning-based techniques. Conventional approaches use probabilistic models [14], game theory [15], ect, to model and generate recommendation results. Deep learning-based methods utilize various deep neural networks for modeling purposes [6]. Recently, the effectiveness of attention mechanisms in personal recommendations has sparked the application of attention mechanisms in group recommendations and produced positive effects [2,8]. These models account for the dynamic impact of group members. It's worth noting that AGREE [2] pioneers the use of the attention mechanism to enhance feature representations of users and groups, fostering improved interactions between them. MoSAN [8] considers the social influence of individuals on others and utilizes sub-attention neural networks to model user interactions. However, due to the potential confounding factors, the representation captured by the attention mechanism might not accurately reflect the correlation between the responding member and the target.

2.2 Causal-Based Recommendation

The causal inference focuses on how to eliminate confounding bias [16]. Causal inference has been recently introduced into recommender systems to eliminate various biases[17] and improve the performance of recommendation systems such as popularity bias [18], clickbaitbias [19] and Matthew effect [20]. These efforts can be categorized into two main groups. One category is counterfactual. Mehrotra et al. [21] used a quasi-experimental Bayesian framework to generate counterfactual data to evaluate the impact of treatment on outcomes. Yuan et al. [22] learns with a small amount of unbiased data generates labels for unobserved data. Wang et al. [19] utilizes counterfactual reasoning to estimate causal effects and solve clickbait problems. Another type is to consider confounding effects in recommendation. Wang et al. [23] suggests a method that utilizes causal relation to improve the recommendation process. Wang et al. [24] focuses on actual user interests influenced by unobserved confounding factors, utilizing a de-confounding technique with linear models for learning. Sato et al. [25] explored the causal effects of recommendations, taking user and item attributes as confounding variables, and addressing confounding challenges through sample reweighting. In this paper, we consider the confounding factors brought about by group members and use ITE to estimate the causal relationship between each member and the outcome.

3 Preparation

In this section, we will provide a concise introduction to the fundamental principles of the Potential Outcome Framework (POF) [11]. The potential outcome

framework comprises three basic elements: unit (I), treatment (T), and outcome (Y). Causality is bound to the intervention, acting on units. The influence of such interventions is assessed by comparing potential outcomes resulting from different interventions, thus determining the causal impact[9]. The unit stands as the smallest object in the study, the treatment is the operation performed on the object of study, and the outcome obtained by the intervention acting on the unit is called the potential outcome. If we translate this into the context of recommendations. In that way, the unit symbolizes the user under consideration for recommendation, the treatment represents the suggested item, and the outcome corresponds to the item’s score. The score indicating the user’s affinity for the item. Overall, the potential outcomes framework provides a concrete formula for describing causal relationships from observed data.

In this paper, the concept of Individual Treatment Effect (ITE) is employed, which pertains to individual units. Its definition is provided below:

$$\text{ITE}_i = Y_i(T = 1) - Y_i(T = 0), \quad (1)$$

In formula 1, $Y_i(T = 1)$ and $Y_i(T = 0)$ stand for the potential outcomes of unit i under different treatment Y . Under the assumption of negligible, positive and stable unit treatment values (SUTVA), $Y_i(T = t)$, $t \in \{0, 1\}$, can be re-obtained from the observed data through $Y_i(T)$ [9]. For a more detailed understanding of the potential outcomes framework, please refer to [11].

4 Causal Group Recommendation Framework

In this section, we start by formally defining the group recommendation problem to be addressed. Subsequently, we present the proposed model in two steps: 1) outlining the manner in which the group recommendation task is structured using the potential outcome framework; and 2) offering a detailed introduction to group representation learning for causal inference.

4.1 Problem Formulation

Let’s assume there is a collection of users denoted as $\mathcal{U} = \{u\}$, a collection of items identified as $\mathcal{I} = \{i\}$, and a collection of groups represented by $\mathcal{G} = \{g\}$. Each group is comprised of users, for example, $g = \{u_1, u_2, \dots, u_n\}$, where n represents the size of group and $u \in \mathcal{U}$. We have two types of observable interaction data in $\mathcal{U}, \mathcal{G}, \mathcal{I}$: user-item interactions denoted as \mathbf{Y} , and group-item interactions denoted as \mathbf{R} . In general, our aim is to provide recommendations for a target group g by suggesting a list of potentially interesting items. This objective is formally defined as [2] follows:

Input: Group, user, item, and group-item and user-item interactions correspond to $\mathcal{G}, \mathcal{U}, \mathcal{I}$, and \mathbf{R}, \mathbf{Y} , respectively. They are all one-dimensional variables with a length of 32. After concatenating the Embedding Layer layers, a two-dimensional variable representation of [32,6] is obtained.

Output: The personalized sorting function f associates an item with each group, $f_g(g, i) \rightarrow \mathbb{R}$.

4.2 Potential Outcome Framework for Group Recommendation

Conventional group recommendation involves assigning a fixed value to each group member and subsequently aggregating their preferences to create a group representation. However, this approach lacks dynamic adjustment of member weights. Consequently, these methods cannot flexibly learn group preferences, and cannot be more accurate for group recommendation. Drawing inspiration from the neural attention mechanism [26], which captures the significance of various elements through data-driven learning. However, we observe that the weights acquired via the attention mechanism inevitably contain errors, with some failing to align with causal effects. In this paper, we leverage the potential outcome framework to influence the attention mechanism, thereby enhancing the precision of attention mechanism learning. This approach successfully enhances the efficiency of group recommendation.

In group recommendation, for every group, we utilize a member vector denoted as $\mathbf{f}_g \in \mathbb{R}^U$, where U signifies the total count of members within the system. In \mathbf{f}_g , each element indicates whether the group member is considered for the group, with values of either 1 or 0. In the group-item interaction set $\mathbf{R} = \{r_{gi}\}$, r_{gi} takes the values 0 or 1, representing negative and positive samples. We treat the potential outcome framework as a group recommendation task, the group as a unit, the group member vector \mathbf{f}_g represents a treatment, and the outcome pertains to whether the group interacts with the item. In other words, our investigation the impact of group members on group-item interaction. According to the varying influence of the members within the group, we assign weights to these members, and then make recommendations for the group. For a given group-item pair (g, i) , consider the set of accessible members for group g as \mathbf{z}_g , then the ITE of a group member $u \in \mathbf{z}_g$ (i.e., $\mathbf{f}_{gu} = 1$) can be computed as follows:

$$\beta_{gi}^u = \text{ITE}_u = p(r_{gi} = 1 | g, \mathbf{f}_g) - p(r_{gi} = 1 | g, \mathbf{f}_g^{-u}), \quad (2)$$

In formula 2, \mathbf{f}_g^{-u} indicates removing the member vector of group member u , that is, $\mathbf{f}_{gu} = 0$, as an intervention style. $p(r_{gi} = 1 | g, \mathbf{f}_g)$ and $p(r_{gi} = 1 | g, \mathbf{f}_g^{-u})$ signify the interaction probabilities, considering the inclusion or exclusion of group member u , correspondingly. This formula delineates the causal connection between group member u and the ultimate group-item interaction probability, signifying the extent of group members' significance in the recommended outcome for the group. The larger β_{gi}^u is, the more important the group member u is for the final prediction result.

4.3 Group Representation Learning

This paper employs a representation learning framework to tackle the challenge of group recommendation. Denote the embedding vectors of user u and item i as \mathbf{e}_u and \mathbf{v}_i , respectively. Our objective is to obtain the representation of a group g , represented as the embedding vector \mathbf{s}_g , that encapsulates the group's

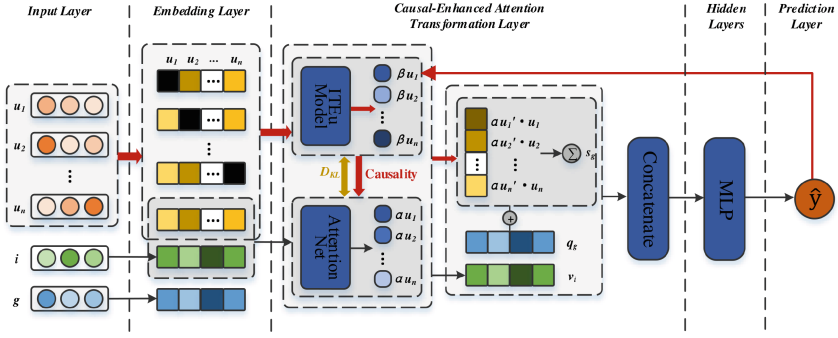


Fig. 1. The CAGR overall framework. The black squares in the Embedding Layer represent the intervened users.

preferences. We aggregate the preferences of all members within the group, considering each member’s input, and merge it with the group’s inherent preference representation \mathbf{q}_g to obtain the ultimate group preference[2]. The formula of the group representation as follows:

$$\mathbf{s}_g = \mathbf{q}_g + \sum_{u \in \mathbf{z}_g} w_u^F (\alpha_{gi}^u \cdot \mathbf{e}_u), \tag{3}$$

In formula 3, $\mathbf{w}^F = \{w_u^F\} \in \mathbb{R}^U$ is a weight parameter, and $\alpha_{gi} = \{\alpha_{gi}^u\} \in \mathbb{R}^{|\mathbf{z}_g|}$ represents a learned parameter, signifying the attention weight assigned to each member in the group. The magnitude of this weight dictates the impact of a group member on the entire group, where a larger weight corresponds to a more influential position within the group. Additionally, the status of group members within the group depends on their past interactions. If a member interacts more frequently with item i than other members, we believe that the member has a better understanding of item i and his opinions will be more informative. Therefore, we consider group members with higher weights to be more important for this group. Calculate α_{gi}^u as follows:

$$\alpha_{gi}^u = \text{softmax} (\text{ReLU} (\mathbf{w}^T [\mathbf{e}_u, \mathbf{v}_i] + b)), \tag{4}$$

In formula 4, $\mathbf{w} \in \mathbb{R}^{2u}$ represents the weight, $b \in \mathbb{R}$ is the deviation parameter, and ReLU serves as the nonlinear activation function.

Our model is illustrated in Fig. 1. $y(g, i)$ is defined as the predictive function for group preferences towards items as follows:

$$y(g, i) = \underbrace{\sigma(h(\dots(h([\mathbf{s}_g, \mathbf{v}_i])))}_{l}, \tag{5}$$

In formula 5, $[\cdot, \cdot]$ denotes the concatenation operation, $\sigma(x)$ represents the sigmoid function, and h denotes the multi-layer perceptron (MLP) with a total of l layers.

4.4 GRL Based on Causal Inference

This paper employs a regression-based pairwise loss function [27]. This loss function ensures that observed interactions are given more significance than unobserved interactions when predicting scores. Then, the optimization of y can be represented as:

$$\mathcal{L}_p = \sum_{(g,i,t) \in \mathcal{O}} (y_{git} - \hat{y}_{git})^2, \quad (6)$$

In formula 6, \mathcal{O} represents the training set, with each instance (g, i, t) signifies the interaction between group g and item i (i.e. positive instance), but not yet with item t (i.e. negative instance). $\hat{y}_{git} = \hat{y}_{gi} - \hat{y}_{gt}$ refers to the predicted range of observed interactions (g, i) and unobserved interactions (g, t) . This paper primarily addresses implicit feedback, which means that if an observed group-item interaction is represented as 1 and an unobserved group-item interaction is 0.

Obviously, based on the co-occurrence frequency of each member with the item, the attention weight α_{gi} captures how important the group member is to the target. To explore causal relationships, it's logical to establish a connection between α_{gi} and β_{gi} . β_{gi} represents group member ITE, which evaluates goal change by active intervention group members. Therefore, this paper introduces a regularization factor [9] to minimize the distance between α_{gi} and β_{gi} . We first regularize β_{gi} :

$$\eta_{gi}^u = \frac{\exp\left(\frac{\beta_{gi}^u}{\rho}\right)}{\sum_{v \in z_g} \exp\left(\frac{\beta_{gi}^v}{\rho}\right)}, \quad (7)$$

In formula 7, ρ is a parameter controlling distribution sharpness. When $\rho \rightarrow \infty$, then $\eta_{gi}^u = \frac{1}{|z_g|}$, and all group members are treated equally. When $\rho \rightarrow 0$, η_{gi}^u is 1 for the group member with the largest β_{gi}^u , then η_{gi}^u is 0 for all other group members. The following is the loss formula between α_{gi} and β_{gi} :

$$\mathcal{L}_c = \sum_{(g,i) \in \mathcal{R}} (\alpha_{gi}, \eta_{gi}) + a \|\alpha_{gi}\|_1, \quad (8)$$

In formula 8, $\eta_{gi} = \{\eta_{gi}^u\} \in \mathbb{R}^{|z_g|}$. Because real-life group decision-making processes are often controlled by a small subset of group members. Therefore, we use the L1 norm to motivate α_{gi} , and (\cdot, \cdot) is the distance function. In the experimental section, we discuss several common distance functions to implement (\cdot, \cdot) , and study its influence and choose the one with the best effect. The overall optimization loss for y is outlined as follows:

$$\mathcal{L} = \mathcal{L}_p + \mathcal{L}_c. \quad (9)$$

5 Experiments

In this section, we conducted extensive experiments on two datasets to validate the effectiveness of our model and answered the following questions:

RQ1: How effective is our designed Causal Attentive Group Recommendation model (CAGR)? Can they provide better group recommendation performance?

RQ2: How does causal inference affect the performance of group recommendation models?

RQ3: How do different hyper-parameter settings affect our model?

RQ4: How do different distance function settings affect our model?

5.1 Experimental Settings

1. Datasets

The two chosen datasets in this study have been adopted from prior research [2, 7]: Mafengwo¹ and CAMRa2011², both originating from the real world.

- **Mafengwo:** This is an event-based dataset obtained through web crawling of a tourism website. The site offers group travel, and each group consists of a minimum of 2 members.
- **CAMRa2011:** This is a family movie-watching recommendation dataset. Families can be regarded as groups. The data set all information interaction is with a score of 0–100.

Table 1. Experimental data statistics

Datasets	Mafengwo	CAMRa2011
# Total users	5,275	602
# Total groups	995	290
# Total items	15,13	7,710
# Avg. group size	7.19	2.08
# Avg. record for a user	7.54	193.26
# Avg. record for an item	26.28	15.09

The dataset’s rating records are transformed into positive instances with a target value of 1, while instances with missing data are labeled as negative instances with a target value of 0. Table 1 presents the statistical findings for the two datasets. In both datasets, an event is regarded as a group, where the attendees of the event are considered as group members, and the event’s venues (or movies) signifies the selected items [2]. Our objective is to suggest an appropriate venue (or movie) for a group event. It’s important to note that these two datasets solely consist of observed interactions, which correspond to positive instances. To create balanced pairs, we randomly select missing data as negative

¹ <http://www.mafengwo.cn>.

² <http://2011.camrachallenge.com/2011>.

instances. Through observation, the optimal sampling rate is determined to be 6. Therefore, the negative sampling rate is fixed at 6. In particular, for every entry in the Mafengwo dataset, we selected 6 venues that the group had not previously visited at random. Similarly, for each log in the CAMRa2011 dataset, we sampled 6 movies that the group had not watched before. Each negative instance was assigned a target value of 0.

2. Evaluation Protocols

In this paper, we used the *leave-one-out* evaluation [28]. This method entails randomly excluding one interaction from each group for testing. Then the training and test sets are disjoint. However, ranking all items within each group could be time-consuming. Therefore, we randomly chose 100 items from non-interacted groups as part of our strategy. Subsequently, the test items were ranked within the pool of 100 items [2]. This study randomly picked 10% of these groups as the test group and excluded their group-item interactions. The experimental evaluation was conducted on Pytorch platform, and the performance of ranking list of this model was evaluated according to the indexes NDCG@k and HR@k. A larger value indicates better performance. Hyper-parameters tuning involve employing grid search. The learning rate and embedding size were tuned within the ranges [0.001, 0.005, 0.01, 0.05] and [16, 32, 64, 128], with a batch size of 256. For gradient-based optimization, the Adam optimizer was used. We consider two values of k: 5 and 10 respectively, to assess the model’s performance.

3. Baselines

To assess our model’s effectiveness, we will compare it against the following methods.

(1) Memory-based approach

These models utilize predefined score aggregation strategies. Firstly, the neural collaborative filtering(NCF) [29] is applied to obtain the recommendation scores of group members, and then some aggregation strategies are adopted to determine the final recommendation results.

- **NCF+AVG** [12]: The fusion of NCF and the average strategy involves using the average strategy to compute the mean preference scores of members.
- **NCF+LM** [30]: NCF combined with the least misery strategy, akin to the *cash principle*, fails to produce satisfactory outcomes for each group member.
- **NCF+MS** [13]: NCF combined with the maximum satisfaction strategy computes the average score of the top n-ranked group members as the group preference.

(2) Model-based approach

- **COM** [3]: This probabilistic approach models group activities by considering individual content factors and member influence for generating recommendations.
- **DPMF-CNN** [31]: This approach employs a dynamic probabilistic matrix factorization model and convolutional neural network to tackle recommendation tasks. It considers the temporal factor, rating matrix, and service description document.

- **AGREE** [2]: This method utilizes the attention mechanism to tackle the preference aggregation issue through data-driven learning of aggregation policies. It then conducts group recommendation using Neural Collaborative Filtering (NCF).
- **SoAGREE** [7]: Unlike AGREE, this approach introduces an additional attention network to integrate each group member’s followees information, thus improving individual user representation and better capturing their personal preferences.

5.2 Performance Comparison (RQ1)

Table 2 showcase data from two real-world datasets across various models. By analyzing these tables, we can deduce the following insights.

Table 2. Experimental comparison results and without \mathcal{L}_c

Methods	CAMRa2011				Mafengwo			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
NCF+AVG	0.5685	0.3818	0.7643	0.4449	0.3290	0.2225	0.6220	0.3420
NCF+LM	0.5598	0.3790	0.7645	0.4452	0.3168	0.2430	0.6305	0.3526
NCF+MS	0.5430	0.3713	0.7604	0.4348	0.3710	0.2385	0.6279	0.3145
COM	0.5789	0.3765	0.7688	0.4370	0.4422	0.3300	0.5433	0.3730
DPMF-CNN	0.5800	0.3842	0.7530	0.4543	0.4670	0.3455	0.6339	0.3768
AGREE	0.5818	0.3883	0.7708	0.4577	0.4810	0.3747	0.6398	0.4239
SoAGREE	0.5818	0.3883	0.7708	0.4577	0.4889	0.3800	0.6475	0.4298
CAGR	0.5972	0.3975	0.7890	0.4712	0.4945	0.3823	0.6405	0.4315
CAGR- \mathcal{L}_c	0.5862	0.3870	0.7731	0.4601	0.4834	0.3725	0.6310	0.4282

1. This observation is evident in the experimental outcomes. The CAGR model introduced in this paper demonstrates superior accuracy on the CAMRa2011 dataset compared to all seven baseline models. Moreover, good results are also achieved with the Mafengwo dataset. By utilizing the attention mechanism, we allocate distinct weights to each member within the group. We have addressed the shortcomings of attention mechanism in group recommendation applications, which may not accurately assess the significance of individual group members in the group recommendation. Therefore, this study incorporates causal inference and utilizes ITE to establish the causal relationship between group members and groups. This method better discriminates the significance of individual group members, leading to improved recommendation performance.

2. Among the baseline models, the least performing ones are NCF+AVG, NCF+LM, and NCF+MS. Because they simply aggregate the group’s preferences without considering the decision-making process. This finding also verifies

that the predefined static score aggregation strategy are inadequate in accurately achieve group decision-making, echoing the observations in [2, 32]. Unlike the baseline memory-based method, attention-based models show higher accuracy. AGREE and SoAGREE effectively capture the impact of individual group members through distinct weight assignments that indicate varying levels of importance. This approach can enhance the precision of group characteristic representation. However, they differ from our model in that while they also utilize attention, they ignore the fact that the attention mechanism itself is susceptible to confounding factors and might not capture the weight accurately. In addition, we found that SoAGREE performed superior on the Mafengwo dataset because the SoAGREE model considers social followees information, which is included in the Mafengwo dataset and not in the CAMRa2011 dataset. Therefore, in the CAMRa2011 dataset, SoAGREE degenerates into AGREE.

5.3 Influence of Causal Inference \mathcal{L}_c (RQ2)

This paper’s main contribution lies in incorporating causal inference into group recommendation, infusing causal relationships into the attention mechanism to derive the group representation. From Table 2 we observe that when causality is excluded from the final loss, performance across all datasets shows a decline. This outcome is attributed to the fact that the incorporation of causality in group recommendation, through the computation of ITE, enhances the identification of the significance of group members. This also leads to the acquisition of more accurate aggregated group representations.

5.4 Influence of Hyper-parameters (RQ3)

This section investigates the influence of key components and hyperparameters, then refines parameter values for optimal performance.

1. Influence of a

The parameter a regulates the extent to which we can promote sparsity in the values of α . In Fig. 2(a) and 2(b), we can observe that the model delivers optimal performance across all datasets when a assumes a more intermediate value. Specifically, the value of 0.6 is optimal for the CAMRa2011 dataset, while 0.5 is found to be the best choice for the Mafengwo dataset. When a is too small, it cannot impose too large a constraint on α_{gi} , which may result in the inability to distinguish key group members from other group members. On the other hand, an excessively large a could result in the suppression of numerous important group members, ultimately constraining the final outcomes. Hence, neither of these extremes is optimal. Achieving an effective trade-off necessitates fine-tuning a within the appropriate range.

2. Influence of ρ

We adjust the parameter ρ to examine the impact of the softness of the distribution η_{gi} . As illustrated in Fig. 2(c) and 2(d), for the CAMRa2011, the most optimal performance is achieved when $\rho = 0.1$. We propose the hypothesis that due to the relatively small number of group members in the dataset and

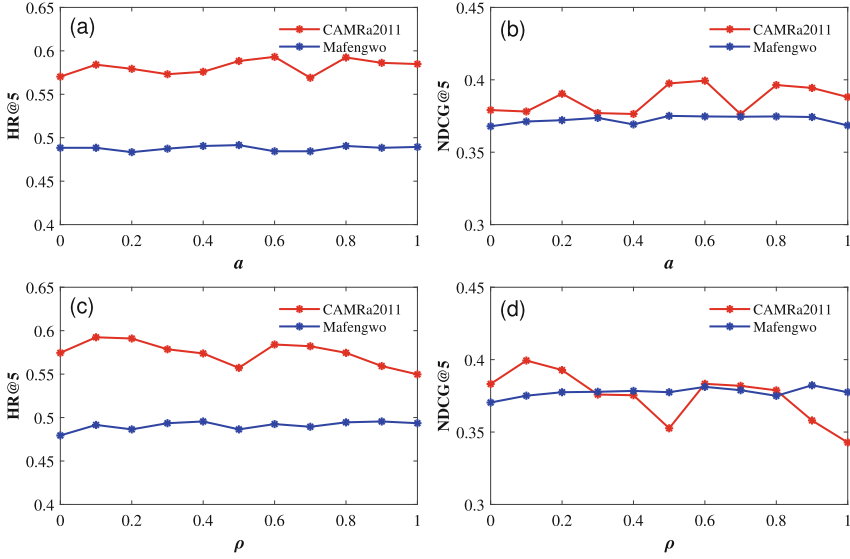


Fig. 2. Adjustment of the weight for the L1 norm and ρ .

the clear and stable group-item interactions, a lower value of ρ could result in a more distinct distribution of η_{gi} . Therefore, important group members can be effectively distinguished to achieve the best results. However, for the Mafengwo, the best performance is achieved when $\rho = 0.9$. We speculate that as the number of group members in the dataset increases, the number of significant members to be identified also grows. Therefore, raising the value of ρ would be beneficial to achieve optimal results.

