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

LNCS 15325

Pattern Recognition

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

25 Part XXV

ICPR 2024 INDIA

IAPR

Springer

MOREMEDIA

Lecture Notes in Computer Science 15325

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 XXV

Springer

Editors
Apostolos Antonacopoulos
University of Salford
Salford, Lancashire, UK

Subhasis Chaudhuri
Indian Institute of Technology Bombay
Mumbai, Maharashtra, India

Rama Chellappa
Johns Hopkins University
Baltimore, MD, USA

Cheng-Lin Liu
Chinese Academy of Sciences
Beijing, China

Saumik Bhattacharya
IIT Kharagpur
Kharagpur, West Bengal, India

Umapada Pal
Indian Statistical Institute Kolkata
Kolkata, West Bengal, India

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-78388-3 ISBN 978-3-031-78389-0 (eBook)
https://doi.org/10.1007/978-3-031-78389-0

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

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 biosignal 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 IJCPR 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	Indian Statistical Institute, Kolkata, India
Josef Kittler	University of Surrey, UK
Anil Jain	Michigan State University, USA

Program Chairs

Apostolos Antonacopoulos	University of Salford, UK
Subhasis Chaudhuri	Indian Institute of Technology, Bombay, India
Rama Chellappa	Johns Hopkins University, USA
Cheng-Lin Liu	Institute of Automation, Chinese Academy of Sciences, China

Publication Chairs

Ananda S. Chowdhury	Jadavpur University, India
Wataru Ohyama	Tokyo Denki University, Japan

Competition Chairs

Richard Zanibbi	Rochester Institute of Technology, USA
Lianwen Jin	South China University of Technology, China
Laurence Likforman-Sulem	Télécom Paris, France

Workshop Chairs

P. Shivakumara	University of Salford, UK
Stephanie Schuckers	Clarkson University, USA
Jean-Marc Ogier	Université de la Rochelle, France
Prabir Bhattacharya	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 Llados	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. Ambasamudram	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
Mingxin Huang
Lei Kang
Wenhui Liao
Yuliang Liu
Yongxin Shi
Da-Han Wang
Yang Xue
Wentao Yang
Jiaxin Zhang
Yiwu Zhong

Reviewers (Conference Papers)

Aakanksha Aakanksha
Aayush Singla
Abdul Muqeet
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
Anwesha Sengupta
Archisman Adhikary
Arjan Kuijper
Arnab Kumar Das

Arnav Bhavsar
Arnav Varma
Arpita Dutta
Arshad Jamal
Artur Jordao
Arunkumar Chinnaswamy
Aryan Jadon
Aryaz Baradarani
Ashima Anand
Ashis Dhara
Ashish Phophalia
Ashok K. Bhateja
Ashutosh Vaish
Ashwani Kumar
Asifuzzaman Lasker
Atefeh Khoshkhahtinat
Athira Nambiar
Attilio Fiandrotti
Avandra S. Hemachandra
Avik Hati
Avinash Sharma
B. H. Shekar
B. Uma Shankar
Bala Krishna Thunakala
Balaji Tk
Balázs Pálffy
Banafsheh Adami
Bang-Dang Pham
Baochang Zhang
Baodi Liu
Bashirul Azam Biswas
Beiduo Chen
Benedikt Kottler
Beomseok Oh
Berkay Aydin
Berlin S. Shaheema
Bertrand Kerautret
Bettina Finzel
Bhavana Singh
Bibhas C. Dhara
Bilge Gunsel
Bin Chen
Bin Li
Bin Liu
Bin Yao
Bin-Bin Jia
Binbin Yong
Bindita Chaudhuri
Bindu Madhavi Tummala
Binh M. Le
Bi-Ru Dai
Bo Huang
Bo Jiang
Bob Zhang
Bowen Liu
Bowen Zhang
Boyang Zhang
Boyu Diao
Boyun Li
Brian M. Sadler
Bruce A. Maxwell
Bryan Bo Cao
Buddhika L. Semage
Bushra Jalil
Byeong-Seok Shin
Byung-Gyu Kim
Caihua Liu
Cairong Zhao
Camille Kurtz
Carlos A. Caetano
Carlos D. Martã-Nez-Hinarejos
Ce Wang
Cevahir Cigla
Chakravarthy Bhagvati
Chandrakanth Vipparla
Changchun Zhang
Changde Du
Changkun Ye
Changxu Cheng
Chao Fan
Chao Guo
Chao Qu
Chao Wen
Chayan Halder
Che-Jui Chang
Chen Feng
Chenan Wang
Cheng Yu
Chenghao Qian
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 Coquenet
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
Dipanjyoti 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 Binamakhashen
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
Jiyan 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
Liviu-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 Kampel
Martina Pastorino
Marwan Torki
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
Narayan Vetrekar
Narendra D. Londhe
Nathalie Girard
Nati Ofir
Naval Kishore Mehta
Nazmul Shahadat
Neeti Narayan
Neha Bhargava
Nemanja Djuric
Newlin Shebiah R.
Ngo Ba Hung
Nhat-Tan Bui
Niaz Ahmad
Nick Theisen
Nicolas Passat
Nicolas Ragot
Nicolas Sidere
Nikolaos Mitianoudis
Nikolas Ebert
Nilah Ravi Nair
Nilesh A. Ahuja
Nilkanta Sahu
Nils Murrugarra-Llerena
Nina S. T. Hirata
Ninad Aithal
Ning Xu
Ningzhi Wang
Niraj Kumar
Nirmal S. Punjabi
Nisha Varghese
Norio Tagawa
Obaidullah Md Sk
Oguzhan Ulucan
Olfa Mechi
Oliver Tüselmann
Orazio Pontorno
Oriol Ramos Terrades
Osman Akin
Ouadi Beya
Ozge Mercanoglu Sincan
Pabitra Mitra
Padmanabha Reddy Y. C. A.
Palaash Agrawal
Palaiahnakote Shivakumara
Palash Ghosal
Pallav Dutta
Paolo Rota
Paramanand Chandramouli
Paria Mehrani
Parth Agrawal
Partha Basuchowdhuri
Patrick Horain
Pavan Kumar
Pavan Kumar Anasosalu Vasu
Pedro Castro
Peipei Li
Peipei Yang
Peisong Shen
Peiyu Li
Peng Li
Pengfei He
Pengrui Quan
Pengxin Zeng
Pengyu Yan
Peter Eisert
Petra Gomez-Krämer
Pierrick Bruneau
Ping Cao
Pingping Zhang
Pintu Kumar
Pooja Kumari
Pooja Sahani
Prabhu Prasad Dev
Pradeep Kumar
Pradeep Singh
Pranjal Sahu
Prasun Roy
Prateek Keserwani
Prateek Mittal
Praveen Kumar Chandaliya
Praveen Tirupattur
Pravin Nair
Preeti Gopal
Preety Singh
Prem Shanker Yadav
Prerana Mukherjee
Prerna A. Mishra
Prianka Dey
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
Ruowei 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
Siladittya 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 Napolãon
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 Büchner

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

A Novel Loss for Contrastive Deep Supervision

Zhengming Ye[1], Yang Hua[1], Wenjie Zhang[1], Xiaoning Song[1,2](✉), Zhenhua Feng[1], and Xiao-Jun Wu[1]

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China
{6223112039,7211905018,wenjie.zhang}@stu.jiangnan.edu.cn, {x.song,fengzhenhua,wu_xiaojun}@jiangnan.edu.cn

[2] DiTu (Suzhou) Biotechnology Co., Ltd., Suzhou 215000, China

Abstract. Recently, augmentation-based contrastive learning has made significant progress in avoiding hard backpropagation by enhancing supervision signals to optimize intermediate layers. As a well-known observation, same-source augmentations from the same image are more similar than same-class augmentations from different images but in the same class. However, the existing contrastive deep supervision methods ignore the differences in augmentations between the same-source and same-class images, and then neglect the unique information extraction of the same-source, resulting in a reduced model performance. To tackle this limitation, we design a novel module to independently consider same-source and same-class losses, which assists the neural network in understanding the invariance of same-source augmentations and the commonality of same-class augmentations. Furthermore, the proposed module prevents the effect of same-class losses on same-source losses. Experimental results on several standard datasets with ten models show that our proposed method significantly improves the image classification performance of models in both supervised and semi-supervised learning. Code and models will be released at GitHub.