5.5 Influence of Distance Function (RQ4)

Table 3. Experimental comparison results of different distance functions

Methods(Distance)	CAMRa2011				Mafengwo			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
CAGR(Euclidean)	0.5752	0.3788	0.7814	0.4445	0.4915	0.3736	0.6389	0.4286
CAGR(Cosine)	0.5745	0.3816	0.7886	0.4496	0.4884	0.3593	0.6289	0.4310
CAGR(Dot)	0.5710	0.3826	0.7828	0.4593	0.4915	0.3555	0.6349	0.4275
CAGR(Pearson)	0.5814	0.3834	0.7772	0.4551	0.4874	0.3511	0.5935	0.3812
CAGR(KL)	0.5972	0.3975	0.7890	0.4712	0.4945	0.3823	0.6405	0.4315

To establish a connection between attention weights α_{gi} and ITE vectors β_{gi} , we used a distance function to minimize the distance between them to distill

causal information into model optimization process. We investigated how the choice of distance function affects the ultimate performance outcomes. In this paper, the distance function experiment encompassed various metrics, including Euclidean distance, Cosine similarity, Dot product similarity, Pearson similarity and KL divergence to analyze their impact[9]. Examining the results presented in Table 3, notably, the Pearson correlation coefficient fared least favorably overall on both datasets. In a holistic context, our model attains optimal results on both datasets when employing the Kullback-Leibler divergence as the designated distance function. This underscores the influence of the distance function on recommendation performance, highlighting the KL divergence as adept at amalgamating α_{gi} and β_{gi} .

6 Conclusion

In this work, we propose the novel CAGR model. Leveraging the strength of attention neural networks in group recommendation and inspired by the potential outcome framework, we incorporate causal inference into the process. This leads to a redefined group inference task that bolsters the attention mechanism's effectiveness. By explicitly regularizing the distance between the Individual Treatment Effects (ITEs) of members and their corresponding attention weights, we effectively integrate causal information into the group representation learning process to accurately capture group preferences. The performance of the CAGR model is extensively evaluated through experiments on two genuine datasets.

In forthcoming research, we aim to address inherent fairness issues within groups. Moreover, we consider enhancing performance by incorporating user data to address the challenge of sparse group data, and also introduced causal inference to improve performance.

References







1. Yin, H., Cui, B.: Spatio-temporal recommendation in social media. Springer (2016)
2. Cao, D., He, X., Miao, L., An, Y., Yang, C., Hong, R.: Attentive group recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 645–654 (2018)
3. Yuan, Q., Cong, G., Lin, C.-Y.: Com: a generative model for group recommendation. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 163–172 (2014)
4. Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., Cao, W.: Deep modeling of group preferences for group-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28(1) (2014)
5. Wu, L., Sun, P., Hong, R., Ge, Y., Wang, M.: Collaborative neural social recommendation. *IEEE Trans. Syst. Man Cybernet. Syst.* **51**(1), 464–476 (2018)
6. Guo, L., Yin, H., Chen, T., Zhang, X., Zheng, K.: Hierarchical hyperedge embedding-based representation learning for group recommendation. *ACM Trans. Inform. Syst. (TOIS)* **40**(1), 1–27 (2021)

7. Cao, D., He, X., Miao, L., Xiao, G., Chen, H., Xu, J.: Social-enhanced attentive group recommendation. *IEEE Trans. Knowl. Data Eng.* **33**(3), 1195–1209 (2019)
8. Vinh Tran, L., Nguyen Pham, T.-A., Tay, Y., Liu, Y., Cong, G., Li, X.: Interact and decide: Medley of sub-attention networks for effective group recommendation. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development In Information Retrieval*, pp. 255–264 (2019)
9. Zhang, J., Chen, X., Zhao, W.X.: Causally attentive collaborative filtering. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3622–3626 (2021)
10. Xu, S., Ge, Y., Li, Y., Fu, Z., Chen, X., Zhang, Y.: Causal collaborative filtering. In: *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 235–245 (2023)
11. Rubin, D.B.: Causal inference using potential outcomes: design, modeling, decisions. *J. Am. Stat. Assoc.* **100**(469), 322–331 (2005)
12. McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P.: Cats: a synchronous approach to collaborative group recommendation. In: *FLAIRS Conference*, vol. 2006, pp. 86–91 (2006)
13. Boratto, L., Carta, S.: State-of-the-art in group recommendation and new approaches for automatic identification of groups. In: *Information Retrieval and Mining in Distributed Environments*. Springer, pp. 1–20 (2011)
14. Liu, X., Tian, Y., Ye, M., Lee, W.-C.: Exploring personal impact for group recommendation. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 674–683 (2012)
15. Carvalho, L.A.M.C., Macedo, H.T.: Users' satisfaction in recommendation systems for groups: an approach based on noncooperative games. In: *Proceedings of the 22nd International Conference on World Wide Web*, pp. 951–958 (2013)
16. Bastias, P.G., Brand, J.E.: *Causal Inference*. Oxford University Press (2020)
17. Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., He, X.: Bias and debias in recommender system: a survey and future directions [arXiv preprint arXiv: 2010.03240](https://arxiv.org/abs/2010.03240) (2020)
18. Abdollahpouri, H., Mansoury, M.: Multi-sided exposure bias in recommendation, [arXiv preprint arXiv:2006.15772](https://arxiv.org/abs/2006.15772) (2020)
19. Wang, W., Feng, F., He, X., Zhang, H., Chua, T.-S.: Clicks can be cheating: counterfactual recommendation for mitigating clickbait issue. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1288–1297 (2021)
20. Wang, W., Feng, F., He, X., Wang, X., Chua, T.-S.: Deconfounded recommendation for alleviating bias amplification. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1717–1725 (2021)
21. Mehrotra, R., Bhattacharya, P., Lalmas, M.: Inferring the causal impact of new track releases on music recommendation platforms through counterfactual prediction. In: *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 687–691 (2020)
22. Yuan, B.: Improving ad click prediction by considering non-displayed events. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 329–338 (2019)
23. Wang, X., Li, Q., Yu, D., Cui, P., Wang, Z., Xu, G.: Causal disentanglement for semantics-aware intent learning in recommendation. *IEEE Trans. Knowl. Data Eng.* (2022)

24. Wang, Y., Liang, D., Charlin, L., Blei, D.M.: Causal inference for recommender systems. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 426–431 (2020)
25. Sato, M., Takemori, S., Singh, J., Ohkuma, T.: Unbiased learning for the causal effect of recommendation. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 378–387 (2020)
26. Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., Chua, T.-S.: Attentional factorization machines: Learning the weight of feature interactions via attention networks, arXiv preprint [arXiv:1708.04617](https://arxiv.org/abs/1708.04617) (2017)
27. Wang, X., He, X., Nie, L., Chua, T.-S.: Item silk road: recommending items from information domains to social users. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 185–194 (2017)
28. Chen, J., Zhang, H., He, X., Nie, L., Liu, W., Chua, T.-S.: Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 335–344 (2017)
29. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
30. De Pessemier, T., Dooms, S., Martens, L.: Comparison of group recommendation algorithms. *Multimedia tools Appl.* **72**, 2497–2541 (2014)
31. Wang, H., Dong, M.: Latent group recommendation based on dynamic probabilistic matrix factorization model integrated with cnn. *J. Comput. Res. Develop.* **54**(8), 1852–1863 (2017)
32. Huang, Z., Liu, Y., Zhan, C., Lin, C., Cai, W., Chen, Y.: A novel group recommendation model with two-stage deep learning. *IEEE Trans. Syst. Man Cybernet. Syst.* **52**(9), 5853–5864 (2021)



E^2DAS : An Efficient Equivariant Dynamic Aggregation Saliency Model for Omnidirectional Images

Nana Zhang¹ , Qian Liu¹ , Dandan Zhu^{2(✉)} , Kun Zhu^{3(✉)} ,
Guangtao Zhai⁴ , and Xiaokang Yang⁴ 

¹ School of Computer Science and Technology, Donghua University, No.2999,
Renmin North Road, Songjiang District, Shanghai 201620, China

² Institute of AI Education, Shanghai, East China Normal University, No.3663,
Zhongshan North Road, Putuo District, Shanghai 200333, China
ddzhui@mail.ccnu.edu.cn

³ Key Laboratory of Embedded System and Service Computing, Ministry
of Education, Tongji University, No.4800, Cao'an Highway, Jiading District,
Shanghai 201804, China
kyzhui@tongji.edu.cn

⁴ Institute of Image Communication and Network Engineering, Shanghai Jiao Tong
University, No. 800, Dongchuan Road, Minhang District, Shanghai 200240, China

Abstract. Recent years have witnessed rapid progress of convolutional neural networks (CNNs) and their successful application in the task of saliency prediction for omnidirectional images (ODIs). Albeit achieving tremendous performance improvements, these CNNs-based saliency models are plagued by two major shortcomings: spatial content-agnostic and computationally intensive. Inspired by the effectiveness of equivariant network in the majority of computer vision tasks, we propose a novel efficient equivariant dynamic aggregation saliency (E^2DAS) model to efficiently tackle the issue of human fixation prediction in ODIs. To be specific, our proposed model consists of an efficient equivariant module, a dynamic convolutional aggregation module, and an optimization computation module. Different from existing saliency models for ODIs, we are the first attempt to introduce an efficient equivariant dynamic convolutional aggregation operation into the saliency prediction task, which can fundamentally alleviate the projection distortion problem and can effectively learn spatial content-adaptive features. Moreover, we clearly observe a considerable decrease in the number of parameters resulting from the replacement of standard convolution with dynamic convolution aggregation. Extensive experiments on several benchmark datasets show the proposed model's superiority over other state-of-the-art methods in terms of performance.

Keywords: Equivariant dynamic aggregation · spatial content-adaptive · saliency prediction · omnidirectional images · light-weight model

1 Introduction

In recent decades, advancements in virtual reality (VR) and stereoscopic technology have led to significant growth and diverse applications, including the presentation of omnidirectional images (ODIs) via head-mounted displays (HMDs). HMDs allow users to fully immerse themselves in virtual experiences and engage in novel forms of interaction. Unlike traditional 2D images, ODIs provide a higher resolution and a wide field of view (FoV). When wearing an HMD, users can freely explore scenes within a FoV of $180^\circ \times 360^\circ$, resulting in a truly immersive experience. However, the high resolution of ODIs presents challenges in terms of storage, streaming, and rendering. Effectively addressing these aspects becomes crucial. Therefore, accurate saliency prediction in ODIs is vital for streamlining data management and optimizing network resource distribution.

Convolutional neural networks (CNNs) have gained immense popularity for saliency prediction tasks and have demonstrated remarkable advancements in performance. In the task of 2D image saliency prediction, many representative CNNs-based saliency models [7] have emerged. Despite the significant progress made in 2D image saliency prediction, these CNNs-based models are not entirely suitable for ODIs and may lead to a degradation in performance. To this end, several representative saliency models [23] designed for ODIs have emerged and substantially improved performance. For instance, Ling *et al.* [23] utilize a sparse matrix-based dictionary to extract image features and apply dimensionality-biased augmentation to perform saliency estimation for ODIs. Lebreton *et al.* [22] introduce two saliency models based on traditional 2D images: boolean map-based saliency [36] and graph-based visual saliency [29]. Although these methods utilizing hand-crafted features have made advancements in saliency prediction for ODIs, they often encounter challenges when dealing with complex scenes. Due to the inherent limitations of hand-crafted features, it is challenging to capture the intricate details and subtle visual cues in ODIs. Thus, more advanced models are urgently needed to effectively exploit the rich and semantic information in ODIs to achieve superior performance of saliency prediction.

Several saliency models CNNs-based have been proposed to enhance the performance of saliency prediction for ODIs, yielding promising results [5, 25]. Specifically, motivated by the structure of generative adversarial networks (GANs [11]), Pan *et al.* [25] designed a GAN-based saliency prediction model for predicting the head fixation on ODIs. Subsequently, to mitigate the issue of projection distortion when projecting ODI to 2D image, some saliency models [1, 5] with attention mechanism and context-aware features are proposed to achieve more accurate saliency prediction and alleviate the projection distortion problem. While these models have achieved performance gains, they face limitations: 1) CNNs' translation equivariance due to parameter sharing cannot effectively address projection distortion in ODIs. 2) these models have high computational costs, making them unsuitable for real-time applications. Therefore, there is a pressing necessity to propose a lightweight saliency model specifically designed for ODIs, considering both reducing projection distortion and minimizing computational costs.

Drawing inspiration from the group equivariant convolutional neural network (G-CNN) [8], we propose a novel efficient equivariant dynamic aggregation saliency (E^2DAS) model. This model effectively mitigates projection distortion and reduces computational complexity by incorporating three core components: an efficient equivariant module, a dynamic convolutional aggregation module, and an optimization computation module. Notably, this represents the initial attempt to incorporate dynamic convolutional aggregation operation into the ODIs saliency prediction task. Different from the conventional convolution filter, E^4 [13] convolutional filter (*i.e.*, rotational equivariant filter) can diminish projection distortion during the mapping of spherical images onto a 2D plane, while also offering substantial reductions in computational costs. Specifically, by incorporating E^4 convolutional filters into the saliency prediction task of ODIs, our filter can perform calculations based on the input features, making the proposed E^2DAS model dynamic and reducing projection distortion. Furthermore, to reduce the computational cost, we decouple the feature aggregation operation into attention calculation and kernel aggregation operations. This decoupling mechanism can reduce feature redundancy in the convolution filter and accelerate the computation. As our proposed model is both computational efficient and feature equivariant, thus we name our model as E^2DAS . In brief, the main contributions of this paper can be outlined as follows:

- We propose a novel efficient equivariant dynamic aggregation saliency model E^2DAS , which effectively addresses the projection distortion problem and reduces computation costs.
- We are the first to introduce an efficient equivariant dynamic convolutional aggregation operation into the saliency prediction task, which can effectively learn spatial content-adaptive features and reduce the model parameters.
- We present a comprehensive analysis on two benchmark datasets widely used in the field of saliency prediction. Through extensive experiments and comparisons, we demonstrate the significant performance gains achieved by our (E^2DAS) model over the current state-of-the-art methods.

2 Related Work

2.1 Saliency Prediction Methods for ODIs

Recently, the metaverse and Artificial Intelligence Generated Content (AIGC) technologies have significantly contributed to advancements in content generation and display, leading to the emergence of numerous saliency methods for ODIs. These methods help reduce transmission load and conserve network resources. To our knowledge, existing saliency prediction methods for ODIs can be broadly classified into two main categories: improved saliency prediction methods for ODIs and CNNs-based saliency prediction methods. In the realm of the improved saliency prediction methods, Ling *et al.* [23] and Federica *et al.* [2] adopted heuristics manner to extract features and achieved saliency prediction in ODIs. Lebreton *et al.* expanded the traditional 2D saliency method to

the ODIs through fine-tuning, and proposed two new methods: GBVS360 [22] and BMS360 [22]. Indeed, with the continuous advancements in CNNs, numerous saliency prediction methods based on CNNs have emerged [2, 23]. PAN *et al.* [25] introduced a method for enhancing saliency prediction performance in ODIs by drawing inspiration from generative adversarial networks (GANs). Since SalGAN360 [4] can predict saliency in ODIs even when the bipolar global information is seriously lost, Chen *et al.* [5] introduced a bifurcation model that preserved both local and global features. Although these CNNs-based methods improve performance, they often incur high computational costs due to the use of standard convolutional filters. Therefore, it is imperative to introduce a lightweight saliency prediction model that strikes a superior balance between prediction performance and computational efficiency.

2.2 Equivariant Networks

CNNs possess translational equivariance, reducing parameters by applying shared convolutional kernel weights to extract features from different spatial locations. To extend equivariance to larger rotational and scale symmetry groups, Cohen and Welling [8] proposed G-CNNs, which ensures rotational [8] and scale [33] equivariance when performing feature extraction. Although G-CNNs incorporate more equivariance, leading to notable performance improvements over conventional CNNs, they encounter two challenges: spatial-agnostic and high computational cost. To address these issues, He *et al.* [13] proposed a generalized framework E^4 with equivariance, which decouples space and additional geometric dimensions in computation to accelerate neural network operations in parallel. At the same time, feature aggregation is divided into kernel generators and encoders, which helps alleviate the spatial content-agnostic issue. Motivated by this, we apply the E^4 network to the task of saliency prediction in ODIs, aiming to address high projection distortion and computational cost simultaneously.

2.3 Dynamic Convolution

Over the past few years, CNNs have gained considerable popularity and demonstrated remarkable performance across diverse computer vision tasks. However, with the rapid growth in data volume due to internet technology advancements, the main goal for researchers [15–17, 28] is to design an efficient CNN model that achieves a balance between performance and computational cost. In particular, MobileNetV1 [16] significantly reduces the number of floating-point operations (FLOPs) by decomposing 3×3 filter into depth and point convolution. Building upon this, MobileNetV2 [28] introduced reverse residuals and linear bottlenecks to further enhance efficiency. MobileNetV3 [15] applied squeeze-and-excitation(SE) operation in the residual layer [17] and adopted a platform-aware neural structure approach [32] to find the optimal network structure. The 1×1 convolution is further reduced by channel shuffling operation. ShiftNet [34] replaced expensive spatial convolution with shift operations and point-state

convolution. However, when the computational constraints become extremely low, the width and depth of these efficient networks will be limited, leading to a substantial reduction in the network’s representation capabilities. To tackle this challenge, Chen *et al.* [6] proposed dynamic convolution, which increases model complexity without requiring an increase in depth or width.

3 Proposed Model

In this section, we thoroughly describe our proposed E²DAS model. The general architecture of our model is illustrated in Fig. 1. It comprises three main modules: an efficient equivariant convolution module for feature extraction, a dynamic convolutional aggregation module for enhancing feature representation ability, and an optimization computation module for measuring the disparity between the predicted saliency map and the ground-truth. Notably, we introduce the equivariant dynamic convolution aggregation operation for the first time in the task of ODIs saliency prediction. This operation not only improves computational efficiency but also enhances feature representation capability. Detailed discussions on each component follow in the next subsection.

3.1 Efficient Equivariant Module

Most top-ranking saliency prediction models of ODIs usually rely on deeper or wider CNNs to achieve strong feature representation. However, the computational complexity of these saliency models often hampers their practical applications in real-world scenarios. Additionally, standard convolution filters with grid setting are not suitable for ODIs saliency prediction because they inevitably incur projection distortion when ODIs are projected onto a 2D plane. More specifically, projection distortion arises from the rotational angle of the spherical image, resulting in stretching especially at the two poles. To address the aforementioned issues, we introduce the equivariant E⁴ convolution filter for the first time into the task of ODIs saliency prediction. This innovative approach effectively alleviates projection distortion and significantly reduces computational burden.

Specifically, the E⁴ convolutional filter is composed of translation and rotation operations. It performs a 90-degree rotation around any center of rotation within a square grid, making it the smallest unit of rotation. With this advantage, E⁴ convolution filter enables the extraction of rotation-equivariant features, thereby enabling accurate prediction of human attention in ODIs. Each E⁴ convolution filter encompasses all possible combinations of translations and 90° rotations around different centers of rotation within the square grid. For the purpose of executing this process, the convolution operation of E⁴ is divided into $K(\cdot)$ operation (kernel generator) and $V(\cdot)$ operation (encoder). Mathematically, E⁴ convolution [13] can be defined as follows:

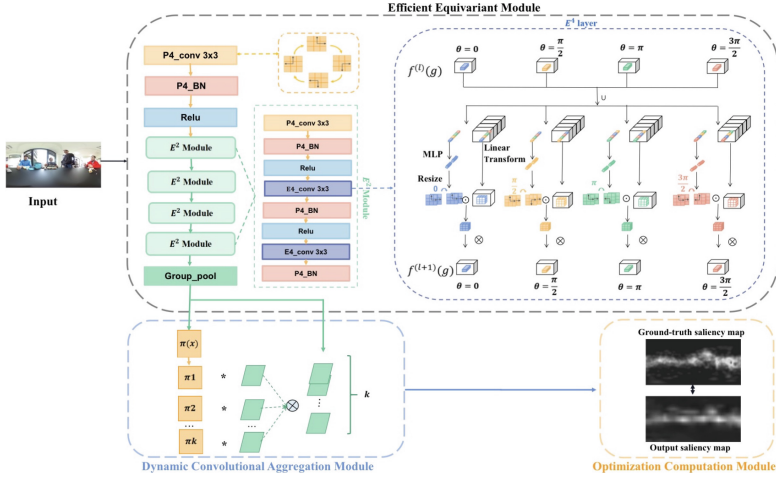


Fig. 1. Architecture of the proposed $E^2 DAS$ model, which comprises an efficient equivariant module, a dynamic convolutional aggregation module, and an optimization computation module. In particular, An example of the E^4 layer in an equivariant convolutional network is shown in the upper right corner [13]. Firstly, we adopt four rotation convolution filters (*i.e.* 0, 90, 180, 270) to extract the rotation invariant features. It’s important to note that these four feature extraction processes are performed in parallel. On one of the channels, the concatenated features through an MLP layer to generate kernel features. On the other channels, the concatenated features pass through a linear layer to generate encoded features. Subsequently, we employ the element-wise product operation to learn the encoded features. Lastly, spatial-wise aggregation is executed to acquire the ultimate feature representation.

$$\mathbf{f}^{(l+1)}(g) = \sum_{\tilde{g} \in \mathcal{N}(g)} K_{g^{-1}\tilde{g}} \left(\bigcup_{g' \in \mathcal{N}_1(g)} \mathbf{f}^{(l)}(g') \right) \odot V \left(\bigcup_{\tilde{g}' \in \mathcal{N}_2(\tilde{g})} \mathbf{f}^{(l)}(\tilde{g}') \right), \quad (1)$$

where $\mathcal{N}(g)$ is the spatial-wise neighborhood pixels that need to be aggregated, *i.e.*, $\mathcal{N}(g) = \{g(v, e_A) \mid v \in \Omega\}$, $\Omega \subseteq \mathbb{R}^k$ and e_A is the identity element of group A . $f^{(l)}$ represents the feature at the l^{th} layer, $\mathcal{N}_i(g)$ represents the neighborhood pixels of $g \in \{gg' \mid g' \in \mathcal{N}_i(e)\}$ and $\mathcal{N}_i(e)$ is the predefined neighborhood of the identity element $e \in G$, which is the group. g' and \tilde{g}' represent a pair of relative locations. \bigcup and \odot denote the rotation-wise concatenation and element-wise product operations. In this way, we can obtain the output representation of E^4 convolution filter. The upper right corner of Fig. 1 shows the detailed structure of the E^4 convolution layer. Notably, this design improves computational efficiency and reduces projection distortion.

3.2 Dynamic Convolution Aggregation Module

The efficient equivariant convolution module in our proposed E²DAS model aims to generate discriminative features to minimize projection distortion in ODIs saliency prediction. The goal of the dynamic convolution aggregation module is to augment the model’s feature representation capability, thereby alleviating the problem of spatial content-agnostic problem. In other words, this module improves the feature representation of the previous module’s output by dynamically aggregating multiple parallel convolution filters using attention weights. The process involves two stages: attention calculation and kernel aggregation. In the attention calculation stage, weights are assigned to each convolution kernel. In the kernel aggregation stage, the parallel filters are combined based on these weights.

Attention Calculation. Taking inspiration from the effectiveness of SE attention mechanism in enhancing the feature representations, we directly introduce the SE attention into the ODIs saliency task in our proposed E²DAS model. Specifically, the attention calculation involves the following steps: Initially, a global pooling operation is employed to condense the global spatial information. Subsequently, two fully connected layers and softmax operations are utilized to generate normalized attention weights for the parallel convolution kernels of efficient equivariant module. These operations reduce the size of the feature map by a factor of 4. Unlike SENet [17], where attention is calculated on the output feature channel, our dynamic convolution aggregation operation calculates attention on the conventional filter. Consequently, the computational cost of attention calculation is relatively low and negligible. For example, for a feature map with an input size of $h \times w \times c_{in}$, the attention computation requires $O(\pi(x)) = hwc_{in} + c_{in}^2/4 + c_{in}k/4$ Multi-Adds. On the other hand, the calculation amount for traditional convolution is $O(\tilde{W}^T x + \tilde{b}) = hwc_{in} c_{out} k_n^2$, in which c_{in} and c_{out} signify the quantities of input and output channels respectively, while k_n represents the size of the kernel. It is evident that the attention computation in the dynamic aggregation module is significantly lower than that of traditional convolution.

Kernel Aggregation. In traditional convolution operations, each layer typically uses a single convolution filter to extract features. However, in dynamic convolution operations, multiple parallel convolution kernels can be dynamically aggregated based on attention weights. Specifically, the dynamic convolution aggregation operation aggregates a set of 4 parallel convolutional kernels using attention weights. The weights w and bias b can be expressed as:

$$\tilde{w} = \sum_k \pi_k(x) \tilde{w}_k, \quad \tilde{b} = \sum_k \pi_k(x) \tilde{b}_k, \quad (2)$$

where x represents the input feature, k denotes the number of convolution kernels, \tilde{w} and \tilde{b} represent the output weights and biases after dynamic convolution

aggregated operations, respectively. It should be emphasized that while dynamic convolution aggregation operation increases the model’s, the output dimension of each layer remains unchanged. Moreover, compared to traditional convolution, the increase in computation due to the convolution kernel is negligible. Therefore, the devised efficient equivariant dynamic aggregation module not only achieves superior saliency prediction performance but also offers the advantage of high computational efficiency.

3.3 Optimization Computation Module

To our knowledge, the loss function plays a crucial role in evaluating the performance of a model by measuring the discrepancy between predicted and ground-truth data. Its primary purpose is to aid the training process by optimizing the network’s parameters and guiding the model towards convergence. To attain the optimal performance of saliency prediction for ODIs, we choose smooth \mathcal{L}_1 loss to guide the model training. Mathematically, the \mathcal{L}_1 loss can be expressed as the following:

$$\text{loss}(x, y) = \frac{1}{n} \sum_i z_i, \quad (3)$$

where z_i can be expressed as the following:

$$z_i = \begin{cases} \frac{1}{2} (x_i - y_i)^2, & \text{if } |x_i - y_i| < 1, \\ |x_i - y_i| - \frac{1}{2}, & \text{otherwise,} \end{cases} \quad (4)$$

where y_i denotes the ground-truth, x_i represents the predicted value, and n signifies the total number of sample points. From Eqn.4, we can clearly observe that the smooth \mathcal{L}_1 loss is actually a piecewise function. If the absolute difference $|x_i - y_i|$ is less than or equal to 1, the equation represents \mathcal{L}_2 loss, which solves the problem of non-smoothness in \mathcal{L}_1 . On the other hand, when the absolute difference is greater than 1, the equation represents \mathcal{L}_1 loss, which handles the issue of outlier gradient explosions.

4 Experiments

4.1 Datasets

Our primary objective is to assess the performance of our proposed model in predicting saliency. To accomplish this, we employ two publicly available datasets: Salient360! [26, 27] and AOI [35]. The Salient360! dataset contains 85 ODIs with salience plots under free-view conditions, involving 63 observers (avg. 42 per stimulus) who view each ODI for 25 s, separated by a 5-second gray screen interval. The dataset covers indoor, outdoor, and face scenes. The AOI dataset has 600 high-res ODIs categorized into human, cityscapes, indoor, and natural landscapes. Both datasets’ ground-truth is in equirectangular format. Due to the ODI dataset’s limited size, direct deep neural network training is unfeasible. Hence, we use transfer learning. We pretrain our model on the larger and more diverse SALICON [20] dataset and fine-tune it using Salient360! and AOI datasets for saliency prediction in ODIs.

4.2 Evaluation Metrics

To comprehensively access the effectiveness of our proposed model, we employ several widely used evaluation metrics, as described in the work by Bylinskii *et al.* [3]. The four evaluation metrics employed are CC (linear correlation coefficient), AUC (Area under ROC curve), NSS (Normalized Scanpath Saliency), and KLD (Kullback–Leibler divergence).

4.3 Implementation Details

For implementing our proposed E²DAS model, we use PyTorch with the Adam optimizer, setting weight decay to 0.00001 and an initial learning rate of 0.001. During training, we adjust the learning rate reductions at specific epochs. At the 60th, 120th, and 160th epochs, the learning rate is decreased by a factor of 0.1. This strategy is commonly used to fine-tune the learning process and allow the model to converge more effectively. With a batch size of 4, our model converges after approximately 200 epochs, and we train it on an NVIDIA GeForce RTX3090Ti GPU platform.

4.4 Comparison with State-of-the-Art Models

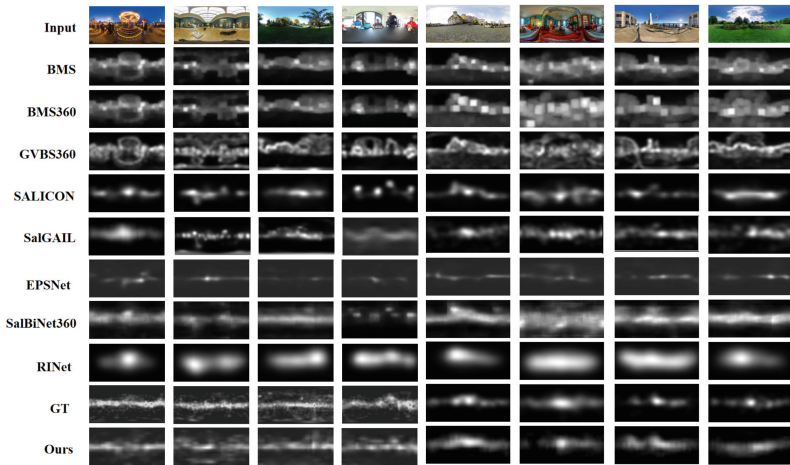


Fig. 2. Compare various methods visually on the Saliency360! and AOI datasets. The first four columns are from Saliency360! and the next four columns are from AOI. Among them, the four images from AOI represent four categories: cityscapes, indoor scenes, human scenes, and natural landscapes.

In this subsection, we conduct a comprehensive comparison between our proposed model, E²DAS, and several state-of-the-art saliency prediction methods.

The comparison is carried out from both qualitative and quantitative perspectives. The comparison methods consist of various approaches, such as BMS [36], BMS360 [22], GBVS360 [22], SALICON [19], SalGAN360 [4], SalBiNet360 [5], SalGAIL [35], RINet [31], and EPSNet [1]. We select these models by considering the public code availability and their representability of the state-of-the-art. The first three models are designed for 2D images and utilize low-level features. The remaining models are CNN-based, representing the most recent advancements in saliency prediction. These CNN-based models include saliency prediction for both 2D images and ODIs. Therefore, we believe that the selected models adequately represent the state-of-the-art in this area.

Qualitative Comparison. To intuitively demonstrate the outstanding performance of the proposed E^2DAS model in ODIs saliency prediction, we visually compare it with multiple state-of-the-art saliency prediction methods on two benchmark datasets: Salient360! and AOI. Specifically, we randomly select 4 ODIs from Salient360! dataset and one ODI from each category on the AOI [35] to compare the performance of our proposed E^2DAS model against other methods. The visual results are presented in Fig. 2, revealing that our model’s saliency maps are more similar to the ground-truth than other methods. This visual comparison serves as compelling evidence of our model’s superiority on both the Salient360! and AOI datasets. In summary, the visualization results clearly indicate that our model’s superiority over other methods, demonstrating its capability to generate saliency maps that closely resemble the ground-truth.

Table 1. A comparison of different saliency prediction approaches on both Salient360! and AOI datasets.

Methods	Salient360!				AOI			
	CC \uparrow	AUC \uparrow	NSS \uparrow	KLD \downarrow	CC \uparrow	AUC \uparrow	NSS \uparrow	KLD \downarrow
BMS [36]	0.560	0.719	0.958	0.587	0.557	0.758	0.975	0.584
BMS360 [22]	0.614	0.751	1.373	0.581	0.714	0.841	1.378	0.584
GBVS360 [22]	0.584	0.835	0.993	0.559	0.590	0.766	0.995	0.559
MLNet [9]	0.429	0.638	0.462	1.367	0.589	0.784	1.064	0.844
SalBiNet360 [5]	0.661	0.749	0.975	0.402	0.722	0.803	1.167	0.448
RINet [31]	0.558	0.772	1.501	0.781	0.736	0.799	1.141	0.380
EPSNet [1]	0.714	0.742	0.864	0.624	0.574	0.836	1.445	0.627
SalGAIL [35]	0.757	0.708	0.893	0.366	0.742	0.853	1.556	0.345
SALCON [19]	0.726	0.770	1.391	0.532	0.511	0.857	0.856	0.637
Ours	0.828	0.860	1.727	0.333	0.872	0.853	1.562	0.323

Quantitative Comparison. To comprehensively evaluate the performance of our E^2DAS model in ODIs saliency prediction, we conduct a comparative analysis with several other methods on two benchmark datasets. Specifically, on the Salient360! dataset, our model outperforms the second-best method, namely CC has improved to 0.828, AUC to 0.860, NSS to 1.727, and a KL divergence to 0.333. Moreover, on the AOI dataset, our model achieves an impressive 0.872 for CC, 0.853 for AUC, 1.562 for NSS, and 0.323 for KL divergence. These outstanding qualitative and quantitative results provide strong evidence for the superiority of our model in saliency prediction across both datasets.

4.5 Ablation Study

In this subsection, we conduct comprehensive ablation studies to affirm the efficiency of each element in our proposed E^2DAS model. The purpose of these experiments is to gain a deeper understanding of the contribution of each component.

Table 2. Ablation analysis of several convolutions over AOI dataset. The backbone is ResNet50 [12].

Convolution type	Metric			
	CC \uparrow	AUC \uparrow	NSS \uparrow	KL divergence \downarrow
Standard convolution	0.851	0.852	1.526	0.328
E^4 convolution	0.731	0.832	1.475	0.426
DY convolution	0.843	0.849	1.505	0.605
E^2DAS convolution	0.873	0.853	1.562	0.323

Effectiveness of Different Convolution Filters. To further validate the effectiveness of our devised convolution filter in the ODIs saliency prediction, we replace our devised convolution filter for each layer in backbone network ResNet50 [12] with standard convolution, dynamic convolution (DY), and E^4 convolution filters, respectively. The detailed comparison results are presented in Table 2 and Fig. 3. As depicted in Table 2, our designed convolution filter significantly improves CC, NSS, AUC, and KL divergence results compared to the standard convolution filter. In contrast, substituting the standard convolution filter with the dynamic convolution filter (DY) and the E^4 convolution filter resulted in lower performance in saliency prediction. Figure 3 clearly demonstrates that the saliency maps produced by $E^2DAS_ResNet50$ [12] exhibit a closer resemblance to the ground-truth, thanks to our rotationally invariant and spatially content-adaptive convolution filter. To summarize, the quantitative and qualitative results obtained from these experiments verify the effectiveness of our devised convolution filter in the proposed E^2DAS model for ODIs saliency prediction.

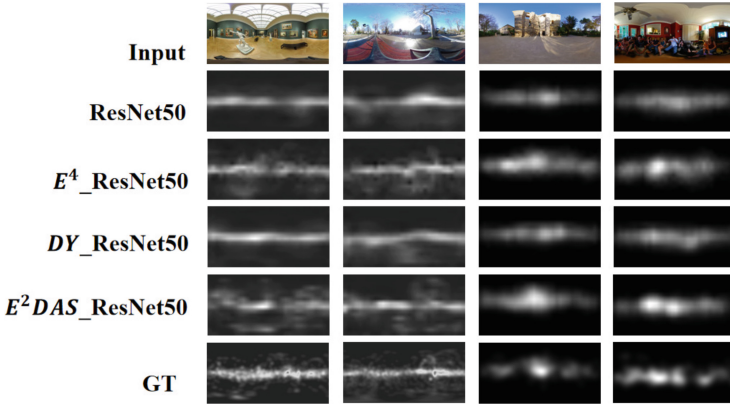


Fig. 3. A visual comparison of various convolution filters on both Salient360! and AOI datasets. The first two rows depict examples from Salient360!, while the remaining ones showcase samples from AOI.

Effectiveness of Different Backbone Networks. To validate the effectiveness of our proposed E^2DAS model using ResNet50 as the backbone network, we conduct experiments by replacing the ResNet50 with VGG16 [30], AlexNet [21], and DenseNet [18], respectively. The experiments are performed on the Salient360! dataset, while keeping the other settings of the E^2DAS model unchanged to ensure a fair comparison. The results, presented in Fig. 4, clearly show that the ResNet50 backbone outperforms other backbone networks (*e.g.*, VGG16, AlexNet, and DenseNet), indicating the effectiveness of our proposed E^2DAS model with the ResNet50 for the saliency prediction on the salient360! dataset.

Effectiveness of Different Numbers of Layer in Backbone Network. In our work, we also investigate the impact of varying backbone network layers in our proposed E^2DAS model. In particular, we adopt ResNet [12] with different layer configurations as the backbone to assess their performance in ODIs saliency prediction. Figure 5 presents the comparison results by using ResNet18, ResNet34, and ResNet50 on the Salient360! dataset and AOI dataset. As shown in this figure, increasing the number of network layers significantly improves saliency prediction performance. Notably, when the number of network layers reaches 50, the performance is optimized. Therefore, we choose the ResNet50 as the backbone of our proposed model.

4.6 Computational Complexity Analysis

To improve ODI saliency prediction, existing methods use deeper or wider CNNs but face high computational costs. We aim for a better performance-cost trade-off by proposing a novel equivariant dynamic aggregation saliency model, addressing

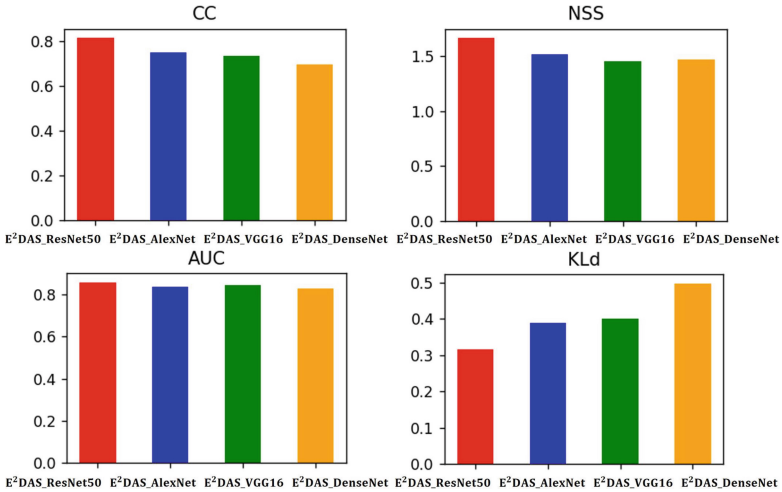


Fig. 4. The proposed E^2DAS model employing various base networks, depicting the performance comparison using four evaluation metrics.

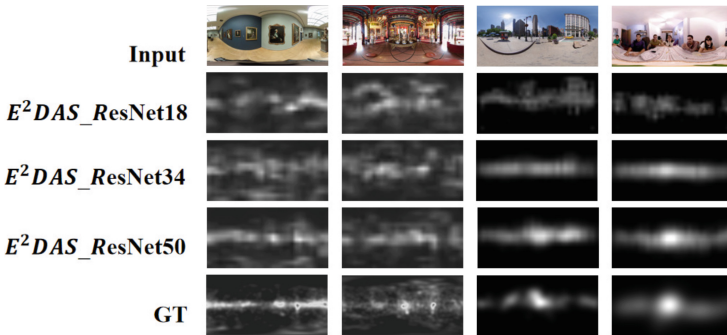


Fig. 5. Qualitative comparison of different layers of ResNet [12] on both the Salient360! dataset and AOI dataset. The first two columns display results from the Salient360! dataset, while the remaining columns showcase results from the AOI dataset.

projection distortion and reducing computational cost. To clearly and intuitively prove the lightweight nature of our proposed model, we compare our model’s size and running time with other saliency methods on the Salient360! dataset using a mobile device with a RAM platform. Our E^2DAS model outperforms others, indicating its lightweight nature and suitability for resource-constrained mobile devices, as shown in Table 3.

Table 3. Comparison of the model size and running time of our model with other methods over the Salient360! dataset.

Comparison Methods	Model Size (MB)	Runtime (s)
BMS [36]	27	0.145
BMS360 [22]	32	0.135
GBVS360 [22]	34	0.284
SAM [10]	96	0.10
MLNet [9]	75	0.318
SALICON [19]	130	0.341
SalNet360 [24]	85	0.103
SalGAN360 [4]	130	0.020
Vavadvsm [14]	138	0.142
SalGAIL [35]	68	0.040
Ours	1.79	0.005

4.7 Discussion

Our proposed E^2DAS model consistently demonstrates excellent performance across various datasets and evaluation metrics. This improvement is attributed to two key points: 1) the integration of E^4 convolution, which introduces spatial knowability, addresses projection distortion, and reduces network parameters for improved computational efficiency. 2) The dynamic aggregation method in our model uses the attention mechanism to adjust relevant information in the output of layer E^4 , achieving improved saliency prediction performance while reducing the number of model parameters. However, it struggles with generalization to real-world ODIs and has limited practical application. Future work will focus on enhancing generalization and exploring practical uses, such as ODI compression and quality assessment.

5 Conclusion

The main goal of this work is to develop a lightweight ODIs saliency prediction model capable of effectively addressing projection distortion and reducing computational costs. To this end, we propose an efficient equivariant dynamic aggregation saliency E^2DAS model, which addresses the issue of projection distortion and effectively learns spatial content-adaptive features. Specifically, the model comprises an efficient equivariant module for extracting rotation-invariant features and reducing model parameters, a dynamic convolutional aggregation module for learning spatial content-adaptive features and enhancing feature representations, an optimization computation module for calculating the difference between the predicted saliency map and ground-truth. We conduct extensive experiments on two benchmark datasets: salient360! and AOI. Experimental

results demonstrate that our proposed model achieves excellent performance in saliency prediction while maintaining a low computational cost. These findings validate the effectiveness and efficiency of the E^2DAS model in handling polar distortion and generating accurate saliency predictions in ODIs. In future research, we aim to extend our proposed model to the scanpath prediction task in omnidirectional videos. We intend to develop a lightweight scanpath prediction model that accurately infers the trajectory of human fixations. This advancement will contribute to a deeper understanding of the mechanisms behind human visual attention.

Acknowledgment. This work is supported in part by the National Natural Science Foundation of China under Grant 62302337, 62402098, 62377011, in part by the Fundamental Research Funds for the Central Universities under Grant 2232024D-25, in part by the Shanghai Super Doctoral Incentive Program, and in part by the foundation of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, under Grant ESSCKF 2023-03.

References






1. Abdelaziz, Y., Djilali, D., Krishna, T., McGuinness, K., O'Connor, N.E.: Rethinking 360° image visual attention modelling with unsupervised learning. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 15394–15404 (2021)
2. Battisti, F., Baldoni, S., Brizzi, M., Carli, M.: A feature-based approach for saliency estimation of omni-directional images. *Signal Process. Image Commun.* **69**, 53–59 (2018)
3. Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., Durand, F.: What do different evaluation metrics tell us about saliency models? *IEEE Trans. Pattern Anal. Mach. Intell.* **41**, 740–757 (2016)
4. Chao, F.Y., Zhang, L., Hamidouche, W., Déforges, O.: Salgan360: Visual saliency prediction on 360 degree images with generative adversarial networks. In: 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 01–04 (2018)
5. Chen, D., Qing, C., Xu, X., Zhu, H.: Salbinet360: saliency prediction on 360° images with local-global bifurcated deep network. In: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 92–100 (2020)
6. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: Attention over convolution kernels. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11027–11036 (2019)
7. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H.S., Hu, S.: Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 569–582 (2015)
8. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 2990–2999. New York, New York, USA (20–22 Jun 2016)
9. Cornia, M., Baraldi, L., Serra, G., Cucchiara, R.: A deep multi-level network for saliency prediction. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 3488–3493 (2016)

10. Cornia, M., Baraldi, L., Serra, G., Cucchiara, R.: Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Trans. Image Process.* **27**, 5142–5154 (2016)
11. Generative adversarial networks: Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., et al., B.X. *Commun. ACM* **63**, 139–144 (2014)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
13. He, L., Chen, Y., shen, z., Dong, Y., Wang, Y., Lin, Z.: Efficient equivariant network. In: *Advances in Neural Information Processing Systems*, vol. 34, pp. 5290–5302 (2021)
14. He, S., Tavakoli, H.R., Borji, A., Mi, Y., Pugeault, N.: Understanding and visualizing deep visual saliency models. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10198–10207 (2019)
15. Howard, A.G., Sandler, M., et al., G.C.: Searching for mobilenetv3. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1314–1324 (2019)
16. Howard, A.G., et al.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR* **abs/1704.04861** (2017)
17. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
18. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269 (2017)
19. Huang, X., Shen, C., Boix, X., Zhao, Q.: Salicon: reducing the semantic gap in saliency prediction by adapting deep neural networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 262–270 (2015)
20. Jiang, M., Huang, S., Duan, J., Zhao, Q.: Salicon: saliency in context. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1072–1080 (2015)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
22. Lebreton, P.R., Raake, A.: Gbvs360, bms360, prosal: extending existing saliency prediction models from 2d to omnidirectional images. *Signal Process. Image Commun.* **69**, 69–78 (2018)
23. Ling, J., Zhang, K., Zhang, Y., Yang, D., Chen, Z.: A saliency prediction model on 360 degree images using color dictionary based sparse representation. *Signal Process. Image Commun.* **69**, 60–68 (2018)
24. Monroy, R., Lutz, S., Chalasani, T., Smolic, A.: Salnet360: saliency maps for omnidirectional images with cnn. *Signal Process. Image Commun.* **69**, 26–34 (2017)
25. Pan, J., et al.: Salgan: Visual saliency prediction with generative adversarial networks. *ArXiv* **abs/1701.01081** (2017)
26. Rai, Y., Callet, P.L., Guillotel, P.: Which saliency weighting for omni directional image quality assessment? 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2017)
27. Rai, Y., Gutiérrez, J., Le Callet, P.: A dataset of head and eye movements for 360 degree images. In: *Proceedings of the 8th ACM on Multimedia Systems Conference*, p. 205–210. *MMSys'17* (2017)
28. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)

29. Schölkopf, B., Platt, J., Hofmann, T.: Graph-Based Visual Saliency, pp. 545–552 (2007)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
31. Song, Y., et al.: Rinet: relative importance-aware network for fixation prediction. IEEE Trans. Multimedia **25**, 9236–9277 (2023)
32. Tan, M., Chen, B., Pang, R., Vasudevan, V., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2815–2823 (2018)
33. Worrall, D., Welling, M.: Deep scale-spaces: Equivariance over scale. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
34. Wu, B., Wan, A., Yue, X., et al., P.H.J.: Shift: A zero flop, zero parameter alternative to spatial convolutions. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9127–9135 (2017)
35. Xu, M., Yang, L., Tao, X., Duan, Y., Wang, Z.: Saliency prediction on omnidirectional image with generative adversarial imitation learning. IEEE Trans. Image Process. **30**, 2087–2102 (2019)
36. Zhang, J., Sclaroff, S.: Exploiting surroundedness for saliency detection: a Boolean map approach. IEEE Trans. Pattern Anal. Mach. Intell. **38**, 889–902 (2016)



FewConv: Efficient Variant Convolution for Few-Shot Image Generation

Si-Hao Liu¹ , Cong Hu¹  , Xiao-Ning Song^{1,2} , Jia-Sheng Chen³ ,
and Xiao-Jun Wu¹ 

¹ Jiangnan University, Wuxi, China
conghu@jiangnan.edu.cn

² DiTu (Suzhou) Biotechnology Co., Ltd., Suzhou 215000, China

³ School of Information, Shanghai Ocean University, Shanghai 200120, China

Abstract. Generative Adversarial Networks (GANs) can synthesize high-quality images by estimating the latent distribution. However, when face with few-shot image datasets, they often suffer from severe overfitting. Previous solutions have primarily focused on data augmentation, model architecture, and loss functions. This paper proposes to address the instability and overfitting issues from the perspective of the convolution process. To tackle these problems, FewConv is introduced as a plug-and-play alternative to traditional convolutions. FewConv independently learns spatial and channel information, reducing the spatial information that needs to be learned while complexifying the channel information. Specifically, FewConv calculates the variance of channel features at each layer to assess their importance and selects the significant portions for depthwise convolution. For channel information, spatial-to-channel feature transformation is performed before pointwise convolution. This makes pointwise convolution need to learn more diverse channel information. The diverse feature input of FewConv enhances its capacity to combat overfitting. Moreover, using FewConv also reduces network parameters and FLOPs, making the network more compact. To validate the effectiveness of FewConv, extensive experiments were conducted on diverse datasets. Models using FewConv achieved better FID scores and exhibited more stable training processes. FewConv is also applied to recognition training on ResNet and MobileNet, with experimental results demonstrating its effectiveness in recognition tasks.