Keywords: Deep Supervision · Contrastive Learning · Image Classification

1 Introduction

With the increasing availability of development frameworks, computational resources, and large-scale datasets [8], Deep Convolutional Neural Networks (CNNs) have become the mainstream backbone for solving various computer vision problems [9,28]. To improve their performance, [7,30] have proposed

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78389-0_1.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 1–16, 2025.
https://doi.org/10.1007/978-3-031-78389-0_1

deeper and more complex CNNs. However, the process of training deeper CNNs is still limited by parameter redundancy or gradient vanishing [17,18]. To address these issues, deep supervision [23,34] is proposed as a solution to the difficulty in optimizing deeper CNNs. Compared to traditional supervised learning methods, which only supervise based on the output of the final layer [18], deep supervision methods train CNNs by introducing the supervised task loss as additional supervision signals in the intermediate layers [23,33]. However, these methods ignore the difference in feature extraction between the shallow and deep layers, where the former focuses on low-level representations such as textures and colors, while the latter represents more task-related semantic information [50]. As presented in [16,48], the application of task-related supervised branches to shallow layers leads to performance degradation of the trained network.

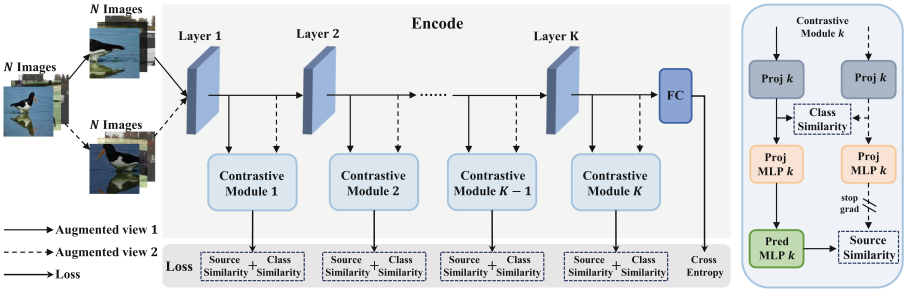


Fig. 1. NCDS architecture. Two augmented views are obtained for each image by performing random data augmentation twice for each image in a batch. The contrastive modules optimize the intermediate layers by maximizing the agreement of same-source views and the agreement of same-class views, respectively. During the inference phase, all contrastive modules are discarded without incurring additional parameters or computations.

Considering that contrastive learning typically learns low-level task-irrelevant data augmentation invariances [2,37], [44] proposed an effective framework named contrastive deep supervision (CDS), which applies contrastive loss to supervise the intermediate layers. In this framework, SimCLR [4] and SupCon [21] are directly adopted to supervise the intermediate layers, the latter has higher classification performance. Unfortunately, CDS did not delve deeper into contrastive learning methods. From a contrastive learning perspective, it is expected that a pair of augmentations from the same image or class will be similar. Remarkably, the similarities of augmentations from the same class are not obvious in their extracted features. For example, even though shelled and unshelled corn belong to the same class, they have different colors and textures. This disparity in similarity may cause the model to acquire distinct knowledge. Specifically, maximizing the similarity between same-source augmentations typically tends to learn the invariance of the various data augmentation. In contrast, maximizing the similarity between same-class augmentations focuses more on learning the common features of the same class.

However, when implementing the CDS, SupCon pulls together representations from the same class and pushes apart those from different classes, which considers these two similarity losses together. This implementation makes the model more focused on learning the similarity between the same class and ignores the knowledge between samples from the same source (the detailed derivation in terms of the gradient is given in Section 3.2). As pointed out in [3], the use of SupCon leads to every augmentation from the same class having the same representation, which significantly reduces the quality of the representation [39]. Especially in shallow layers, the negative impact on feature extraction is more severe. Therefore, it needs to be more refined that the application of SupCon as supervision signals in the intermediate layers.