Keywords: Few-shot generation · Generative adversarial networks · Variant convolution

1 Introduction

Generative Adversarial Networks (GANs) have demonstrated remarkable prowess in generating high-quality and diverse images [11, 12, 27, 45]. For instance, StyleGAN and BigGAN can produce results nearly indistinguishable from real data [15–17]. However, their performance heavily relies on large

amounts of high-quality training data [14, 46]. Simply applying GAN models to few-shot image generation tasks often results in training failures, making it a persistently challenging problem [13, 22]. Initially, researchers attributed model failures on few-shot image datasets to insufficient data [1, 26, 41, 47]. Consequently, most approaches start with pre-trained models on large source datasets, fine-tuning them to align closer with the target dataset [21]. While fine-tuning can mitigate overfitting, methods like DistanceGAN [5] and subsequent works [28, 40] maintain image pair distances during fine-tuning to prevent overfitting. Additionally, incorporating an additional fully connected layer, as seen in MineGAN [35] and MineGAN++ [36], or using adaptor networks [49], as demonstrated in One-shot GAN [43] and WeditGAN [6], can guide fine-tuning and improve generation results. However, these methods often rely on compatible pre-training datasets, limiting their applicability when such datasets are unavailable [28, 48].

Researchers explore diverse strategies, including training methodologies, data augmentation, and model architectures, to combat overfitting in few-shot image generation [9, 24, 30, 33, 34]. However, the convolution process itself has received little attention despite the existence of various convolution variants. Some researchers advocate for altering the receptive fields of convolutions to enable networks to learn more intricate spatial structures [2, 29]. Conversely, others aim to improve convolutional efficiency by reducing redundant parameters [3]. In this study, we propose a novel approach to enhance the convolution process from a fresh perspective. In few-shot image generation tasks, overfitting during training needs addressing. We contend that convolutions inherently possess powerful capabilities for spatial and channel information extraction. However, this excessive capability leads to overfitting phenomena. To tackle these issues, FewConv is introduced. FewConv adeptly weakens the spatial structure learning of convolutions, preventing premature overfitting during training. Specifically, variance is used to evaluate channel feature importance, and top layers with the highest variance are selected for spatial information extraction. Simultaneously, FewConv enriches channel information, making the 1×1 convolution face more complex channel information, thereby reducing overfitting risks. Moreover, FewConv’s generic design facilitates easy deployment as a plug-and-play unit, replacing convolutions without architectural changes or hyperparameter tuning. FewConv enhances image quality and training stability, while substantially reducing network parameters and FLOPs. Evaluations on ResNet and MobileNet confirm FewConv’s effectiveness, maintaining comparable recognition accuracy with reduced parameters and FLOPs. Remarkably, in ResNet-101, FewConv outperforms the original model. This indicates that FewConv is successfully mitigating the risk of convolutional overfitting in a proper way. Our main contributions are summarized as follows:

- We designed a plug-and-play convolution operation called FewConv to replace the original convolution. It exclusively focuses on the most salient spatial features, while simultaneously diversifying the input features across channels.

- Through extensive experimentation, FewConv not only improves the image generation quality in few-shot image generation tasks but also provides a more stable training process and lower overfitting risk.
- FewConv also maintains comparable recognition accuracy to the original models in recognition tasks, while significantly reducing the number of parameters and FLOPs, increasing the model’s compactness.

2 Related Work

2.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs) represent a promising category of deep networks trained adversarially. GANs involve a competition between the generator and discriminator, ultimately converging to a Nash equilibrium where the generated data distribution matches the real distribution. They have made significant strides in tasks like image and video generation, restoration, and segmentation [32, 37, 38, 50]. StyleGAN [16], an extension of CNNs, encompasses various functionalities including unconditional and conditional generation, style transfer, and image editing, showcasing the robust capabilities of GAN models. Despite advancements seen in StyleGAN2 and StyleGAN3 [15, 31], which enhance structure and mitigate artifacts, these GANs heavily rely on large, parameter-rich training datasets. Failure to meet these conditions leads to unstable training and diminished generation capabilities. This study delves into the convolution process to uncover the root causes of overfitting and mode collapse with limited image data and proposes necessary improvements.

2.2 Few-Shot Image Generation

The research on few-shot image generation using GANs has both scientific and practical significance. However, training the discriminator with limited real data can lead to overfitting. To address this issue, most methods use pre-trained models. Then fine-tuning is performed on the pre-trained models [28, 35, 39]. One-shot domain adaptation [42] can be achieved by using lightweight adapters and classification heads on pre-trained models. Additionally, researchers adopt feature fusion as a solution to this problem. However, these techniques require similar semantics between training sets; otherwise, the generated images may exhibit obvious artifacts [7, 8, 18, 19, 44]. Researchers also make further improvements in data augmentation, model constraints, and loss functions to better train models. This work hopes to explore the root causes of GAN failure under few-shot conditions and solve this problem using simple and effective methods.

2.3 Variant Convolution

To develop efficient models for mobile and embedded devices, depthwise separable convolution was introduced in MobileNet [10]. This separation significantly reduces computational load and parameters. However, a straightforward

replacement may lead to accuracy decline. The common approach is to widen the network, but this increases parameters, offsetting benefits. Octave convolution [4] addresses redundant channel information in network features, reducing parameters and enhancing accuracy. MUXConv [25] partitions features into original, sub, and superpixels, enriching feature learning and model expressiveness. PConv [3] extracts essential information through channel pruning and T-shaped kernels, boosting network speed. SCConv [20] tackles redundancy by separating and reconstructing features.

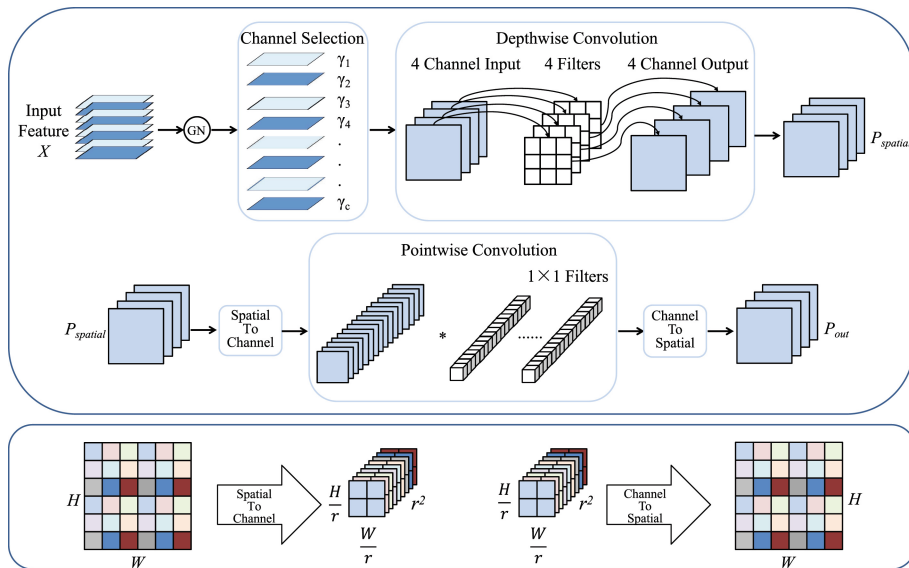


Fig. 1. The convolution processing of FewConv. The top figure is the main convolution process of FewConv, and the bottom figure is the mapping diagram between spatial features and channel features.

3 Methodology

3.1 Motivation

When traditional convolutions are directly applied in few-shot image generation, the discriminator tends to overfit specific image patterns due to their powerful learning capacity. This leads to a lack of diversity in generated images. To address this, it's beneficial to regulate the learning capability of convolutions. Depthwise separable convolution offers a solution by decomposing standard convolution into depthwise and pointwise convolutions, effectively decoupling spatial and channel learning. Leveraging this concept, FewConv is introduced in this work to control convolutional learning capacity.

3.2 Depthwise Separable Convolutions

In the traditional convolutional computation process, each kernel performs a dot product with the entire input feature map. In contrast, depthwise separable convolution divides the convolution process into depthwise convolution and pointwise convolution. The depthwise convolution focuses on extracting spatial information, while the pointwise convolution focuses on extracting channel information. Given an input feature map $P \in \mathbb{R}^{W \times H \times C_{\text{in}}}$, where W is the width of the input feature map, H is the height of the input feature map, and C_{in} is the number of channels in the input feature map, the depthwise convolution processes it with a sliding window.

At each window position, a set of $K \times K$ trainable convolutional kernels, denoted as $W_{\text{depthwise}} \in K \times K \times C_{\text{in}}$, is applied to a patch of the same size. Since depthwise convolution only extracts spatial information from the feature map and does not change the number of channels, the output of the convolution operator remains C_{in} -dimensional, denoted as $O = W_{\text{depthwise}} * P$.

Channel information is then extracted using a computationally efficient 1×1 convolution. This set of trainable 1×1 convolutional kernels is denoted as $W_{\text{point}} \in 1 \times 1 \times C_{\text{out}}$. As the pointwise convolution also needs to serve the purpose of expanding or compressing channels, the output of the convolution operator is a C_{out} -dimensional feature $O = W_{\text{point}} * P$.

3.3 FewConv

Although depthwise separable convolution can effectively reduce FLOPs and the number of network parameters, directly replacing conventional convolution leads to significant accuracy degradation. In order to appropriately constrain the capabilities of the convolutional layer, the input features are preprocessed. Considering the substantial feature redundancy along the channel dimension, a selective approach is adopted to fully exploit both channel redundancy and spatial redundancy. As Fig. 1 illustrated, a channel selection module is introduced with the objective of distinguishing between information-rich and less informative parts along the channel dimension. This is achieved by utilizing affine transformation parameters from the group normalization layer.

Specifically, the group normalization layer is employed to normalize features, with the scaling factor used to assess the importance of different channels. Given an input feature $X \in \mathbb{N} \times W \times H \times C_{\text{in}}$, the input feature is first normalized as follows:

$$\text{BN}(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \quad (1)$$

where μ and σ are the mean and standard deviation of X , ϵ is a small positive constant added for division stability, and γ and β are trainable affine transformation parameters. The scaling factor γ reflects the variance between spatial pixels of each batch and channel. A larger γ indicates richer spatial and channel information, with less redundant information.

Subsequently, based on γ , α channels of feature maps are selected, $P_{choice} \in \mathbb{N} \times W \times H \times C_{in} \times \alpha$. P_{choice} represents the most information-rich part of the entire feature map and ensures that subsequent convolutional layers learn sufficient knowledge for various tasks. In order to minimize the redundant learning of spatial information in the convolutional layer, depthwise convolution operations are performed on the selected, most information-rich part of the feature, denoted as P_{choice} . The output feature is named $P_{spatial}$:

$$P_{spatial} = W_{depthwise} * P_{choice} \quad (2)$$

This operation, which independently calculates the spatial information for each channel, not only effectively reduces the FLOPs of the convolution operation but also avoids overfitting caused by redundant learning.

Certainly, the flow of information between channels is equally important. After independently learning spatial information, the feature maps are rearranged from spatial to channel dimensions. This results in each $r \times r$ spatial feature being sequentially arranged along the channel dimension. Specifically, we define an $r \times r$ window and an input feature $P_{spatial} \in \mathbb{N} \times W \times H \times C_{choice}$. Each feature in this window is sequentially arranged along the channel dimension, and the window then slides with a stride of r until processing the entire feature map. The output feature size is $P_{spatial} \in \mathbb{N} \times \frac{W}{r} \times \frac{H}{r} \times (C_{choice} \times r^2)$.

Subsequently, pointwise convolution is used to process the rearranged feature map, facilitating the flow of information across channels:

$$P_{out} = W_{point} * P_{spatial} \quad (3)$$

Therefore, the following pointwise convolution not only needs to extract channel information but also needs to learn certain spatial information. The diverse combination of information ensures that pointwise convolution does not overfit due to redundant features and a single texture pattern. Finally, the convolved features are restored from the channel dimension back to the spatial dimension. After that, the feature map size is $P_{out} \in \mathbb{N} \times W \times H \times C_{choice}$. Finally, the output features are concatenated with the unconvolved part, and then go through a layer of 1×1 convolution to reach the number of output channels:

$$O = W_{point} * \text{concat}(P_{out}, P_{unconvolved}) \quad (4)$$

3.4 Integrating FewConv Into Backbone Networks

FewConv, based on the spatial and channel separable convolution approach, weakens the spatial extraction capability of depthwise convolution, concentrating it on the most crucial parts. It enhances the learning content of pointwise convolution, no longer limiting its focus solely to channel information. This approach not only facilitates effective learning of the required information for each component but also mitigates overfitting issues arising from redundant information. Thus, FewConv is well-suited for tasks such as few-shot image generation. FewConv is compatible with standard convolutions, seamlessly insertable

into various conventional convolutional networks like ResNet and GAN without requiring special adjustments. For generation tasks, a channel selection rate of $1/4$ is recommended, reducing network flops and parameters while improving the generated output quality and preventing overfitting. For recognition tasks, a $1/4$ channel selection rate still achieves comparable efficiency to the baseline, making the entire network more compact.

It is noteworthy that FewConv lacks channel expansion and compression capabilities. Therefore, after FewConv, pointwise convolution is chosen to employ acting on the entire channel to perform this task. Despite the additional use of a 1×1 convolution, it incurs minimal flops and parameter costs compared to standard convolution. For downsampling operation, specifying the appropriate stride value for depthwise convolution is sufficient. For channels not involved in the FewConv convolution, an average pooling layer is utilized for downsampling, and the outputs are concatenated. Our channel rearrangement is applicable to feature maps of any size $r \times r$ and any scaling rate. However, we advise against using excessively large scaling rates, as they may significantly increase network flops and parameters.

3.5 Efficiency Analysis

FewConv is designed as a plug-and-play module that can be easily embedded into various well-designed neural architectures to reduce computational and storage costs. To illustrate the advantages of FewConv over traditional convolutions more intuitively, we analyze the theoretical reduction in memory usage. The parameters of a standard convolution $Y = M^k X$ can be calculated as:

$$P_s = k \times k \times C_1 \times C_2 = k^2 C_1 C_2 \quad (5)$$

where k is the kernel size of the convolution, C_1 and C_2 are the numbers of input and output feature channels, respectively.

The parameters of the proposed FewConv module are composed as follows:

$$P_{Few} = k \times k \times \alpha C_1 \times g + 1 \times 1 \times \alpha C_1 \times r \times \alpha C_1 \times r + 1 \times 1 \times C_1 \times C_2 \quad (6)$$

where α is the feature selection rate for group normalization, g is the group size for group convolution, r is the transformation rate from spatial to channel, and C_1 and C_2 are the sizes of input and output feature channels, respectively. In experiments, typical parameter settings are $\alpha = 1/4$, $r = 4$, $g = 1$, $k = 3$, and $C_1 = C_2 = C$. The number of parameters can be reduced by 3.5 times, where $P_s/P_{Few} \approx 3.5$, while the performance of the model can be even better than that of standard convolutions.

4 Experiments

In this section, we conducted experiments to assess FewConv’s effectiveness across diverse datasets for both few-shot image generation and image classification tasks. We replaced only the 3×3 kernel in the recognition network with the FewConv module. In the few-shot image generation task, FewConv was also utilized for downsampling operations. To ensure a fair comparison, we replace only the convolutional layers in both tasks, keeping hyperparameters constant between the baseline and our method. All models, including the baseline re-implemented with FewConv, are trained from scratch on NVIDIA 2080TI GPUs using default data augmentation and training strategies, without additional techniques. Multiple training runs with consistent configurations are conducted to mitigate fluctuations, and the median results is reported for each experiment.

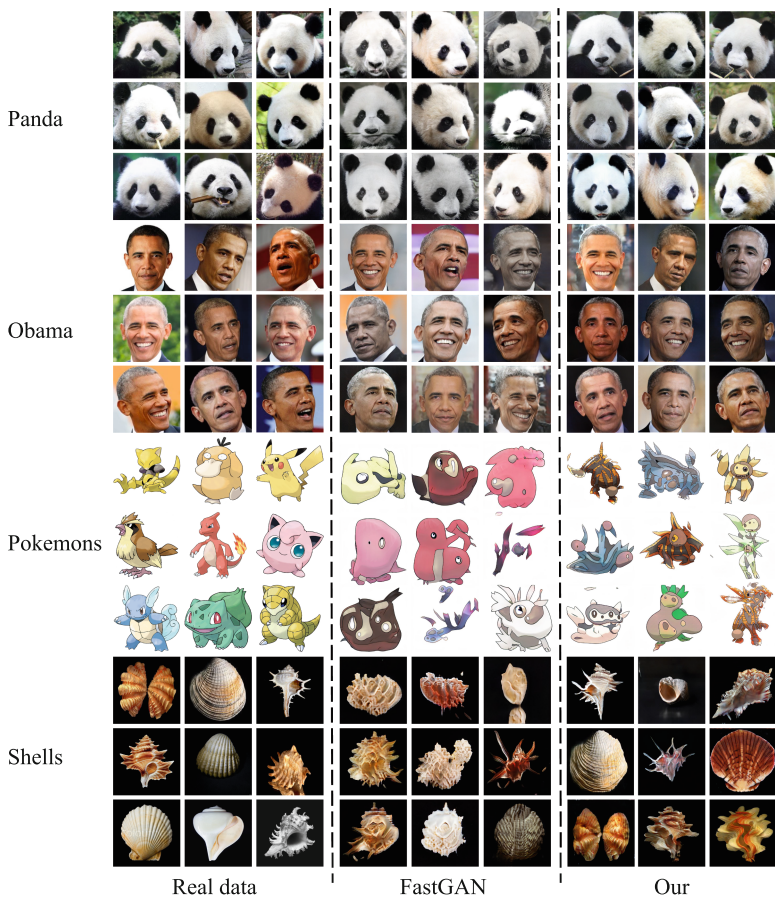


Fig. 2. Qualitative comparison between FastGAN and re-implemented using FewConv. Both models are trained from scratch for 10 h with a batch-size of 8.

Table 1. FID comparison at 256×256 resolution on few-shot datasets.

	Animal Face-Dog	Animal Face-Cat	Obama	Panda	Grumpy-cat
Image number	389	160	100	100	100
styleGAN2	58.85	42.44	46.87	12.06	27.08
styleGAN2 finetune	61.03	46.07	35.75	14.5	29.34
FastGAN	50.66	35.11	41.05	10.03	26.65
Ours	51.28	35.04	38.88	9.85	25.46

Table 2. FID comparison at 1024×1024 resolution on few-shot datasets.

	Skull	Shell	Anime-Face	Pokemon	Art-painting	Flowers	FFHQ
Image number	100	60	120	800	1000	1000	1000
styleGAN2	127.98	241.37	152.73	190.23	74.56	45.23	25.66
styleGAN2 finetune	107.68	220.45	61.23	60.12	N/A	N/A	36.72
FastGAN	130.05	155.47	59.38	57.19	45.08	30.24	30.42
Ours	106.76	98.40	57.84	43.23	40.89	23.98	29.50

Baselines. In the few-shot image generation task, FastGAN [23], a state-of-the-art unconditional few-shot image generation model, is selected as the baseline. Alongside FastGAN, comparison models including StyleGAN2, trained for few-shot image generation with optimal configurations and differentiable data augmentation. Another variant of StyleGAN2 is trained on a large dataset and fine-tuned on a smaller dataset. In FastGAN, all components except the output layer’s convolutional block are replaced by FewConv. Additionally, the downsampling layers in the discriminator utilize depthwise convolution and average downsampling from FewConv for the required operations. For the baseline of image recognition, ResNet and MobileNet, widely used for comparison, are chosen. Popular variants like ResNet-34, ResNet-50, and ResNet-101 are used in the experiments to demonstrate the effectiveness of FewConv. These networks are trained from scratch and optimized using SGD with a cosine learning rate.

Dataset. The experimental setup is as same as FastGAN’s setup and run the experiment on multiple datasets with a wide range of content categories, including Animal-Face Dog and Cat, 100-shot-Obama, 100-shot-panda, and Grumpy-cat, at 256×256 resolution. The 1024×1024 resolution datasets include Flickr Face-HQ(FFHQ), Oxford-flowers, art paintings, photographs of natural landscapes, Pokemon, anime face, skull and shell. We randomly chose 1000 images from the FFHQ dataset for few-image training purposes. As a result, we will

Table 3. Comparison of the number of model parameters and computational complexity between FastGAN and re-implemented using FewConv.

	Params(M)	FLOPs(M)
FastGAN’G	29.13	79271.04
Ours’G	13.42	12680.63
FastGAN’D	8.39	13229.01
Ours’D	5.97	5735.94

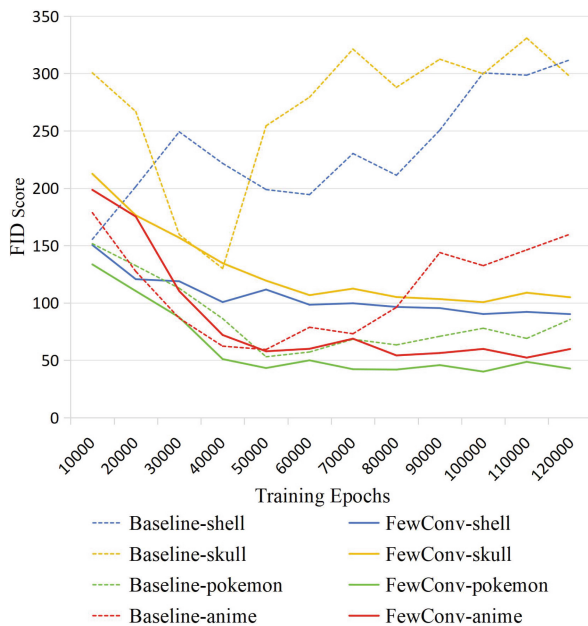


Fig. 3. Trends in FID scores between FastGAN and re-implemented using FewConv.

need to retrain the other models for the FFHQ training dataset in order to achieve comparable FID scores. The image recognition task will employ the CIFAR dataset, which includes CIFAR-10 and CIFAR-100. These datasets consist of 50,000 training images and 10,000 validation images, divided into 10 and 100 classes, respectively.

Metrics. For image generation quality, Frchet Inception Distance is mainly used to evaluate it by measuring the similarity between two datasets of images or Evidence Lower Bound (ELBO) in each dataset. The level of similarity indicates the distance between the generated images and the original images. A lower FID value suggests a better generated image quality. Learned perceptual similarity (LPIPS) is used to measure the similarity between the original and generated

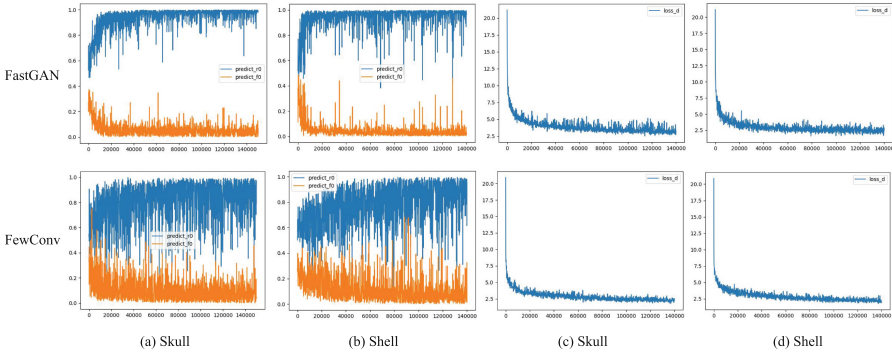


Fig. 4. Comparison of signal output and loss value of discriminator between FastGAN and re-implemented using FewConv. In the left four figures, the blue curve is the signal output of the discriminator to the real image, and the orange curve is the signal output of the discriminator to the fake image. The right four figures show the output of adversarial loss. (Color figure online)

images by calculating the pairwise perceptual distance. Lower LPIPS means greater similarity between two images. We let each class of generating model generates 5,000 images and calculate the FID between the generated image and the entire training dataset. Top-1 accuracy is reported as the evaluation metric for image classification.

4.1 Few-Shot Generation

Figure 2 shows the image generation results. As depicted in Tables 1 and 2, FastGAN with FewConv outperforms the baseline across most datasets at resolutions of 256×256 and 1024×1024 . FewConv not only boosts generative capabilities but also notably reduces computational and parameter Table 3. This highlights redundancy in parameters within deep generative networks, hindering adaptability to few-shot image datasets and leading to overfitting on simple structures. By selectively employing depthwise convolution for spatial learning only on the most representative parts of the features, FewConv effectively mitigate the risk of overfitting, to some extent, at the cost of slightly reducing spatial learning capabilities. By introducing spatial information into channel data in pointwise convolution, FewConv enriches learning content. The increased richness of channel information implies a more diverse input feature stream, which is highly beneficial for harnessing the learning capabilities of pointwise convolution.

To assess the superior anti-overfitting capability of adversarial neural networks with FewConv, four datasets prone to overfitting (Pokemon, anime face, skull, and shell) are selected. The models undergo an additional 60,000 epochs of training, with parameters saved every 10,000 epochs to calculate the FID between generated and real images, as shown in Fig. 3. The FID trends reveal that FewConv-trained models exhibit greater stability, with minimal mode col-

lapse and lower overfitting risk. Even after 60,000 epochs, FewConv models achieve lower FID scores compared to FastGAN, which experiences a notable FID increase post 50,000 epochs. This suggests that networks with traditional convolutions are more susceptible to overfitting and mode collapse. Visualizing the original and FewConv-enhanced discriminator signal outputs, as shown in Fig. 4, we observe that FastGAN outputs near-certainty values (close to 1 for real data and 0 for generated data), indicating potential overfitting around 20,000 epochs. Conversely, FewConv models output values closer to 0.8 for real data and around 0.15 for generated data, facilitating Nash equilibrium. Additionally, monitoring adversarial loss values during training in Fig. 4, shows that FewConv models achieve more stable losses, ensuring a smoother training process and superior outcomes.

Table 4. LPIPS of back-tracking with G.

	FFHQ	Art paintings	Dog Face	Cat Face
Image number	1000	1000	389	160
FastGAN @ 40k iter	2.425	2.624	1.918	1.821
Ours @ 40k iter	2.200	2.465	1.756	1.773
FastGAN @ 80k iter	2.342	2.601	1.986	1.897
Ours @ 80k iter	2.096	2.402	1.750	1.788

A well-trained GAN should be able to reverse a real image to the latent space, with lower levels of overfitting indicating results closer to the real image. Four datasets, AnimalFace-Cat, AnimalFace-Dog, FFHQ, and Art-Paintings, were used for this experiment, and the LPIPS was used to evaluate the final reversal results. The datasets were randomly divided into training and test sets in a 9:1 ratio. The models were trained for 1000 epochs on the training set to prevent the latent vectors from deviating from the distribution. The model parameters were then fixed, and the LPIPS loss was used to update the input random noise. As shown in Table 4, the models using FewConv outperformed FastGAN in terms of reconstruction results, with lower performance loss as the number of iterations increased. This suggests that models using FewConv have lower risk of mode collapse.

4.2 Image Recognition

For CIFAR-10 and CIFAR-100, we adopt a training setup similar to ResNet. The network undergoes 200 epochs of training using the SGD optimizer. At each stage, we set the weight decay to $5 \times e^{-4}$ and momentum to 0.9. The learning rate starts at 0.05 and decays to 0.1 at epochs 100 and 150. Training is performed on a single GPU with a batch size of 128. Each model is trained five times, and reported the median Top-1 accuracy along with FLOPs and parameter quantities for each network. In Table 5, although MobileNet with FewConv experiences

a slight increase in FLOPs and parameters, it achieves accuracy improvements of 1.14% and 1.72% on CIFAR-10 and CIFAR-100, respectively, at a 17% additional computational cost. The parameter increase is mainly due to widening of 1×1 convolutions for spatial to channel mapping. For ResNet, FewConv significantly reduces both FLOPS and parameter quantities, while maintaining comparable accuracy. For ResNet-34, despite a 3.68% accuracy drop on CIFAR-10, we use only 16.7% of the original parameters and 12.1% of the FLOPS. As network depth increases, the accuracy gap decreases. For ResNet-50, the accuracy difference is only about 0.1% to 0.2%, with a 38.4% reduction in FLOPS and a 37.3% reduction in parameters. Finally, for ResNet-100, using only 58.3% of the FLOPS and 61.0% of the parameters, there is an accuracy improvement of 0.21% and 0.4% on CIFAR-10 and CIFAR-100, respectively. FewConv effectively avoids interference from redundant features and parameters in deeper networks by extracting only partial spatial features and allowing 1×1 convolutions to learn more complex representations in the channel domain, thereby enhancing network learning capability.

Table 5. Comparison of parameters, FLOPs, and recognition accuracy between the original model and the model re-implemented using FewConv.

	flops	params	cifar10	cifar100
MobileNet	587.95M	3.22M	91.39	71.39
MobileNet-FC	693.13M	5.97M	92.53	73.11
ResNet34	3678.23M	21.29M	93.31	74.08
ResNet34-FC	445.29M	3.56M	89.63	72.91
ResNet50	4131.72 M	23.53M	92.42	72.90
ResNet50-FC	2545.1 M	14.73 M	92.16	72.76
ResNet101	7864.41M	42.52 M	92.37	72.31
ResNet101-FC	4587.39	25.94 M	92.58	72.71

For the ResNet-34 model, we conduct ablation experiments to validate our explanations and identify causes of accuracy loss. Different div parameters are experimented with to represent the number of channels from which spatial information is learned. As div decreases in Table 6, indicating fewer channels selected, there is no significant accuracy change, indicating spatial redundancy across channels. FewConv uses fewer computations and parameters to acquire spatial information more reasonably, avoiding unnecessary redundancy. Next, we validate the inference that mutual mapping between spatial and channel dimensions enhances the learning capability of 1×1 convolutions. “w/o map” indicates not using the spatial-channel mapping. Without this mapping, accuracy drops by 4%, showing 1×1 convolutions’ powerful learning capability. “w/o down” indicates not using FewConv for downsampling operations. Omitting FewConv for downsampling operations in the ResNet-34 model results in a 51% increase in

Table 6. Ablation experiments on the ResNet-34 model.

	flops	params	cifar10
ResNet34	3678.23M	21.29M	93.31
div = 1	984.42M	15.63M	90.03
div = 2	753.37 M	10.35 M	90.50
div = 3	522.32 M	5.28 M	90.33
div = 4	445.29M	3.56M	89.63
w/o map	445.29M	3.56M	85.28
w/o down	1895.29M	11.26M	92.29
w/o down 50%	1170.29M	7.41M	92.04

FLOPs and parameters, achieving accuracy close to the original model. “w/o down 50%” in Table 6 indicates using FewConv for downsampling only in the second half of the network, resulting in 0.1% difference in accuracy compared to the original network. This is because in shallower networks, the use of average pooling layers results in the loss of too much spatial information, affecting the accuracy of the network, which becomes less significant as the number of layers increases.

5 Conclusion

This paper introduces a plug-and-play convolution module named FewConv, suitable for few-shot image generation tasks. FewConv focuses on learning the most important spatial information from those significant and variable feature parts. By avoiding the learning of redundant features, it reduces the risk of convolutional overfitting on certain spatial information while lowering the computational and parameter storage costs of convolution operations. Additionally, for channel information, a mapping between spatial and channel dimensions is utilized to complexify it, resulting in pointwise convolutions facing more diverse and richer feature inputs. This enhances the expressive learning capability of pointwise convolutions. Through extensive experiments, the use of FewConv stabilizes the few-shot image training process, mitigates overfitting risks, and improves image generation quality. Similarly, we evaluated FewConv’s performance in recognition tasks, where experimental results show that FewConv achieves comparable or better outcomes than the original models at a lower cost of computation and parameters. In the future, we will further explore these intriguing issues and hope that this work can benefit various downstream tasks, providing a new research direction for subsequent studies.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China (Grant No. 62006097, U1836218), in part by the Natural Science Foundation of Jiangsu Province (Grant No. BK20200593), in part by the China Post-doctoral Science Foundation (Grant No. 2021M701456).

References

1. Careil, M., Verbeek, J., Lathuilière, S.: Few-shot semantic image synthesis with class affinity transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 23611–23620 (2023)
2. Chen, J., He, T., Zhuo, W., Ma, L., Ha, S., Chan, S.H.G.: Tvconv: efficient translation variant convolution for layout-aware visual processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12548–12558 (2022)
3. Chen, J., et al.: Run, don't walk: chasing higher flops for faster neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12021–12031 (2023)
4. Chen, Y., et al.: Drop an octave: reducing spatial redundancy in convolutional neural networks with octave convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3435–3444 (2019)
5. Deshpande, I., et al.: Max-sliced wasserstein distance and its use for gans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10648–10656 (2019)
6. Duan, Y., Niu, L., Hong, Y., Zhang, L.: Weditgan: few-shot image generation via latent space relocation. arXiv preprint [arXiv:2305.06671](https://arxiv.org/abs/2305.06671) (2023)
7. Gu, Z., Li, W., Huo, J., Wang, L., Gao, Y.: Lofgan: fusing local representations for few-shot image generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8463–8471 (2021)
8. Hong, Y., Niu, L., Zhang, J., Zhao, W., Fu, C., Zhang, L.: F2gan: fusing-and-filling gan for few-shot image generation. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 2535–2543 (2020)
9. Hou, L.: Regularizing label-augmented generative adversarial networks under limited data. *IEEE Access* **11**, 28966–28976 (2023)
10. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
11. Hu, C., Li, Y., Feng, Z., Wu, X.: Attention-guided evolutionary attack with elastic-net regularization on face recognition. *Pattern Recogn.* 109760 (2023)
12. Hu, Y., Wang, Y., Zhang, J.: Dear-gan: degradation-aware face restoration with gan prior. *IEEE Trans. Circuits Syst. Video Technol.* **33**(9), 4603–4615 (2023). <https://doi.org/10.1109/TCSVT.2023.3244786>
13. Jiang, L., Dai, B., Wu, W., Loy, C.C.: Deceive d: adaptive pseudo augmentation for gan training with limited data. *Adv. Neural. Inf. Process. Syst.* **34**, 21655–21667 (2021)
14. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. *Adv. Neural. Inf. Process. Syst.* **33**, 12104–12114 (2020)
15. Karras, T., et al.: Alias-free generative adversarial networks. *Adv. Neural. Inf. Process. Syst.* **34**, 852–863 (2021)
16. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410 (2019)
17. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)

18. Li, H., Wu, X.J.: Densefuse: a fusion approach to infrared and visible images. *IEEE Trans. Image Process.* **28**(5), 2614–2623 (2019). <https://doi.org/10.1109/TIP.2018.2887342>
19. Li, H., Wu, X.J., Kittler, J.: Mdlatlr: a novel decomposition method for infrared and visible image fusion. *IEEE Trans. Image Process.* **29**, 4733–4746 (2020). <https://doi.org/10.1109/TIP.2020.2975984>
20. Li, J., Wen, Y., He, L.: Sconv: spatial and channel reconstruction convolution for feature redundancy. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6153–6162 (2023)
21. Li, Y., Zhang, R., Lu, J., Shechtman, E.: Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780* (2020)
22. Lin, H., Han, G., Ma, J., Huang, S., Lin, X., Chang, S.F.: Supervised masked knowledge distillation for few-shot transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19649–19659 (2023)
23. Liu, B., Zhu, Y., Song, K., Elgammal, A.: Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In: *International Conference on Learning Representations* (2021)
24. Liu, Z., Song, X., Feng, Z., Xu, T., Wu, X., Kittler, J.: Global context-aware feature extraction and visible feature enhancement for occlusion-invariant pedestrian detection in crowded scenes. *Neural Process. Lett.* **55**(1), 803–817 (2023)
25. Lu, Z., Deb, K., Boddeti, V.N.: Muxconv: information multiplexing in convolutional neural networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12044–12053 (2020)
26. Mangla, P., Kumari, N., Singh, M., Krishnamurthy, B., Balasubramanian, V.N.: Data instance prior (disp) in generative adversarial networks. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 451–461 (2022)
27. Ni, M., Li, X., Zuo, W.: Nuwa-lip: language-guided image inpainting with defect-free vqgan. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14183–14192 (2023)
28. Ojha, U., et al.: Few-shot image generation via cross-domain correspondence. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10743–10752 (2021)
29. Qi, Y., He, Y., Qi, X., Zhang, Y., Yang, G.: Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6070–6079 (2023)
30. Shi, B., Li, W., Huo, J., Zhu, P., Wang, L., Gao, Y.: Global-and local-aware feature augmentation with semantic orthogonality for few-shot image classification. *Pattern Recogn.* **142**, 109702 (2023)
31. Skorokhodov, I., Tulyakov, S., Elhoseiny, M.: Stylegan-v: a continuous video generator with the price, image quality and perks of stylegan2. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3626–3636 (2022)
32. Srivastava, A., Chanda, S., Pal, U.: Aga-gan: attribute guided attention generative adversarial network with u-net for face hallucination. *Image Vis. Comput.* **126**, 104534 (2022)
33. Suzuki, T.: Techaugment: data augmentation optimization using teacher knowledge. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10904–10914 (2022)

34. Tran, N.T., Tran, V.H., Nguyen, N.B., Nguyen, T.K., Cheung, N.M.: On data augmentation for gan training. *IEEE Trans. Image Process.* **30**, 1882–1897 (2021)
35. Wang, Y., Gonzalez-Garcia, A., Berga, D., Herranz, L., Khan, F.S., Weijer, J.V.D.: Minegan: effective knowledge transfer from gans to target domains with few images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9332–9341 (2020)
36. Wang, Y., et al.: Minegan++: mining generative models for efficient knowledge transfer to limited data domains. *Int. J. Comput. Vision* **132**(2), 490–514 (2024)
37. Wu, X., Wang, H., Wu, Y., Li, X.: D3t-gan: data-dependent domain transfer gans for few-shot image generation. *arXiv preprint arXiv:2205.06032* (2022)
38. Xia, G., Luo, D., Zhang, Z., Sun, Y., Liu, Q.: 3d information guided motion transfer via sequential image based human model refinement and face-attention gan. *IEEE Trans. Circuits Syst. Video Technol.* **33**(7), 3270–3283 (2023). <https://doi.org/10.1109/TCSVT.2022.3232330>
39. Xiao, J., Li, L., Wang, C., Zha, Z.J., Huang, Q.: Few shot generative model adaption via relaxed spatial structural alignment (2022)
40. Xiao, J., Li, L., Wang, C., Zha, Z.J., Huang, Q.: Few shot generative model adaption via relaxed spatial structural alignment. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11204–11213 (2022)
41. Xu, J., Liu, B., Xiao, Y.: A variational inference method for few-shot learning. *IEEE Trans. Circuits Syst. Video Technol.* **33**(1), 269–282 (2023). <https://doi.org/10.1109/TCSVT.2022.3199496>
42. Yang, C., et al.: One-shot generative domain adaptation. *arXiv preprint arXiv:2111.09876* (2021)
43. Yang, C., et al.: One-shot generative domain adaptation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7733–7742 (2023)
44. Yang, M., Wang, Z., Chi, Z., Feng, W.: Wavegan: frequency-aware gan for high-fidelity few-shot image generation. In: *European Conference on Computer Vision*, pp. 1–17. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-19784-0_1
45. Yuan, M., Peng, Y.: Bridge-gan: interpretable representation learning for text-to-image synthesis. *IEEE Trans. Circuits Syst. Video Technol.* **30**(11), 4258–4268 (2020). <https://doi.org/10.1109/TCSVT.2019.2953753>
46. Zhang, D., Khoreva, A.: Pa-gan: improving gan training by progressive augmentation (2019)
47. Zhao, M., Cong, Y., Carin, L.: On leveraging pretrained gans for generation with limited data. In: *International Conference on Machine Learning*, pp. 11340–11351. PMLR (2020)
48. Zhao, S., Liu, Z., Lin, J., Zhu, J.Y., Han, S.: Differentiable augmentation for data-efficient gan training. *Adv. Neural. Inf. Process. Syst.* **33**, 7559–7570 (2020)
49. Zhu, X.F., Wu, X.J., Xu, T., Feng, Z.H., Kittler, J.: Robust visual object tracking via adaptive attribute-aware discriminative correlation filters. *IEEE Trans. Multimedia* **24**, 301–312 (2022). <https://doi.org/10.1109/TMM.2021.3050073>
50. Zhu, Y., Zhao, W., Tang, Y., Rao, Y., Zhou, J., Lu, J.: Stableswap: stable face swapping in a shared and controllable latent space. *IEEE Trans. Multimedia*, 1–14 (2024). <https://doi.org/10.1109/TMM.2024.3369853>



FixPix: Fixing Bad Pixels using Deep Learning

Sreetama Sarkar^(✉), Xinan Ye, Gourav Datta, and Peter A. Beerel

University of Southern California, Los Angeles, CA 90089, USA
{sreetama,xinanye,gdatta,pabeerel}@usc.edu

Abstract. Efficient and effective on-line detection and correction of bad-pixels can improve yield and increase the expected lifetime of image sensors. This paper presents a comprehensive Deep Learning (DL) based on-line detection and correction approach, suitable for a wide range of pixel corruption rates. A confidence calibrated segmentation approach is introduced, which achieves nearly perfect bad pixel detection, even with a few training samples. A computationally light-weight correction algorithm is proposed for low rates of pixel corruption, that surpasses the accuracy of traditional interpolation-based techniques. In addition, a vision transformer (ViT) auto-encoder based image reconstruction approach is presented which yields promising results for high rates of pixel corruption or clustered defects. Unlike previous methods, which use proprietary images, we demonstrate the efficacy of the proposed methods on the open-source Samsung S7 ISP and MIT-Adobe FiveK datasets. Our approaches yield up to 99.6% detection accuracy with <0.6% false positives and corrected images within 1.5% average pixel error from 70% corrupted images. We achieve correction error at par with the state-of-the-art (SoTA) DL methods for clustered defects with less than half the computational cost.

Keywords: CMOS image sensor · pixel defect · bad pixel detection · bad pixel correction · deep learning

1 Introduction

There have been remarkable technological advances in the development of CMOS image sensors with improvement in quality, efficiency, and fault-tolerance [20]. Nevertheless, pixel defects can occur in these sensors during the manufacturing process or later during operation, are permanent, and increase in number over the lifetime of the sensor. These defects degrade the sensor yield and effectiveness and consequently increase cost. Pixel defects are important for a wide range of image sensors, but particularly for sensors that are regularly exposed to high levels of light, electrical energy, or radiation, such as in satellites and telescopes, which leads to high rates of pixel corruption.

Supported by Samsung.

Traditionally, pixel defects are detected only during manufacturing [17]. However, the resulting static pixel defect maps do not capture defects developed during the lifetime of the sensor. Online pixel defect detection usually relies either on the analysis of neighborhood pixels within the current frame [2, 29] or on multiple frames [23, 27], rendering them useless when pixel defects occur in nearby pixels or clusters. In this work, we propose bad pixel detection leveraging multiple frames capable of detecting clustered defects without the need to store pixel information from individual frames, thereby eliminating any increased memory overhead while achieving perfect detection.

The detected pixel defects are typically corrected using interpolation algorithms, such as nearest neighbor interpolation [28], linear filtering [19], and median filtering [37]. Traditional approaches are heuristic-based and often tailored for a particular sensor type and error pattern. Clustered defect correction algorithms [22, 32] assume that the defect locations are already known, whereas, defect detection and correction in commercial image signal processors (ISP) are not equipped to detect or correct clustered defects. Sophisticated approaches like adaptive filtering [32] aim to estimate edges and directions and are extremely complicated and harder to optimize. This motivates a learning-based method that is applicable to a wide range of error patterns, error rates and sensor types. Motivated by successes in a wide variety applications, deep learning (DL) have also been explored in the area of pixel defect detection and correction [18, 22].

In this paper, we propose DL based online bad pixel detection and correction on Bayer images suitable for both photographic and computer vision (CV) applications. Our goal is to improve sensor yields during manufacturing as well as increase their effective lifetime. More specifically, we first propose to detect bad pixels, which gives us the error rate in the image. We then propose two different strategies for correcting low and high rates of pixel corruption. For low error rates, we propose a lightweight patch-based pixel correction on extracted patches around the detected bad pixel. For very high error rates and clustered defects, we propose a ML-based complete reconstruction algorithm. We demonstrate results by injecting errors on two different datasets, Samsung S7 ISP [33] and MIT-Adobe FiveK dataset [3] that have RAW Bayer CFA format images. Our approach for detecting and correcting image errors can be easily extended to all types of images, including grayscale, RGB, or IR images.

Contributions. Our contributions can be summarized as follows. (1) We propose a binary segmentation method for effective detection of bad pixels. While this approach achieves nearly perfect detection for large datasets, the detection rate drops for smaller datasets. To mitigate this gap, we propose confidence calibration using multiple images during inference. Our confidence-calibrated segmentation approach yields an improvement of up to 20% over regular binary segmentation. (2) We propose a lightweight patch based pixel correction using multi-layer perceptron (MLP) models for low error rates, that outperforms existing interpolation techniques. More specifically, our MLP model exceeds reported values for Adaptive Defect Correction [35] by 7.05dB and linear [19] and median

[37] interpolation by 4.85dB, for the same error rate. (3) For extremely high rates of pixel corruption, we propose a fail-safe autoencoder based image reconstruction approach, that needs no prior detection. This approach achieves a Normalized Mean Squared Error (NMSE) of 1.55% for up to 70% corrupted pixels. For 5×5 defect clusters, it achieves an NMSE as low as 0.3%, which is at par with SoTA DL approaches [22], with less than half the parameters and computational cost.