In this paper, we present a generic training framework, named NCDS, which can be applied to various CNN-based models to significantly improve their classification performance. In this framework, we separate same-source augmentations from same-class augmentations(same-class set in the following exclude same-source augmentations unless otherwise specified). Then, we use the contrastive learning method to design same-source loss and same-class loss, respectively. This enables the intermediate layers to acquire more comprehensive knowledge. As illustrated in Figure 1, we design the contrastive module in each intermediate layer, which independently calculates same-source and same-class losses. For same-source loss, we allow a pair of same-source representations to predict each other, without relying on other representations of the same or different class. Besides, same-class loss contrasts same-class augmentations (excluding same-source augmentations) as positives against the negatives from different classes with the help of dynamic queues. Compared to CDS with SupCon, the design of the contrastive module allows for a reduction in the effect of same-class loss on same-source loss and in the reliance on large batch size. Remarkably, all contrastive modules are discarded during the inference process without adding any additional computational or storage overheads. Furthermore, our approach can be more readily extended to semi-supervised learning methods.

Extensive experiments on several standard datasets using ten different models demonstrate that our proposed method significantly improves image classification performance in both supervised and semi-supervised learning settings. The results also verify that our approach can provide improved supervision for the intermediate layer, helping neural networks to learn better visual representations. Our contributions can be summarized as follows:

- We first observe the difference in similarity between same-source and same-class augmentations. The contrastive module is designed to consider them separately, which helps the model to acquire more comprehensive knowledge.
- We propose a training framework called NCDS, which uses same-source and same-class loss to optimize the intermediate layers to help the model extract better visual representations.

- For large datasets with many classes, dynamic queues are used to store intermediate features for computing same-class losses. This allows the model to achieve excellent performance without relying on a large batch size.
- Extensive experiments on several standard datasets with ten models demonstrate that our proposed method significantly improves the image classification performance of models in both supervised and semi-supervised learning.

2 Related Work

2.1 Contrastive Learning

Recently, contrastive learning has received considerable attention due to its exceptional performance and transferability [20,35,43]. The instance discrimination method is first proposed by InstDisc [36], which uses a memory bank to store a substantial number of negative samples. InvaSpread [40] employs the Siamese network to train positives and negatives from the same mini-batch, thus reducing the storage burden. Then, MOCO [12] uses the momentum update mechanism to build a long and consistent queue of negative samples, which leads to a smaller gap between unsupervised and supervised representation learning. SimCLR [4] proposes a simple Siamese network framework, which benefits from data augmentations, larger batch sizes, and more training steps. In addition, SwAV [1] introduces a scalable online clustering method to contrastive learning, which replaces other larger sets of negative samples with fewer sets of cluster centers. Instead of using negative samples during network training, BYOL [11] proposes using the representation to predict different augmented views from the same image. They have successfully verified the effectiveness of this approach through extensive experimental results. Moreover, SimSiam [6] is introduced to demonstrate that meaningful representations can be learned in a simple Siamese network even without the use of large batches, momentum encoders, and negative sample pairs. And the team also discovers that stop-gradient operation is crucial to preventing collapse. [19] shows that more robust features can be extracted by contrastive models for transfer learning. However, self-contrastive learning tends to learn superficial features that are unrelated to the class, which suppresses the learning of class-relevant ones [5,29,39].

2.2 Supervised Contrastive Learning

Traditional supervised models typically use cross entropy loss, which lacks robustness towards noisy labels [31,49] and the possibility of poor margins [10,26]. To fill this gap, SupCon [21] introduces contrastive learning to supervised models, which pulls representations of the same class together and pushes representations of different classes apart. However, SupCon does not follow the feature extraction process of contrastive learning, which may force anchors and their hard positive samples to be similar. While the anchor and the hard positive samples belong to the same class, they may not be similar. SupCon results in loss of the capacity to distinguish between subclasses in the model, which may lead to class collapse [3,39]. And [3] demonstrates that the transferability of models can be improved by avoiding class collapse. In addition, [39] provides a framework for supervised and unsupervised contrastive learning that could be used to characterize class collapse and feature suppression. Joint self-supervised and supervised learning can mitigate feature suppression and class collapse, but strong theoretical arguments are still lacking.