2 Background

Bad Pixel Detection: There are two main types of bad pixel detection methods: online and offline. Offline methods detect bad pixels during the manufacturing process, while online methods are used to detect defects throughout the sensor’s lifetime. Traditionally, offline detection involves observing which pixel values remain unchanged across images and creating a map of defective pixels [17]. This map is then stored in a non-volatile memory integrated with the sensor chip to guide downstream pixel correction logic [17]. Online detection is crucial for identifying defects during the lifetime of a sensor, and can be done by analyzing either a single frame or multiple frames. Single-frame detection [2, 4, 7, 29] involves comparing the values of the pixel in question to those of its neighboring pixels. Two commonly used defect detection and correction algorithms in commercial ISP are Pinto [29] and Kakarala [2]. Pinto [29] uses a 3×3 neighborhood and identifies a pixel as defective if it has the highest or lowest value in the neighborhood. Naturally, this method cannot detect multiple defective pixels in the same neighborhood. [2, 4, 7] set upper and lower thresholds based on values of neighboring pixels, and any pixel that falls outside of this range is flagged as defective. [14] performs rule-based analysis of pixel deviation from local average estimates of same color neighbors. None of these methods are equipped to detect clustered defects or multiple bad pixels in close vicinity. [23, 27, 34] uses multi-frame processing to detect defects. [27] uses a combination of neighborhood pixels and temporal consistency. The pixels exceeding the average neighborhood pixel values by a pre-defined threshold are stored as candidate bad pixels and monitored over a number of time steps. Pixels whose value remains unchanged over a sufficient time period are declared defective. [23] uses Bayesian statistics of image sequences collected over days for defect detection. These methods incur a large memory overhead for storing image statistics. More recently, deep learning has also been used for bad pixel detection [18, 25, 39]. Kalyanasundaram et al. [18] proposed a MLP model for detection of isolated defects, although it needs some initial pre-processing steps. [39] uses convolutional neural networks (CNNs) while [25] uses a YOLOv3 [30] based architecture for defect detection.

Bad Pixel Correction: Pixel defect correction is typically performed using interpolation. While nearest neighbor interpolation [28] replaces the defective pixel with its nearest non-defective pixel value in 2D space, linear filtering

[19], and median filtering [37] compute the mean and median of a few non-defective neighboring pixels to replace the defective pixel. Bad pixel correction in Kakarala [2] and Pinto [29] are performed using linear and median filtering, respectively, which leads to image blurring when edges are present in the patch. More advanced interpolation techniques such as Adaptive Defect Correction (ADC) [35] tries to estimate edges and directions for defect correction although it can only work on Bayer pattern images. Sparsity-based Defect Interpolation [32] devises a sparsity based high-complexity iterative algorithm that leverages complex-valued frequency selective extrapolation and outperforms previous interpolation techniques. These correction approaches assume that defect locations are already known during manufacturing. DL approaches have also been explored for pixel defect corrections [5,22]. For pixel-defect correction of flat-panel radiography images, [22] uses different DL approaches: a single-layer ANN, a multi-layer CNN, a concatenated CNN, and GANs, and infer that their concatenated CNN performs the best for correcting clustered defects.

3 Bad Pixel Detection

Semantic segmentation [6,13,24,31] is a common task in computer vision, where each pixel in an image is assigned a specific class. Segmentation is popular for understanding image context in applications like autonomous driving [15] or medical image analysis [31]. We formulate bad pixel detection as a binary segmentation problem consisting of two classes: good pixels and bad pixels (Fig. 1). We perform detection using U-Net [31], originally proposed for medical image segmentation. It has an encoder-decoder architecture making the network U-shaped (as shown in Fig. 1), where the encoder consists of a set of downsampling layers while the decoder consists of a set of upsampling layers. The model takes single channel Bayer images with defective pixels as input and generates a binary map indicating good and bad pixels as output, which has the same dimension as the input. This method can be extended for RGB inputs by training a U-Net model with three input channels instead of one input channel. The model is trained using a combination of binary cross-entropy and dice loss.

However, simply using binary segmentation cannot achieve perfect detection, particularly when there are limited number of images for training the segmentation model (see Sect. 5.2). Due to nature of defects, the corrupted pixels always

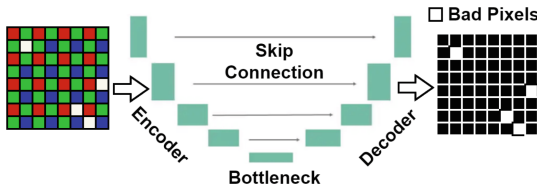


Fig. 1. Bad pixel detection using binary segmentation

occur in the same location across images. A single image may not be enough to correctly identify all bad pixel locations. We leverage the predictions from multiple test images for more reliable bad pixel detection. For semantic segmentation, the model outputs a set of probability or confidence scores indicating if a pixel belongs to a particular class. Instead of taking probability values from a single image, we take mean probability score of n images during test time, which is then thresholded to obtain final class labels, as shown in Fig. 2. This approach is termed as confidence-calibrated segmentation. Our results show significant improvement in detection performance (see Sect. 5.2). Notably, we maintain a pixel-wise cumulated probability score (sum of probability scores) across multiple frames, instead of maintaining individual pixel-wise probabilities for each frame. Therefore, we just need to store the information corresponding to the number of pixels in a single frame, making our method more memory-efficient as compared to existing multi-frame detection approaches.

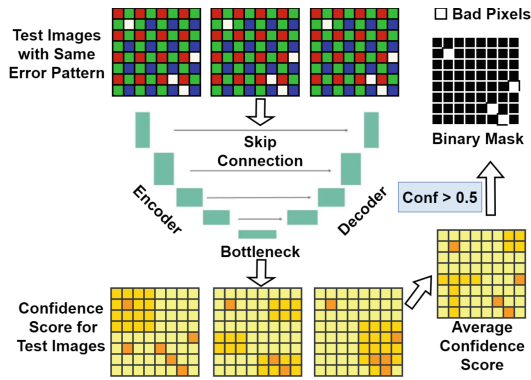


Fig. 2. Bad pixel detection using confidence-calibrated segmentation

4 Bad Pixel Correction

For correction of bad pixels, we propose two different approaches to deal with different error rates. The error rate can be measured using the detection method proposed above. First, we propose a patch-based correction approach, where a $n \times n$ patch around the detected bad pixel is extracted, and passed through the correction network to obtain the actual value of the erroneous central pixel. While this method performs reasonably well for low error rates, it fails when the bad pixels are clustered or the number of bad pixels in a patch is very high. For this, we propose a fail-safe, a Vision Transformer based Autoencoder (ViT AE) [12, 16] for pixel correction using complete image reconstruction.

MLP Based Correction: While bad pixel detection needs to be applied periodically during the lifetime of the sensor (e.g., during the boot-up process), bad pixel correction has to be performed on every single captured image. Hence, the correction algorithm should be preferably lightweight. We build a 2-layer MLP, consisting of 2 fully-connected layers with ReLU activation, to predict the central pixel from neighboring pixel values. The ReLU layers introduce non-linearity, which helps the network better estimate the optimal pixel value and outperform traditional interpolation approaches. We compare our approach with linear [19] and median [37] filtering (Fig. 4) and observe that our approach yields $\sim 14.2\times$ lower NMSE than these methods (Fig. 3).

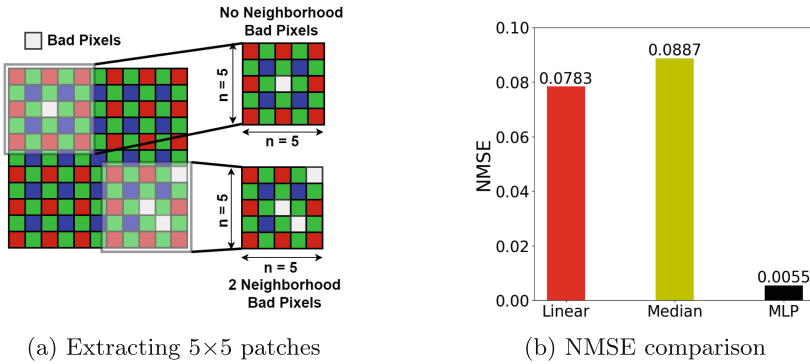


Fig. 3. Patch based bad pixel correction

A $n \times n$ patch may contain multiple bad pixels in the neighborhood of the central pixel, as shown in Fig. 4, which makes the problem of pixel correction harder. We adopt two different approaches to mitigate this problem: *increasing patch size* and *training models with neighborhood bad pixels*. Increasing patch size provides the model with a larger window of neighborhood pixels for prediction of the central pixel, which is particularly advantageous when there are multiple bad pixels in close vicinity. On the other hand, training with corrupted pixels imparts the ability to infer correct prediction discarding defective pixels in the neighborhood. While very effective for low error rates, for high levels of image corruption, these methods experience notable deterioration (see Sect. 5.2), motivating a secondary approach.

Image Reconstruction Using a ViT AE: An Autoencoder (AE) [1, 36] has an encoder-decoder architecture, where the encoder learns the latent features from input images and the decoder reconstructs the image using those latent features. They are used for a wide range of vision tasks, including anomaly detection [1, 40, 41], segmentation [6, 31] and super-resolution [11, 26]. Denoising

autoencoders (DAE) [36] or masked autoencoders (MAE) [16] are used as pre-training for very large models. While DAE injects noise, MAE masks out large portions of the input, and the model learns to reconstruct the image from partial input information, learning robust features.

Unlike these approaches, we use AE with the goal to recover original pixel values from corrupted images. We design a ViT based AE inspired from MAE [16], which takes corrupted single-channel Bayer images as input. However, differing from [16], we do not mask image portions or use mask tokens, but input all embeddings from the corrupted images into the encoder blocks. The AE is trained by minimizing normalized error on the corrupted pixels. We demonstrate that this method, although computationally expensive compared to an MLP and hence unnecessary for low error rates, yields significant benefits for high rates of pixel corruption. More importantly, this method does not require exact bad pixel locations. Therefore, there is no need to perform detection every single time prior to correction, thereby saving detection cost. Moreover, the size of the AE model scales with input size, meaning, based on the model size that can fit into the sensor chip, we can break the input image into patches and perform patch wise reconstruction. For an input of 15×15 , we demonstrate results using an AE model with only 2 encoder and decoder layers consisting of only 11K parameters (Table 2).

5 Experimental Results

5.1 Experimental Setup

Models and Dataset: Our approaches are evaluated on the Samsung S7 ISP [33] and the Canon EOS 5D subset of the MIT-Adobe FiveK dataset [3] datasets. The S7 ISP is a small dataset, consisting of 110 image pairs, captured using the Samsung S7 rear camera, whereas the MIT FiveK is a large-scale dataset, containing 5,000 photographs taken with SLR cameras, from which we extract 777 Canon images, similar to [38]. We consider the raw images for this task and inject bad pixels to the images to evaluate our approaches. The datasets are split into train, validation, and test sets in a ratio of 8:1:1. Detection is performed using U-Net [31] segmentation model, and correction is performed using a 2-layer MLP [21] and ViT AE, inspired from [16].

Bad Pixel Injection: Pixel defects in image sensors have different types. While *dead pixels* are permanently stuck at 0, *hot pixels* or *stuck pixels* maybe permanently bright. Pixel defects may also cause them to deviate from their original value. In our framework, the bad pixel value is obtained by adding at least $\pm\delta$ variation to original pixel value, but still within the permissible range of pixel values. We test our approach over a wide range of δ . The lower the deviation from its original value, the harder it is to detect a bad pixel, whereas, higher the deviation in neighboring pixels, harder it is for correction. The number of bad pixels expected is determined by manufacturing facilities as well as sensor

lifetime models. We test over a wide range of error rates from 0.01% to 85% with the location of bad pixels selected at random.

Training Framework: Both U-Net and ViT AE models are trained for 50 epochs on S7 ISP and 10 epochs on MIT FiveK datasets. For U-Net, we use a step learning rate (lr), starting with a lr of 0.001 and decaying by 0.5 every 10 epochs. ViT AE is trained with an initial lr of 0.01, a linear increase in lr for the first 5 epochs, and cosine decay in lr for the rest of the training epochs. The MLP models for correction are trained for 50 epochs using a learning rate of 0.01. Since raw images have very high dimension ($\sim 3000 \times 4000$), each image is broken down into 64 patches before being fed into U-Net or ViT AE, to reduce computation. They can be decomposed into even smaller patches based on computational constraints.

Evaluation Metrics: The detection approach is evaluated using *Precision* and *Recall*. *Precision* is defined as $TP/(TP + FP)$ and *Recall* is defined as $TP/(TP + FN)$ where TP, FP, and FN refer to true positives, false positives, and false negatives, respectively. In this case, bad pixels are considered positives. Thus, *recall* quantifies the detection rate and *precision* quantifies the false positive ratio. The correction approach is evaluated using NMSE given by $\frac{\|p_{pred} - p_{act}\|_2^2}{\|p_{act}\|_2^2}$ where p_{act} and p_{pred} refer to actual and predicted pixel values. PSNR or peak signal to noise ratio is used to measure the quality of the corrected image with respect to the original image. PSNR is given by $10 \log_{10}(\frac{1}{MSE})$ where MSE is the mean squared error between original and corrected image.

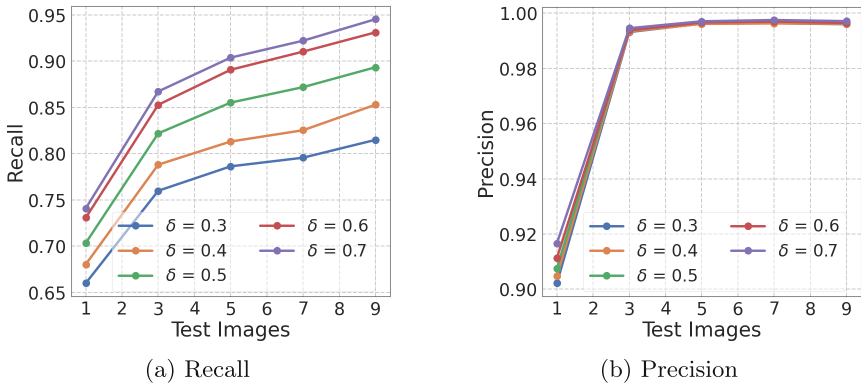
5.2 Results and Analysis

Table 1 summarizes the results for bad pixel detection and correction using the proposed approaches for error rates ranging from 0.01% to 70% and bad pixel values deviating from the original pixel value by 70%. Detection results are reported for a single test image. While for the larger dataset MIT FiveK, we are able to achieve 99.6% detection accuracy, even with a single test image, the obtained detection rate is lower for the smaller S7 ISP dataset. More specifically, we observe a lower recall or detection rate for an error rate of 0.01%. The number of bad pixels in the training set for an error rate of 0.01%, is much smaller than the number of good pixels, resulting in a skewed distribution for the binary segmentation task, which probably leads to poor training and consequently, a performance drop. To mitigate this, we leverage prediction confidence for multiple test images (see Fig. 4). NMSE values are reported for both patch-based correction using an MLP (NMSE_{MLP}) and image reconstruction using an ViT (NMSE_{AE}). The MLP model is applied on 5×5 patches surrounding the pixel to be corrected, whereas, the AE is applied on the entire image, both having the specified error rate. While patch-based pixel correction is effective for low error rates, it suffers up to 31% pixel error when 70% pixels are bad. The AE model successfully reduces this error to 1.55%.

Table 1. Detection and correction results for widely different error rates (with $\delta = 0.7$).

Dataset	Error (%)	Detection		Correction	
		Recall	Precision	$NMSE_{MLP}$	$NMSE_{AE}$
S7 ISP	0.01	0.85	0.96	0.005	0.053
	70	0.95	0.99	0.26	0.098
MIT FiveK	0.01	0.996	0.994	0.0009	0.0036
	70	0.994	0.995	0.31	0.0155

Improving Detection using Confidence Calibration: In Table 1, we observe that for the S7 ISP dataset, we are not able to detect nearly 15% of the bad pixels, even for a δ variation of 0.7. To address this, we use our confidence calibration approach. Figure 4 illustrates precision and recall values for a fixed error pattern and different δ variations, with an increase in the number of images used during inference. A lower δ variation from the original pixel value makes it harder to detect the bad pixel, resulting in lower recall. The increasing trend in precision and recall with increased number of images used during inference reaffirms our hypothesis that using multiple images during test time helps in more reliable pixel detection. We observe that using 9 test images, the maximum number supported by our test set, yields an improvement of $\sim 20\%$ in detection rate, compared with a single test image. Note, the recall values in Table 1 and Fig. 4 are different for same δ and error rate. This is because of the difference in the injected error pattern or the location of the injected bad pixels.

**Fig. 4.** Precision and recall vs # of test images for confidence calibration on S7 ISP dataset (error rate=0.01%)

Correction Using MLP Vs ViT AE: In Fig. 5, we compare patch based pixel correction vs AE based reconstruction for different error rates. MLP models are trained with varying patch sizes with pre-defined error rates. We observe that increasing patch size for MLP models is ineffective when the number of bad pixels scales with patch size. ViT, however, learns from the global image context and maintains a low NMSE even for very high error rates, achieving 2.2% NMSE for 85% corrupted pixels. While ViT AE performs significantly better for high error rates, patch wise detection and correction is more effective for low error rates. This is primarily because local context is more helpful if the neighborhood pixels are not corrupted. However, if there are too many bad pixels in the neighborhood as in case of clustered defects, the global context is more effective in pixel correction. From Fig. 5, we observe that the MLP still performs better for an error rate of 40% on the S7-ISP dataset, while it suffers a small increase on MIT Adobe 5K. Therefore, we set an error rate of 40% as the threshold for switching to AE based reconstruction.

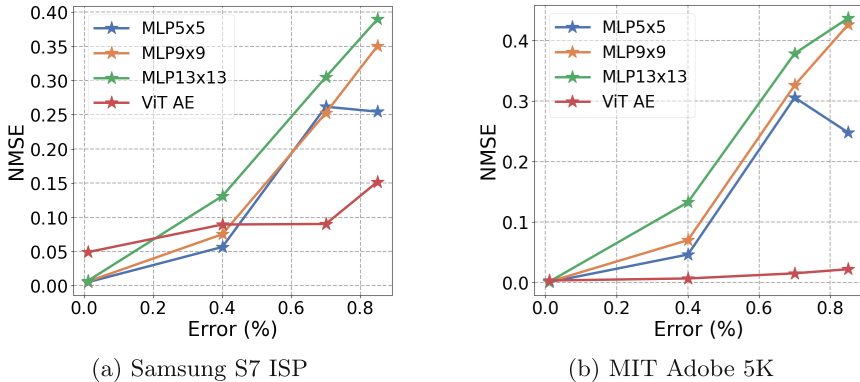


Fig. 5. Comparison of ViT AE and MLP based correction with S7 ISP and MIT FiveK datasets with a wide range of error rates

Comparison with SoTA Interpolation Methods: We compare MLP based pixel correction with existing interpolation techniques for an error rate of 20%, where 5 erroneous pixels are present in a 5×5 patch. Our MLP model achieves a PSNR of 30.55 dB on the S7 ISP dataset, which is higher than reported for all interpolation techniques described in Sect. 2. More specifically, a comparison with the results reported in Table 6 suggests that our model exceeds ADC [35] by 7.05dB and linear [19] and median [37] interpolation by 4.85dB and Sparsity-based Defect Interpolation [32] by 0.15dB. The numbers for the interpolation methods are taken from [32]. Thus, simple learning-based techniques is found to outperform complex rule-based handcrafted interpolation methods (Fig. 6).

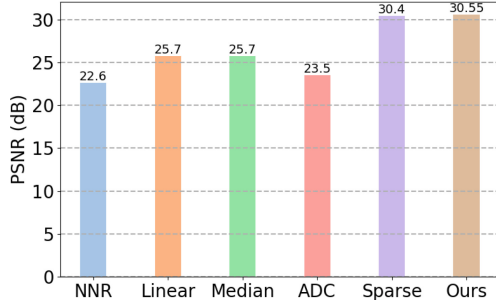


Fig. 6. Comparison of MLP based correction with existing interpolation methods

Comparison with SoTA DL Approaches: In Table 2, we compare our ViT-based reconstruction approach with the results of Concatenate Convolutional Neural Network from the paper [22], which claims to achieve the best performance among the different DL models. We present results on the MIT FiveK dataset using a 5×5 defect cluster in the center of a 15×15 patch, similar to [22]. Since the defect cluster location is known in advance according to the assumption in [22], we divide the image into 5×5 patches and mask the defective patch embedding in the center before sending the embeddings into the encoder. Thus, the value of the defective patch in the center is predicted based on the neighboring 8 patches. Since the input size is only 15×15 , we use only 2 encoder and 2 decoder layers for the ViT model with an embedding length of 16. Remarkably, our method attains similar performance with less than half the parameters and computational cost and does not need the location of the bad pixels. In particular, if we do not know the bad pixel locations, we can simply run our model on all image patches independently, reconstructing the entire image.

Table 2. Comparison of ViT-based reconstruction vs pixel correction using Concatenate CNNs [22] for a 5×5 defect cluster

Method	NMSE	Params	FLOPs
Concat CNN [22]	0.003	26.84 K	203.89 KMac
ViT AE (Ours)	0.004	11.36 K	102.3 KMac

5.3 Ablation Studies

Necessity for Training with Corrupted Pixels: For patch-based pixel correction using MLP (Fig. 5), we assess if models need to be trained with bad pixels in the neighborhood. In Fig. 7(a), we present results for models trained with patches at different error rates. We observe that no single model achieves

the best performance for all error rates. A model trained using no bad pixels in the neighborhood performs poorly when it is fed with a patch containing up to four bad pixels. On the other hand, a model trained with four bad pixels incurs a small increase in error for cases with no neighborhood defects, although it yields relatively low error for all defect rates. Hence, we train separate models for a given error rate.

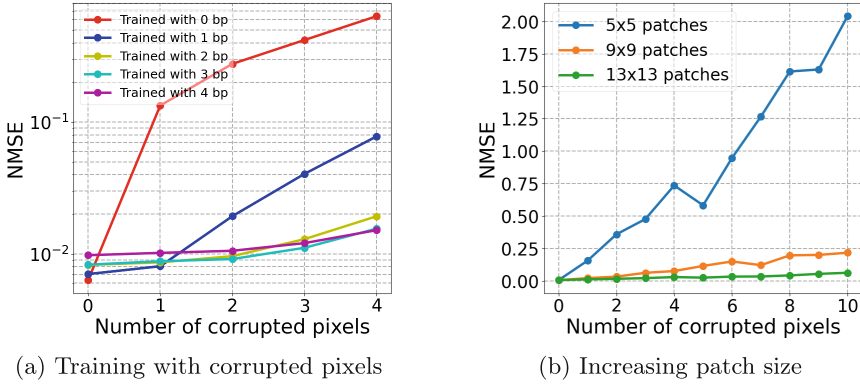


Fig. 7. Patch-based pixel correction on the S7 ISP dataset

Impact of Increasing Patch Size: In Fig. 7(b), we demonstrate results with an increased patch size of 9×9 and 13×13 when there are up to 10 bad pixels in the neighborhood. The models are trained on patches with no bad pixels in the neighborhood and tested on patches with multiple bad pixels. Increasing patch size provides a clear advantage.