2.3 Deep Supervision

Deep supervision aims to improve the optimization of deep neural networks, which are complex and contain a large number of layers. Instead of implementing supervision only at the output layer, DSN [23] introduces the companion loss to supervise hidden layers, which makes the learning process for hidden layers more transparent. Subsequently, [34] proposes a strategy to add the deep supervision branches in the appropriate position. On this basis, DKS [32] introduces a synergy loss that integrates pairwise knowledge matching between all supervision branches. DHM [25] proposes Dynamic Hierarchical Mimicking, which optimizes the interaction among the backbone network and supervision branches in different layers. To avoid applying task-related loss supervision directly to shallow layers, CDS [44] proposes a contrastive deep supervision framework that introduces contrastive learning to deep supervision. Furthermore, deep supervision has been widely used in semantic segmentation [27,42,47], knowledge distillation [45], object detection [24], and the construction of neural network frameworks [46].

3 Method

In this section, we first present an overall pipeline of our network framework. Next, we analyse the $\mathcal{L}_{SupCon}$ loss and derive in detail that the contribution of same-source samples to the gradient of $\mathcal{L}_{SupCon}$ is small. Last, we design $\mathcal{L}_{NCDS}$ to better supervise the intermediate layers.

3.1 NCDS Framework

NCDS is a training framework that applies supervision to the intermediate layers. As illustrated in Figure 1, Encoder $F = F_K \circ F_{K-1} \circ \cdots \circ F_1$ can be divided into the number of K layers, each of which is added a contrastive module. In the contrastive module, we use Q_k, G_k, and H_k to denote the projection, projection MLP, and prediction MLP, respectively, in the contrastive module after the k^{th} layer. The final classifier C is in the last layer.

In the training process, a minibatch of N images is randomly sampled for twice stochastic data augmentation, resulting in a set of $2N$ augmented images

$\{x_1, ..., x_N, x_{N+1}, ..., x_{2N}\}$. x_i is an example, and we use the following architecture to extract the k^{th} representations.

$$\begin{aligned} q_{k,i} &= Q_k \circ F_k \circ \cdots \circ F_1(x_i) \\ g_{k,i} &= G_k \circ Q_k \circ F_k \circ \cdots \circ F_1(x_i) \\ h_{k,i} &= H_k \circ G_k \circ Q_k \circ F_k \circ \cdots \circ F_1(x_i) \\ c_i &= C \circ F_K \circ \cdots \circ F_1(x_i), \end{aligned} \tag{1}$$

where ℓ_2 normalized $q_{k,i}$, $g_{k,i}$, and $h_{k,i}$ are the output representations of the k^{th} contrastive module from x_i, c_i is the final classification representation from x_i. Notably, we design a novel contrastive loss function $\mathcal{L}_{NCDS}$, which is based on the loss function of the CDS [44] framework, to train our models. The previous framework had the problem that intermediate layers tend to learn the features of same-class pairs while ignoring same-source pairs. We will provide a comprehensive analysis of these drawbacks in Section 3.2.

In addition, the same-source representations and same-class representations are considered separately to obtain more comprehensive knowledge. Thus, x_i and x_{N+i} are two augmented images from same image, which can be considered as a same-source pair. Define $\hat{P}(i) \equiv \{\hat{p} \in \{1 \ldots 2N\} \backslash \{i, N+i\} : \tilde{y}_{\hat{p}} = \tilde{y}_i\}$ as the set of indices of all augmented images from the same class as x_i, which excludes i and $N+i$. Therefore, x_i and $x_{\hat{p}}$ are considered as same-class pairs, where $\hat{p} \in \hat{P}(i)$. q_i and $q_{\hat{p}}$ is used to calculate the same-class loss of x_i. We will present its details and the calculation of the same-source loss of x_i with h_i and g_{N+i} in Section 3.3.

3.2 Analysis of $\mathcal{L}_{SupCon}$

In this section, we will analyse the demerit of the previous loss function [44]. First, we use $I \equiv \{1 \ldots 2N\}$ to denote the set of indices of all augmented images in a batch and use $q_i = Q_K \circ F(x_i)$ to represent the normalized output of the projection head after the K^{th} layer. Supervised contrastive learning loss [21] can be formalized as follows:

$$\mathcal{L}_{SupCon} = \sum_{i \in I} \mathcal{L}_i^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(q_i \cdot q_p / \tau)}{\sum\limits_{d \in D(i)} \exp(q_i \cdot q_d / \tau)}, \tag{2}$$

where $D(i) \equiv I \backslash \{i\}$ is the set of indices of all augmented images except i, $P(i) \equiv \{p \in D(i) : \tilde{y}_p = \tilde{y}_i\}$ is the set of indices corresponding to all augmented images from the same class as x_i except i, $|P(i)|$ is the cardinality of $P(i)$, $\cdot$ is the dot product symbol, τ is a temperature hyperparameter.

Supervised contrastive learning regards all augmentations with the same label as positives, even if they are from the same image. Intuitively, $\mathcal{L}^{sup}$ aims to pull positives together while pushing negatives away.

However, we observe that the contribution of the same-source representations in the gradient of $\mathcal{L}^{sup}$ is small. Therefore, we argue that $\mathcal{L}^{sup}$ lacks learning on same-source augmentations. Next, we present a detailed derivation of this observation.

We use u_i to denote the unnormalized representation of q_i, which can be formulated as $q_i = u_i / \|u_i\|_2$. Thus, the gradient of $\mathcal{L}^{sup}$ with respect to u_i can be described as follows:

$$\frac{\partial \mathcal{L}_i^{sup}}{\partial u_i} = \frac{\partial q_i}{\partial u_i} \frac{\partial \mathcal{L}_i^{sup}}{\partial q_i}, \tag{3}$$

Firstly, the following derivation can be given:

$$\frac{\partial q_i}{\partial u_i} = \frac{\partial}{\partial u_i}\left(\frac{u_i}{\|u_i\|_2}\right) = \frac{1}{\|u_i\|_2}(\mathrm{I} - q_i q_i^T), \tag{4}$$

where I is the identity matrix. Next, we derive the gradient of the $\mathcal{L}^{sup}$ with respect to q_i as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_i^{sup}}{\partial q_i} &= \frac{-1}{|P(i)|} \sum_{p \in P(i)} \frac{\partial}{\partial q_i} \left\{ \log \frac{\exp(q_i \cdot q_p / \tau)}{\sum\limits_{d \in D(i)} \exp(q_i \cdot q_d / \tau)} \right\} \\ &= \frac{-1}{\tau} \left\{ \sum_{p \in P(i)} \frac{q_p}{|P(i)|} - \frac{\sum\limits_{d \in D(i)} q_d \exp(q_i \cdot q_d / \tau)}{\sum\limits_{d \in D(i)} \exp(q_i \cdot q_d / \tau)} \right\}, \end{aligned} \tag{5}$$

Finally, Equations 3 can be derived by combining Equations 4-5 using the chain rule as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_i^{sup}}{\partial u_i} = \frac{\partial q_i}{\partial u_i} \frac{\partial \mathcal{L}_i^{sup}}{\partial q_i} &= \frac{1}{\tau \|u_i\|_2} \left\{ \frac{\sum\limits_{n \in N(i)} (q_n - q_i\,(q_i \cdot q_n)) \exp(q_i \cdot q_n / \tau)}{\sum\limits_{d \in D(i)} \exp(q_i \cdot q_d / \tau)} \right. \\ &\left. + \sum_{p \in P(i)} (q_p - q_i\,(q_i \cdot q_p)) \left(\frac{\exp(q_i \cdot q_p / \tau)}{\sum\limits_{d \in D(i)} \exp(q_i \cdot q_d / \tau)} - \frac{1}{|P(i)|} \right) \right\}, \end{aligned} \tag{6}$$

where $N\,(i) \equiv \{n \in D\,(i) : \tilde{y}_n \neq \tilde{y}_i\}$ is the set of indices of negatives with different labels as x_i. The detailed derivation of equations 4-6 is given in the Appendix.