6 Summary and Conclusions

This paper presents novel and comprehensive DL based solutions for both the detection and correction of bad pixels for image sensors, for a wide range of error rates, and pixel variations. We achieve detection rate up to 99.6% with less than 0.6% false positives. The correction algorithm yields significantly better results than classical interpolation based approaches. We also offer a fail-safe reconstruction approach for extremely high error rates, which achieves 1.55% average pixel error for 70% corrupted pixels. Our future work includes exploring how the correction algorithm can be combined with in-sensor computing solutions.

Since our pixel detection and correction pipeline operates on the pre-ISP raw images, it can also completely bypass the ISP operations [10], which are typically expensive and performed off-chip. This also enables the pathway for our pipeline to be integrated with existing in-pixel computing paradigms [8,9], that can significantly improve the sensor energy efficiency for CV tasks.

Acknowledgements. We thank Dr. Souvik Kundu for his guidance in this research.

References

1. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. *Spec. Lect. IE* **2**(1), 1–18 (2015)
2. Baharav, I., Kakarala, R., Zhang, X., Vook, D.W.: Bad pixel detection and correction in an image sensing device. uS Patent 6,737,625 (2004)
3. Bychkovsky, V., Paris, S., Chan, E., Durand, F.: Learning photographic global tonal adjustment with a database of input/output image pairs. In: *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition* (2011)
4. Chan, C.H.: Dead pixel real-time detection method for image. uS Patent 7,589,770 (2009)
5. Chen, J., et al.: Hisp: heterogeneous image signal processor pipeline combining traditional and deep learning algorithms implemented on fpga. *Electronics* **12**(16), 3525 (2023)
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(4), 834–848 (2017)
7. Cho, C.Y., Chen, T.M., Wang, W.S., Liu, C.N.: Real-time photo sensor dead pixel detection for embedded devices. In: *2011 International Conference on Digital Image Computing: Techniques and Applications*, pp. 164–169. IEEE (2011)
8. Datta, G., et al.: A processing-in-pixel-in-memory paradigm for resource-constrained TinyML applications. *Sci. Rep.* **12** (2022). <https://api.semanticscholar.org/CorpusID:247318544>
9. Datta, G., et al.: In-sensor & neuromorphic computing are all you need for energy efficient computer vision. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5 (2023). <https://doi.org/10.1109/ICASSP49357.2023.10094902>
10. Datta, G., Liu, Z., Yin, Z., Sun, L., Jaiswal, A.R., Beerel, P.A.: Enabling ISP-less low-power computer vision. In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2429–2438 (2022). <https://api.semanticscholar.org/CorpusID:252815580>
11. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(2), 295–307 (2015)
12. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: *International Conference on Learning Representations* (2021). <https://openreview.net/forum?id=YicbFdNTTy>
13. Drozdal, M., Vorontsov, E., Chartrand, G., Kadoury, S., Pal, C.: The importance of skip connections in biomedical image segmentation. In: Carneiro, G., et al. (eds.) *LABELS/DLMIA -2016. LNCS*, vol. 10008, pp. 179–187. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46976-8_19
14. El-Yamany, N.A.: Robust defect pixel detection and correction for Bayer imaging systems. In: *Digital Photography and Mobile Imaging* (2017). <https://api.semanticscholar.org/CorpusID:63047311>
15. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2012)

16. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16000–16009 (2022)
17. Imperx, I.: Lynx camera series defect pixel correction, application note an-l05 (2006)
18. Kalyanasundaram, G., Pandey, P., Hota, M.: A pre-processing assisted neural network for dynamic bad pixel detection in Bayer images. In: International Conference on Computer Vision and Image Processing (2020). <https://api.semanticscholar.org/CorpusID:233432214>
19. Kovac, M.: Removal of dark current spikes from image sensor output signals. US Patent 3,904,818 (1975)
20. LaPedus, M.: Scaling CMOS image sensors (2020). <https://semiengineering.com/scaling-cmos-image-sensors/>
21. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
22. Lee, E., Hong, E., Kim, D.S.: Using deep learning for pixel-defect corrections in flat-panel radiography imaging. *J. Med. Imaging* **8**, 023501 – 023501 (2021). <https://api.semanticscholar.org/CorpusID:232142409>
23. Leung, J., Chapman, G.H., Koren, I., Koren, Z.: Automatic detection of in-field defect growth in image sensors. In: 2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, pp. 305–313. IEEE (2008)
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
25. Ma, W., Zhang, S., Zheng, Z.: 41.4: display panel defect detection algorithm based on group convolutions. In: SID Symposium Digest of Technical Papers, vol. 50, pp. 463–467. Wiley Online Library (2019)
26. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Adv. Neural Inf. Process. Syst.* **29** (2016)
27. Mijatović, L., Dean, H., Rožić, M.: Implementation of algorithm for detection and correction of defective pixels in fpga. In: 2012 Proceedings of the 35th International Convention MIPRO, pp. 1731–1735. IEEE (2012)
28. Pape, D., Reiss, W.: Defect correction apparatus for solid state imaging devices including inoperative pixel detection. US Patent 5,047,863 (1991)
29. Pinto, V., Shaposhnik, D.: Dynamic identification and correction of defective pixels. uS Patent 8,098,304 (2012)
30. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
31. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
32. Schöberl, M., Seiler, J., Kasper, B., Föbel, S., Kaup, A.: Sparsity-based defect pixel compensation for arbitrary camera raw images. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1257–1260 (2011). <https://api.semanticscholar.org/CorpusID:1496911>
33. Schwartz, E., Giryes, R., Bronstein, A.M.: DeepISP: toward learning an end-to-end image processing pipeline. *IEEE Trans. Image Process.* **28**, 912–923 (2018). <https://api.semanticscholar.org/CorpusID:2356935>

34. Tajbakhsh, T.: Efficient defect pixel cluster detection and correction for bayer cfa image sequences. In: Digital Photography VII, vol. 7876, pp. 174–182. SPIE (2011)
35. Tanbakuchi, A.A., van der Sijde, A., Dillen, B., Theuwissen, A.J.P., de Haan, W.: Adaptive pixel defect correction. In: IS&T/SPIE Electronic Imaging (2003). <https://api.semanticscholar.org/CorpusID:4089006>
36. Vincent, P., Larochele, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103 (2008)
37. Wang, S., Yao, S., Faurie, O., Shi, Z.: Adaptive defect correction and noise suppression module in the CIS image processing system. In: Applied Optics and Photonics China (2009). <https://api.semanticscholar.org/CorpusID:62230334>
38. Xing, Y., Qian, Z., Chen, Q.: Invertible image signal processing. In: CVPR (2021)
39. Ye, R., Pan, C.S., Chang, M., Yu, Q.: Intelligent defect classification system based on deep learning. *Adv. Mech. Eng.* **10**(3), 1687814018766682 (2018)
40. Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., Hua, X.S.: Spatio-temporal autoencoder for video anomaly detection. In: Proceedings of the 25th ACM International Conference on Multimedia, pp. 1933–1941 (2017)
41. Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 665–674 (2017)



Real-World Coarse to Fine-Grained Source-Free Multidomain Adaptation

Anoushka Banerjee^(✉) and Ananth Ganesh

Hitachi Research & Development, Bengaluru, India
{anoushka.banerjee, ananth.ganesh}@hitachi.co.in

Abstract. This work proposes a novel variation to source-free domain adaptation (SFDA) that achieves generalizability through interdomain-intraclass manifold fusion and encoding-guided clustering for classifying new and unseen categories. Multidomain adaptation is accomplished without accessing the source data using two stages; (i) interdomain-intraclass manifold fusion (IMF) and (ii) interclass-cluster-cohesive fine-grained classification (IFC). IMF stage adheres to real-world source data privacy concerns related to proprietary and intellectual property rights. In the IMF stage, intermediate embeddings generated from the source model (without accessing source domain data) are used to propagate source domain knowledge. Instead of linearly combining different classes, which can diminish class discriminability and is non-intuitive in the real world, we introduce interdomain-intraclass embedding mixup to filter the domain invariant class-specific features. The IFC stage enforces strong intraclass cohesion and interclass separation. We conduct our experimental analysis on the fine-grained vehicle detection (FGVD) dataset, a complex and chaotic unconstrained road dataset.

Keywords: Domain Adaptation · Fine-grained · Intraclass Mixup · Multitarget · Open-set Recognition · Source-free

1 Introduction

A usual premise in domain adaptation approaches is the on-demand availability of source data for re-adaptation to target domains [1, 18]. However, in real-world scenarios, access to source data in tandem with off-the-shelf models is often restricted due to proprietary rights, privacy concerns, intellectual property rights, and storage constraints [7, 18]. As a result, a nascent but potential direction for research is source-free domain adaptation, wherein source data is unavailable for re-adapting the source model to target domains. The need for more comprehensive research to mitigate the lack of source data hindering source to target domain knowledge transfer forms the basis for this work.

Adaptability and generalizability are indicators of the adroitness of deep learning algorithms, leading to extensive research related to domain adaptation [4]. However, when deep learning algorithms are deployed in real-world scenarios

they fail in new and unseen environments [4]. This is because most algorithms are tested on datasets captured in controlled environments, lacking real-world randomness and complexities. To overcome this limitation, we use a fine-grained vehicle detection (FGVD) [17] dataset that was captured in the wild using a camera mounted on a car. This dataset precisely captures real-world challenges; complex & chaotic traffic scenarios, extreme interclass similarity and intraclass variance, occlusion, lighting changes, and changes in camera view angle.

To effectively expose algorithms to real-world challenges, it is essential to curate training strategies that are intuitive and align with human cognition. However, in the process of achieving multitarget domain adaptation, some domain adaptation approaches use non-intuitive data augmentation techniques; mixup [18, 31] and CutMix [30], which mix different categories at the input. This mixing of unrelated categories does not accurately reflect real-world scenarios, wherein it is more reasonable to mix within categories across domains. Therefore, in the first stage of our algorithm, as shown in Fig. 1, we introduce interdomain-intraclass manifold fusion (IMF) to increase the generalizability across multiple target domains by enhancing the learning of domain invariant, salient, class-specific features. This stage aligns with human cognition as when we are exposed to different variants of a particular class, we learn to focus on salient features while paying less attention to less specific features. For example, when children see red apples, they associate the color red with apples until they encounter green or golden apples, at which point they begin to focus on the shape of the apple.

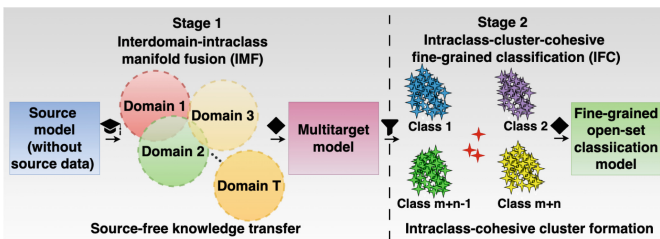


Fig. 1. The overall approach comprises two end-to-end stages; 1: interdomain-intraclass manifold fusion (IMF) stage, and 2: intraclass-cluster-cohesive fine-grained classification (IFC) stage.

In a constantly evolving and dynamically changing real-world scenario, it is imperative to design reliable algorithms. The algorithms can be designed robust to unforeseen changes by imparting the ability to recognise new and unseen objects. After the first stage, we follow the open-set paradigm for tuning the algorithm for coarse to fine-grained classification by introducing intraclass-cluster-cohesive fine-grained classification (IFC) as the second stage. In the second stage, the cluster formation is supervised by apriori assignment of separate cluster centres for each category to ensure better intraclass cluster cohesion and interclass

separability in the latent space, as shown in Fig. 1. Therefore, the major contributions of this work are:

- We propose a novel variation to source-free domain adaptation (SFDA) that achieves generalizability by exploiting interdomain-intra-class manifold fusion and encoding-guided intra-class clustering for classifying new and unseen categories.
- We introduce an IMF strategy. In this approach, intra-class embeddings are linearly combined across multiple target domains. To the best of our knowledge, most approaches leverage mixup [18, 31] or CutMix [30] for combining interclass embeddings or raw image, which is non-intuitive in the real world.
- We introduce a fine-grained open-set IFC stage & amalgamate it to multitarget domain adaptation to ensure robustness to extreme real-world challenges.
- We demonstrate the applicability of class anchoring cluster (CAC) loss [23] for fine-grained open-set recognition for multitarget domain adaptation in real-world scenarios. CAC loss enhances intra-class cluster cohesion and interclass separability.
- We address the issue of extreme class imbalance in the real world and preserve source domain knowledge as a byproduct of the IMF strategy.
- We introduce a practicable algorithm and validate its proficiency through an exhaustive experimental study on real-world data. We verify the suitability of the proposed approach through comparisons with state-of-the-art domain adaptation techniques.

2 Related Works

We conducted a literature review of works related to domain adaptation, which is as brought out:

2.1 Domain Adaptation Approaches Leveraging Data Augmentation/Adversarial Framework

Consistency with nuclear-norm maximization and mixup (CoNMix) [18] leverages interclass mixup [31] strategy for achieving domain adaptation, whereas [13, 29], and [20] augment features using generative adversarial networks (GAN) for enforcing domain invariance. In RevGrad [10], a gradient reversal layer (GRL) is introduced for achieving domain adaptation. The GRL attempts to fool the domain critic by making feature distribution of the source and target domains similar. Moreover, in adversarial discriminative domain adaptation (ADDA) [28] framework, the source and target convolutional neural network (CNN) encoders share weights without being tied, following that the target CNN is adapted to the target domain in an adversarial setup. Wasserstein distance guided representation learning (WDGRL) [26] attempts to improve the adversarial domain adaptation methods by proposing the use of Wasserstein distance [2] to compare the source and target distribution. Wasserstein GAN (WGAN) ensures a

favourable generalization bound and gradient property. However, data augmentation has limitations that the augmented data may not encompass the variations witnessed in the real-world. Additionally, GAN-based or any adversarial approach is susceptible to mode collapse and non-convergence. Except for CoNMix [18], none of the approaches are designed for SFDA.

2.2 Source-Free Domain Adaptation (SFDA)

In response to the practical constraint of restricted access to source data, SFDA is an emerging field of research [1, 7]. CoNMix [18] is amongst the introductory works towards multitarget SFDA techniques. On gleaning through [18], it is felt that mixing two different classes as in CoNMix is non-intuitive. Combining two classes leads to confusion between class decision boundaries, and diminishes the class discrimination ability of the classifier. Instead of adapting to multi-target domains (as in CoNMix [18]), Data frEe multi-sourCe unsupervISed domain adaptatiON (DECISION) [1] uses multiple source domains. However, in the real-world, it is difficult to find multiple off-the-shelf source domain models belonging to different but related domains. In lieu of source data, the approach described in source-free domain adaptation via distribution estimation (SFDA-DE) [7] estimates source distribution using anchors generated from pseudo labels. Pseudo labels are generated by freezing the classification layers, which implicitly assumes that class centres in latent space learned by the source model are well separated. However, this assumption may not hold as categorical cross entropy loss does not guarantee well-separated class-specific clusters [23]. A few approaches counter-balance the unavailability of source data by generating samples using generative adversarial networks (GANs) [19], but generated samples from GANs have less diversity. Hence, GAN-generated samples are not a reliable substitute for the lack of source samples [11].

2.3 Fine-Grained Vehicle Classification

In [16] the data imbalance problem, arising in fine-grained vehicle classification, is addressed using a combination of Faster R-CNN and dense attention network (DAN). Fine-grained vehicle classification is addressed in [27] by using multi-view cameras, and [9] intentionally introduces confusion in activations. However, most of these works perform their experimental analysis using datasets containing fine-grained samples captured in controlled environments. Most objects of interest in videos/images captured in controlled environments are well illuminated, well focused, and proportionately placed at the centre of the frame [17]. Only a handful of works address fine-grained classification in real-world complex scenarios [17]. The dataset in [17] precisely captures real-world challenges such as frequent occlusion of salient features, changes in lighting, out-of-focus objects, variation in image resolution, variation in object size, extreme intraclass variance and interclass similarity.

2.4 Open-Set Recognition

Closed-set recognition in deep learning for object identification and classification contradicts the open-set nature of real-world [5]. Thus, in real-world applications, it is not ideal to forcefully classify a new or unseen test sample as one of the closed set classes. Instead, it should be recognised as a new or unseen class [5,6]. Therefore, for real-world adoption of any domain adaptation approach, it is necessary to follow an open-set classification paradigm. Also, in the purview of domain adaptation, it is more realistic to assume that source and target domains have few common and few uncommon classes [24,25]. Anchor clustering is leveraged in [23] to improve open-set recognition performance. However, there is a need for more exhaustive research in source-free open set recognition in domain adaptation for the real world.

In this process, we comprehend the gaps in existing literature, thus fueling our research towards a real-world adoption of source-free domain adaptation. We culminate our literature survey by identifying the gaps and propose an algorithm to overcome the gaps. We introduce our proposed method in the next section.

3 Proposed Approach

In this section, we discuss our proposed approach for source-free adaptation to multiple target domains. Our proposed approach comprises two end-to-end stages:

1. Interdomain-intraclass manifold fusion (IMF) stage.
2. Intracluster-cluster-cohesive fine-grained classification (IFC) stage.

The proposed approach is illustrated in Fig. 2. We introduce an IMF strategy for distilling knowledge from a source domain to multiple target domains. After learning a generalized multitarget domain classifier, we introduce an IFC stage for open-set fine-grained classification. This stage is designed to handle any dynamical real-world scenario, wherein new and unseen categories may emerge unexpectedly.

3.1 Problem Setting

For addressing the problem of source-free domain adaptation (SFDA), we leverage the source model $F_s(x)$ (without accessing source training data) to obtain an adapted multidomain target model $G_t(x)$; wherein the multiple target domains are denoted by $(\mathbf{t}_1, \dots, \mathbf{t}_T)$ and source domain by \mathbf{s} . After deriving the generalized model $G_t(x)$, we re-train the encoder in a fine-grained open-set paradigm. We include K classes, further split into $M + N + 1$ fine-grained classes where; each target domain \mathbf{t}_i is assumed to have M common categories, N domain-specific categories, and an open class represented by 1 for all new and unseen categories that might unexpectedly emerge in the test bed.

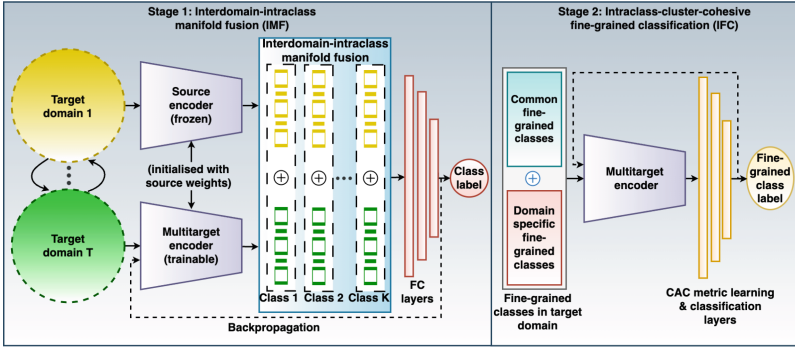


Fig. 2. Proposed approach comprising two end-to-end stages; 1: IMF stage, and 2: IFC stage.

3.2 Overall Framework

The details of the proposed framework is discussed stage-wise:

Stage 1: Interdomain-Intracluster Manifold Fusion (IMF). In this stage we propose to fuse intermediate intracluster and interdomain embeddings i.e., we combine embeddings of the same class but from different domains. In CoNMix [18], first target models are trained for each target domain, and then the knowledge from each target model is distilled into a multitarget model. We simplify the training process by combining two stages in CoNMix [18] into a single IMF stage.

To obtain the multitarget encoder, we replicate the architecture of source model $F_s(x)$, and conjoin it with the source encoder in an untied weight sharing mode. The multitarget encoder branch is trainable while the source branch is frozen. Both branches are initialized using source model weights. The trainable branch learns general features across domains and is adapted as a multitarget domain classifier. The input images from different domains are passed alternately through both branches. This setup keeps the source knowledge intact in the frozen branch while distilling source knowledge to the multitarget domain encoder. To elucidate, the frozen encoder ensures that during training the source knowledge preserved in the source-learned filters is propagated to the multitarget encoder through interdomain-intracluster mixup. Thus, implicitly leading to knowledge distillation and multitarget domain adaptation.

We insert an interdomain-intracluster manifold fusion layer after the conjoined encoders to strategically combine the output from each branch as shown in Fig. 3. The input images belonging to the same class but different domains i.e. interdomain-intracluster embeddings are fused using an improvised mix-up strategy which we name as “cohort” mixup.

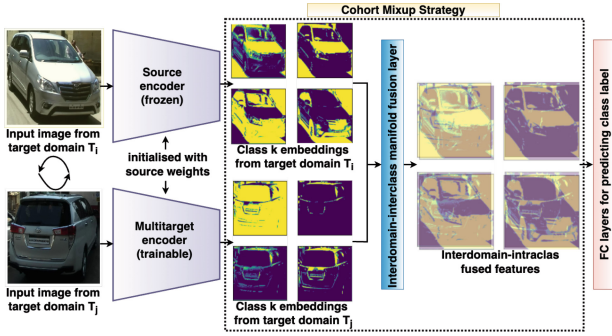


Fig. 3. Illustration of interdomain-intra class feature fusion introduced as cohort mixup. In cohort mixup, embeddings belonging to the same class but different domains are linearly combined. Cohort mixup expands the model view by combining images from different domains.

Cohort Mixup. In general, mixup is performed across classes which is non-intuitive. As it is evident that apples do not compare with oranges and thus, we should not combine the embeddings of apples and oranges. However, to learn a general embedding of apple class we can combine red, golden or green apples from different domains i.e. different geographic locations. By mixing embeddings of apples of different colours the more class-specific feature i.e. shape will become more prominent and the less salient feature i.e. colour will be weighed low.

Therefore, in the proposed cohort mixup we amalgamate the feature embeddings from frozen and trainable branches belonging to the same class but different domains in the interdomain-intra class manifold fusion layer. Let $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]^\top$ and $\mathbf{x}_j = [x_{j1}, x_{j2}, \dots, x_{jd}]^\top$ be the vector representation of feature embeddings obtained for input images belonging to different target domains \mathbf{t}_i and \mathbf{t}_j but from the same class k . Here, d is the size of the feature embedding. We perform cohort mixup on \mathbf{x}_i and \mathbf{x}_j to obtain a domain invariant class-specific feature embedding ε as

$$\varepsilon = \lambda(F_s(\mathbf{x}_i)) + (1 - \lambda)(F'_s(\mathbf{x}_j)), \tag{1}$$

where F_s is the frozen source encoder, F'_s is the weight-updatable branch, and λ is a randomly sampled value from beta distribution [15]. The illustration of cohort mixup operation is depicted in Fig. 4.

The real world exhibits extreme class imbalance, where a few classes have the most samples and most classes have few samples. The extreme class imbalance problem in the real-world can be dealt with as a by-product of cohort mixup.

The output from the interdomain-intra class manifold fusion layer is forwarded to the classification head for identifying the object category.

Single Target Domain Scenario. Our multidomain adaptation technique is equally applicable when only one target domain is available. In such a scenario

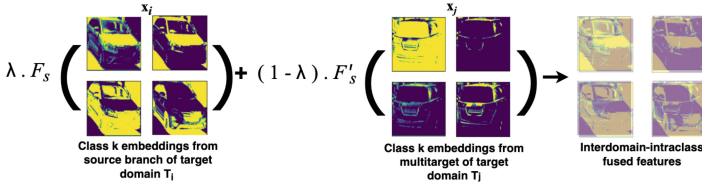


Fig. 4. Illustration of cohort mixup operation.

images from the single domain are fed to both the branches. The input through the trainable branch adapts to the target domain while the knowledge preserved in the source domain encoder is distilled into the target encoder.

Loss Function. Contrary to other approaches, cohort mixup does not mixup labels. Thus, categorical cross-entropy loss without any modifications can be leveraged (unlike [18, 31]). We use categorical cross-entropy loss as the objective function:

$$\mathcal{L}_{class} = - \sum_{i=1}^K y_i \cdot \log \hat{y}_i \quad (2)$$

where K is the number of classes, y_i corresponds to i^{th} value in true class distribution for an input image spanned over K classes, and \hat{y}_i corresponds to i^{th} model output which is predicted probability score for i^{th} class.

Stage 2: Intraclass-Cluster-Cohesive Fine-Grained Classification (IFC). To the best of our knowledge, we did not encounter any approach that introduces open-set fine-grained recognition post source-free multitarget domain adaptation stage. Domain adaptation is achieved in stage: 1, the second stage is added to the framework for endowing the model with finesse to recognise fine-grained categories in an open-set paradigm. Fine-grained classification is defined as classifying within a class, where classes are more specific and distinguished. Thus, inordinately visually similar real-world categories may lie in different classes, rendering fine-grained real-world classification extremely challenging.

Fine-grained classification is achieved by finetuning the multitarget domain model. To impart the model with robustness to identify new and unseen categories, we train following an open-set fine-grained paradigm. Let each domain consist of $M + N + 1$ categories, where M is the number of fine-grained classes common across domains, N is the number of fine-grained categories specific to a particular target domain, and 1 represents an open category. The open category is the collective class for all new and unseen categories that might unexpectedly emerge in the test bed.

We use class anchor clustering (CAC) loss [6] for enforcing class cohesive cluster formation to achieve fine-grained classification. The class cluster formation is encoding guided, ensuring better intraclass cohesion and interclass separation. Let x be the input to multitarget model $G_t(x)$, \mathbf{C} be the set of predefined class centres; $(\mathbf{c}_1, \dots, \mathbf{c}_N)$ (the one-hot encoded label y vector for each class is defined as the class centre), and γ be a hyperparameter, then CAC loss is defined as

$$\mathcal{L}_{CAC}(x, y) = \mathcal{L}_P(x, y) + \gamma \mathcal{L}_A(x, y), \quad (3)$$

where \mathcal{L}_P is triplet loss [6] defined as

$$\mathcal{L}_P(x, y) = \log \left(1 + \sum_{j \neq y}^{M+N} e^{d_y - d_j} \right), \quad (4)$$

and d Euclidean distance between the class center and network logits, \mathcal{L}_A [6] is as defined

$$\mathcal{L}_A(x, y) = d_y = \|G_t(x) - \mathbf{c}_y\|_2. \quad (5)$$

4 Experimental Study and Results

In this section, we discuss our experimental setup, results, and inferences. We use the FGVD dataset, the details of which are described in the subsequent section.

4.1 Fine-Grained Vehicle Detection (FGVD) Dataset and Data Splits

FGVD [17] is a distinct and exclusive dataset captured in the wild, comprising complex and chaotic traffic scenarios. There are six coarse categories: (i) car, (ii) scooter, (iii) motorcycle, (iv) truck, (v) autorickshaw, and (vi) bus. It consists of 5502 dense traffic scenes, with a total of 24450 instances of vehicles and 210 fine-grained labels. The 210 fine-grained labels (comprising of type, manufacturer, and model) are provided only for classes ‘car’, ‘scooter’ and ‘motorcycle’. The fine-grained labels are highly granular, capturing even the most minute and local differences. The challenges in the FGVD dataset are illustrated in Fig. 5. The extreme class imbalance in fine-grained categories for car and motorcycle is shown in Fig. 6 and Fig. 7. It is seen from Fig. 6 and Fig. 7 that few classes comprise most instances and most classes comprise few instances depicting extreme class imbalance. To the best of our knowledge, no fine-grained dataset (except for FGVD) is publicly available with highly precise fine-grained labels for vehicles in tandem with complex traffic scenarios [17]. We crop vehicle instances from video frames for our experimental study. For performing a comprehensive study, we segregate the data into multiple domains A, B, and C. We interchange one amongst domains A, B, and C as source, and treat the rest domains as target domains. Data in each domain is carefully chosen with some randomization to

replicate the real-world scenario of cities of different tiers with vehicles in accordance with demographics and income level. We divide 24450 vehicle images into three parts using a 70:15:15 ratio for training, validation, and testing respectively.



Fig. 5. Illustration of challenges in FGVD dataset: (a) part of the vehicle captured; only a part gets captured due to moving traffic, (b) interclass similarity; many cars appear highly similar and are difficult to identify from the front view, (c) occlusion and multiple objects; the autorickshaw is occluded by a motorcycle, and (d) illumination changes; due to changes in daylight & weather conditions.

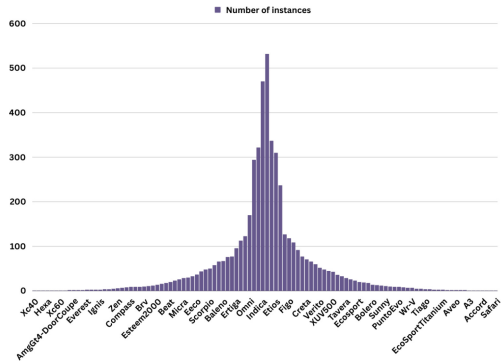


Fig. 6. Number of instances vs. fine-grained categories of ‘car’ depicting high class imbalance. There is a total of 112 fine-grained labels for ‘car’.

4.2 Stage 0: Source Training

This stage is a precursor to the source-free domain adaptation stages. In this stage, we train the source model. After this stage, the source data will not be available for adapting the source model to target domains. The source encoder is a stack of nine convolution layers with ReLU [14] activation function. Each convolutional layer has 3×3 sized kernels, the stride is set to 1 or 2 between

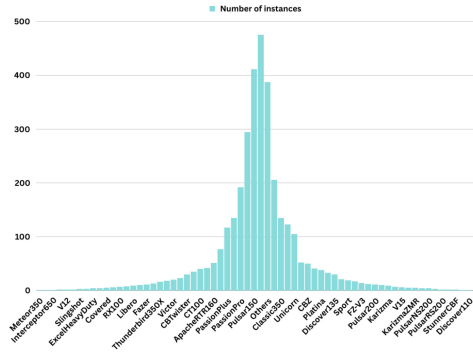


Fig. 7. Number of instances vs. fine-grained categories of ‘motorcycle’ depicting high class imbalance. There is a total of 67 fine-grained labels for ‘motorcycle’.

alternative layers. After each convolution layer, we use a batch normalization layer, and dropout at an interval of three layers. The learning rate is set to 0.0001.

The result of this stage is shown in Table 1. We obtain 90.78%, 81.62%, and 90.02% accuracy for vehicle classification in domains A, B, and C respectively using the backbone chosen for our experimental study. We use ResNet-50 [12] and vision transformer (ViT) [8] for comparisons as these are commonly used backbones [18]. We obtain accuracy in the range 15% - 20% with ResNet-50 and ViT, and hence discard ResNet-50 and ViT backbones for further experimental study. A primary reason for poor performance from ResNet-50 and ViT is poor image resolution and small image size ranging from 28×28 to 64×64 . Due to small image size and low-resolution features dilute in early layers of deep and complex architectures (ResNet-50/ViT) rendering the networks with no gradient for learning. We visualize the feature maps as shown in Fig. 8 and observe that the features become progressively sparse and almost diluted till the fourth convolutional layer of ResNet-50. Additionally, transformer-based architectures are proficient in learning complex pixel-to-pixel relations, and hence suitable for applications wherein background clues aid in recognising objects of interest [8]. However, for identifying the type of vehicle, only the vehicle of interest is to be looked at irrespective of background conditions; number of vehicles in vicinity, daylight, landscape, and weather conditions.

Table 1. Source domain test accuracy (accuracy in %)

Backbone	A	B	C
Ours	90.78	81.62	90.02
ResNet-50	19.49	12.02	16.66
ViT	16.05	15.53	15.71

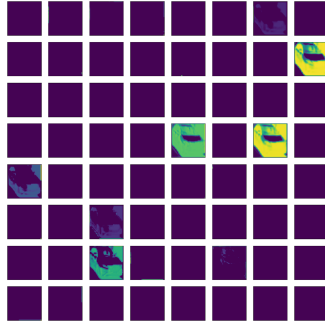


Fig. 8. Feature maps visualized from the fourth convolutional layer of ResNet-50 [12]. The feature maps are sparse and contain little information.

Table 2. Source-free multitarget domain adaptation results (accuracy in %)

Source \rightarrow Target	A \rightarrow B	A \rightarrow C	B \rightarrow A	B \rightarrow C	C \rightarrow A	C \rightarrow B
Ours	93.75	94.94	93.30	91.33	90.11	91.27
CoNMix [18]	39.65	18.34	56.37	48.60	14.77	58.35
SHOT [21]	51.92	50.00	56.55	53.46	51.37	52.03
SHOT++ [22]	61.48	61.33	62.19	62.06	53.08	52.29
WDGRL [26]	55.81	25.58	62.70	51.05	14.37	41.09
ADDA [28]	34.24	18.43	50.65	55.75	32.64	36.37
RevGrad [10]	74.43	60.05	72.58	71.36	73.31	60.19

4.3 Stage 1: Interdomain-Intraclass Manifold Fusion (IMF)

In this stage, in a round-robin manner, we choose one amongst A, B and C to act as the source domain and leverage the corresponding domain model (obtained from stage 0) as the source model. The remaining domains are treated as the target domains. We experiment with all combinations of source and target domains. The results obtained from this stage are presented in Table 2.

We obtain 93.75% and 94.94% accuracy for vehicle classification during source-free multitarget domain adaptation from source A to target domains B and C respectively. We obtain 93.30% and 91.33% accuracy while treating B as the source domain, and A & C as target domains. We obtain 90.11% and 91.27% accuracy while treating C as the source domain, and A & B as target domains. Despite extreme data challenges such as high occlusion, variation in illumination, weather variations, and chaotic traffic, our proposed method yields

promising results for practical usage in real-world complex scenarios. Furthermore, irrespective of the combination of source domain and target domains, the accuracy is consistently greater than 90%. This observation validates that the proposed approach is domain invariant.

Comparison with Other Approaches. We compare the proposed method with CoNMix [18], SHOT [21], SHOT++ [22], WDGRL [26], ADDA [28], and RevGrad [10]. The results of the comparison are presented in Table 2. Upon comparison with CoNMix [18] it is observed that our method clearly outperforms. The reason inferred is that the ViT backbone used in CoNMix [18] is not suitable for real-world datasets. The datasets captured in the wild comprise poor-resolution images and smaller instances of objects. Transformers perform well when coalescing background concepts are necessary for object identification [8]. However, transformers do not work well when objects are to be identified in isolation. For example, in complex traffic scenarios, only the vehicle of interest has to be looked at irrespective of the adjacent vehicles and road conditions. Similarly, in comparison to SHOT [21] and SHOT++ [22] our proposed outperforms at least by a margin of 30% accuracy. SHOT [21] and SHOT++ [22] use the same frozen classifier obtained from the source domain for classifying objects in the target domain resulting in the subpar performance when tested against real-world challenges. A primary reason for this observation is the high reliance on the quality of frozen source classifier leading to lower performance against large domain shifts encountered in the real world. We adapted methods WDGRL [26], ADDA [28] and RevGrad [10] in source-free paradigm for fair comparisons. It is observed from Table 2, that our method clearly stands out. Overall, there is a source domain dependency observed i.e. all other approaches tend to better adapt if the source domain is B. Whereas, our approach is source domain invariant, as consistently across all domains the accuracy is above 90%.

4.4 Stage 2: Intracluster-Cohesive Fine-Grained Classification (IFC)

We achieve source-free domain adaptation in the previous stage. The experiments for this stage are performed to enable the model finesse to recognise fine-grained vehicle categories and to endow the model with open-set recognition ability. We randomly choose certain fine-grained labels to be common across domains, some specific to the domain and some open set. The empirically obtained value of γ is 0.1.

Despite extreme challenges such as extreme class imbalance, high interclass similarity, high intraclass variance, occlusion (as depicted in Fig 5, Fig. 6, and Fig. 7), it is seen from Table 3 that our method performs consistently well across domains and vehicle categories.

Table 3. Stage 2: Open-set fine-grained vehicle classification (accuracy in %)

Fine-grained category	A \rightarrow B	A \rightarrow C	B \rightarrow A	B \rightarrow C	C \rightarrow A	C \rightarrow B
Car	81.08	77.89	74.69	72.72	74.15	73.59
Scooter	81.25	82.85	86.58	80.00	77.64	77.72
Motorcycle	83.33	78.94	66.67	90.00	81.81	82.80

4.5 Ablation Study

We ablate the cohort mixup layer and re-run our experiments. We observe a considerable drop in accuracy of at least 10% as seen in Table 4. Following this, we replace CAC loss [23] with categorical cross-entropy loss and see a drop in accuracy ranging between 10%–30%. From the results of our ablation study, it is reinforced that both cohort mixup and CAC loss are necessary.

Table 4. Ablation study results (accuracy in %)

Source \rightarrow Target	A \rightarrow B	A \rightarrow C	B \rightarrow A	B \rightarrow C	C \rightarrow A	C \rightarrow B
Without Cohort mixup	84.92	70.91	78.41	74.66	60.00	80.29
Without CAC	73.94	68.91	56.37	62.10	61.36	56.18
Ours	93.75	94.94	93.30	91.33	90.11	91.27

The strategically developed IMF & IFC stages overcome the potential limitations: (i) performance deterioration due to large domain shift, (ii) restricted access to source data, (iii) inadequate generalizability, and (iv) high inaccuracy for identification of fine-grained classes with high intraclass variance and high interclass similarity to a significant extent by achieving higher than 90% accuracy in the face of extreme domain shifts as seen from Table 4.

Expanding into Other Application Areas. Source-free domain adaptation is extremely relevant across multiple application areas such as wildlife detection, crop monitoring, worker safety equipment detection, and manufacturing. Domain adaptation and generalization are widely explored within wildlife detection [3, 4]. Wildlife data is captured as a continuous video stream using camera traps, and thus often source data is inaccessible due to large storage requirements [3, 4]. The variations in biodiversity across locations necessitate a multitarget domain adaptation model for wildlife detection and crop monitoring. From an industrial perspective, worker safety equipment such as helmets exhibit high intraclass variance, for example, headgear in mining differs from the biochemical industry. In the case of manufacturing defect detection, the faults (e.g., cracks) are mostly similar, but the industry changes. The experimental study addresses real-world

challenges that are universal across domains, including large domain shifts, low-resolution data, environmental changes, out-of-focus objects, occlusion, varying lighting, high intraclass variance, and high interclass similarity across fine-grained classes. The proposed approach shows promising cross-domain applicability, achieving at least 90% accuracy against these universally applicable real-world challenges.

Our experimental study validates our proposed approach towards a robust, domain invariant, and practicable solution for real-world challenges.

5 Conclusion and Future Scope

In this paper, we propose an approach that supplements the existing works on source-free domain adaptation for real-world environments. Our model employs two end-to-end stages for multidomain data adaptation without using the source data. Despite this, our approach achieves good performance on real-world data and identifies new and unseen data into an open-set category. One of the merits of our approach involves combining separate training stages into a single stage for each target domain. Through cohort mixup, our model has an enhanced view of objects making it more intuitive. The class imbalance issue arising in the real world is dealt with implicitly through cohort mixup. Furthermore, the open-set fine-grained stage provides deeper insights for any real-world statistical study. This work is readily extendible to wildlife detection, worker safety equipment detection, manufacturing, and crop monitoring. In a nutshell, the comprehensive experimental study validates the efficacy of this approach in a real-world setting and corroborates the practicable use of this approach. In future, we aim to integrate eXplainable AI (XAI) with source-free domain adaptation.

References

1. Ahmed, S.M., Raychaudhuri, D.S., Paul, S., Oymak, S., Roy-Chowdhury, A.K.: Unsupervised multi-source domain adaptation without access to source data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10103–10112 (2021)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223. PMLR (2017)
3. Banerjee, A., Dinesh, D.A., Bhavsar, A.: Perusal of camera trap sequences across locations. In: De Marsico, M., Sanniti di Baja, G., Fred, A. (eds.) Pattern Recognition Applications and Methods: 10th International Conference, ICPRAM 2021, and 11th International Conference, ICPRAM 2022, Virtual Event, February 4–6, 2021 and February 3–5, 2022, Revised Selected Papers, pp. 152–174. Springer International Publishing, Cham (2023). https://doi.org/10.1007/978-3-031-24538-1_8
4. Beery, S., Van Horn, G., Perona, P.: Recognition in terra incognita. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 456–473 (2018)
5. Bendale, A., Boulton, T.E.: Towards open set deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1563–1572 (2016)

6. Cao, K., Brbic, M., Leskovec, J.: Open-world semi-supervised learning. arXiv preprint [arXiv:2102.03526](https://arxiv.org/abs/2102.03526) (2021)
7. Ding, N., Xu, Y., Tang, Y., Xu, C., Wang, Y., Tao, D.: Source-free domain adaptation via distribution estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7212–7222 (2022)
8. Dosovitskiy, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
9. Dubey, A., Gupta, O., Guo, P., Raskar, R., Farrell, R., Naik, N.: Pairwise confusion for fine-grained visual classification. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 70–86 (2018)
10. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180–1189. PMLR (2015)
11. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
13. Huang, S.W., Lin, C.T., Chen, S.P., Wu, Y.Y., Hsu, P.H., Lai, S.H.: Auggan: Cross domain adaptation with gan-based data augmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 718–731 (2018)
14. Ide, H., Kurita, T.: Improvement of learning for cnn with relu activation by sparse regularization. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2684–2691 (2017)
15. Johnson, N.L., Kotz, S., Balakrishnan, N.: Continuous univariate distributions, volume 2, vol. 289. John wiley & sons (1995)
16. Ke, X., Zhang, Y.: Fine-grained vehicle type detection and recognition based on dense attention network. *Neurocomputing* **399**, 247–257 (2020)
17. Khoba, P.K., Parikh, C., Jawahar, C., Sarvadevabhatla, R.K., Saluja, R.: A fine-grained vehicle detection (fgvd) dataset for unconstrained roads. In: Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing, pp. 1–9 (2022)
18. Kumar, V., Lal, R., Patil, H., Chakraborty, A.: Conmix for source-free single and multi-target domain adaptation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 4178–4188 (2023)
19. Kurmi, V.K., Subramanian, V.K., Namboodiri, V.P.: Domain impression: A source data free domain adaptation method. In: Proceedings of the IEEE/CVF winter Conference on Applications of Computer Vision. pp. 615–625 (2021)
20. Li, S., Xie, M., Gong, K., Liu, C.H., Wang, Y., Li, W.: Transferable semantic augmentation for domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11516–11525 (2021)
21. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: International Conference on Machine Learning, pp. 6028–6039. PMLR (2020)
22. Liang, J., Hu, D., Wang, Y., He, R., Feng, J.: Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(11), 8602–8617 (2021)
23. Miller, D., Sunderhauf, N., Milford, M., Dayoub, F.: Class anchor clustering: a loss for distance-based open set recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3570–3578 (2021)
24. Panareda Busto, P., Gall, J.: Open set domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 754–763 (2017)

25. Panareda Busto, P., Iqbal, A., Gall, J.: Open set domain adaptation for image and action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(2), 413–429 (2020)
26. Shen, J., Qu, Y., Zhang, W., Yu, Y.: Wasserstein distance guided representation learning for domain adaptation. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)
27. Silva, B., Barbosa-Anda, F.R., Batista, J.: Exploring multi-loss learning for multi-view fine-grained vehicle classification. *J. Intell. Robot. Syst.* **105**(2), 43 (2022)
28. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176 (2017)
29. Volpi, R., Morerio, P., Savarese, S., Murino, V.: Adversarial feature augmentation for unsupervised domain adaptation. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5495–5504 (2018)
30. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: regularization strategy to train strong classifiers with localizable features. In: *Proceedings of the IEEE/CVF International Conference On Computer Vision*, pp. 6023–6032 (2019)
31. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. *arXiv preprint [arXiv:1710.09412](https://arxiv.org/abs/1710.09412)* (2017)

Author Index

A

AbdAlmageed, Wael 356
Ahuja, Nilesh 322
Aizawa, Hiroaki 259
Akpu, Chukwuma Hilary 113

B

Banerjee, Anoushka 456
Baraldi, Lorenzo 147
Beerel, Peter A. 441

C

Chaithra, 47
Chakrabarti, Prantar 18
Chen, Binhan 65
Chen, Jia-Sheng 424
Chen, Jinkun 98
Chen, Song 65
Chu, Lingyang 372
Cornia, Marcella 147
Cucchiara, Rita 147

D

Dai, Weinan 98
Datta, Gourav 441
Datta, Parual 322
Deng, Weihong 226
Dhar, Joy 31
Ding, Shumin 275
Ding, Xiangqian 339
Du, Weidong 289

E

Eswara, Nagabhushan 322

F

Faruque, Omar 164

G

Ganesh, Ananth 456

Goswami, Buddhadev 18

Goyal, Puneet 31

Guan, Donghai 196

Gudi, Ravindra 18

H

Hong, Xia 113

Hu, Cong 424

Hu, Jiani 226

Hussein, Mohamed 356

I

Igaue, Yuki 259

J

Janakiraman, Sudharshan Subramaniam 356

Jiang, Yifeng 98

Jiu, Mingyuan 1

K

Kang, Yi 65

Kanroo, Muhammad Suhaib 31

Kawoosa, Hadia Showkat 31

Kumawat, Sudhakar 243

Kurita, Takio 259

L

Lee, Yeon-Jeong 129

Li, Bingrui 275

Li, Chunlei 275

Li, Fanzhang 289

Li, Shaoxin 372

Lin, Liang 210

Liu, Qian 407

Liu, Ruizhi 181

Liu, Si-Hao 424

Liu, Yuanjing 98

Liu, Zhoufeng 275

Lu, Guangtong 289

M

Mohan, Biju R. 47
 Mostafa, Seraj Al Mahmud 164

N

Nagahara, Hajime 243
 Namboodiri, Anoop 303

P

Patel, Vishal M. 81
 Poppi, Samuele 147
 Pu, Tao 210
 Punjabi, Nirmal 18
 Purushotham, Sanjay 164

R

Remagnino, Paolo 181

S

Sahbi, Hichem 1
 Sarkar, Sreetama 441
 Sarto, Sara 147
 Seong, Joon-Kyung 129
 Sharma, Priyanshu 47
 Shum, Hubert P. H. 181
 Singh, Aniket 303
 Somaraj, Adithya B. 18
 Somayazulu, V. Srinivasa 322
 Song, Xiao-Ning 424
 Song, Yeong-Hun 129
 Sun, Xin 98
 Sydir, Jaroslaw 322

T

Tan, Jianchang 339
 Tao, Jinglei 98
 Tickoo, Omesh 322

W

Wang, Chenxi 164
 Wang, Jianwu 164
 Wang, Mei 226
 Wang, Xiaoxiang 391
 Wang, Xinhua 391
 Wang, Xinyu 210
 Wei, Hong 113
 Wei, Mingqiang 196
 Wu, Qiaojun 65
 Wu, Xiao-Jun 424
 Wu, Xiaoqi 391

X

Xi, Jiangtao 275
 Xu, Liancheng 391

Y

Yang, Shuai 226
 Yang, Xiaokang 407
 Ye, Xinan 441
 Yoo, Sang Wook 129
 Yu, Shusong 339
 Yuan, Weiwei 196
 Yue, Jia 164

Z

Zhai, Guangtao 407
 Zhang, Dongyu 210
 Zhang, Nana 407
 Zhang, Qinglin 196
 Zhou, Mo 81
 Zhu, Dandan 407
 Zhu, Hailong 1
 Zhu, Huanzhang 372
 Zhu, Kun 407