With the above content, we could discuss the effect of same-source positive on the gradient of $\mathcal{L}_i^{sup}$. The positives part in Equation 6 is as follows:

$$\frac{1}{\tau \|u_i\|_2} \sum_{p \in P(i)} (q_p - q_i\,(q_i \cdot q_p)) \left(\frac{\exp(q_i \cdot q_p / \tau)}{\sum\limits_{d \in D(i)} \exp(q_i \cdot q_d / \tau)} - \frac{1}{|P(i)|} \right), \tag{7}$$

In general, the same-source pair tends to be more similar than the same-class pair. Thus, assuming that q_i and q_{N+i} are more aligned than q_i and $q_{\hat{p}}$, i.e., $q_i \cdot q_{\hat{p}} \ll q_i \cdot q_{N+i} \to 1$, where $\hat{p} \in \hat{P}(i)$. Then from Equation 7:

$$\|q_p - q_i\,(q_i \cdot q_p)\|_2 = \sqrt{1 - (q_i \cdot q_p)^2}. \tag{8}$$

For same-source and same-class representations, we have $\|q_{\hat{p}} - q_i (q_i \cdot q_{\hat{p}})\|_2 \gg \|q_{N+i} - q_i (q_i \cdot q_{N+i})\|_2 \to 0$ in Equation 8. Hence, the gradient of $\mathcal{L}_i^{sup}$ from the same-source positive in Equation 6 is small, while the gradient from the same-class positives is relatively large. In this instance, the model prefers to learn the class-relevant knowledge from same-class positives rather than the class-irrelevant knowledge from same-source positives. Therefore, we believe that directly using $\mathcal{L}_{SupCon}$ to supervise the intermediate layers may not allow the model to learn comprehensive knowledge. However, the proposed novel loss function $\mathcal{L}_{NCDS}$ could effectively fill this gap, and its details are presented in Section 3.3.

3.3 The Novel Loss

To reduce the effect of same-class positives on the gradient of same-source positives, we separate the learning process into two independent tasks, including for same-source pairs and same-class pairs, respectively.

First of all, the loss function $\mathcal{L}_{class}$ for same-class positives is defined as:

$$\mathcal{L}_{class} = \begin{cases} \sum\limits_{i \in I} \frac{-1}{\left|\hat{P}(i)\right|} \sum\limits_{\hat{p} \in \hat{P}(i)} \log \frac{\exp(q_i \cdot q_{\hat{p}}/\tau)}{\sum\limits_{\hat{d} \in \hat{D}(i)} \exp(q_i \cdot q_{\hat{d}}/\tau)}, & \left|\hat{P}(i)\right| > 0 \\ 0, & \left|\hat{P}(i)\right| = 0 \end{cases} \tag{9}$$

Where $\left|\hat{P}\,(i)\right|$ is the cardinality of $\hat{P}\,(i)$, $\hat{D}(i) \equiv D(i) \backslash \{N+i\}$ is a set of indices which contains all indices of augmented images except i and $N+i$. $\mathcal{L}_{class}$ is a variant of $\mathcal{L}_{SupCon}$(Equation 2), the main difference between $\mathcal{L}_{SupCon}$ and $\mathcal{L}_{class}$ is that $\mathcal{L}_{SupCon}$ regards same-source representations as positives, whereas $\mathcal{L}_{class}$ ignores them.

It is worth noting that $\mathcal{L}_{class}$ is too small to provide effective supervision when the batch size is much smaller than the number of classes. Specifically, too small a batch size may result in a lack of same-class representations for contrasting. To address this issue, dynamic queues are designed in the intermediate layers to store 128-dimensional same-class representations. The dynamic queues store the representations of the samples in the previous iteration and use these stored features to support training in the following iteration. This strategy eliminates the need for $\mathcal{L}_{class}$ to rely on a large batch size. We also present the ablation experiments on the length of the queue in Section 4.

Then we minimize $\mathcal{L}_{source}$ to learn the similarity between same-source pair. More specifically, $\mathcal{L}_{source}$ directly predicts g_{N+i} from h_i to avoid using other representations, including same-class positives or negatives. Following [6], $\mathcal{L}_{source}$ can be formulated as:

$$\mathcal{L}_{source} = \sum_{i=1}^{N} -\frac{1}{2} \left\{h_i \cdot sg\left(g_{N+i}\right) + h_{N+i} \cdot sg\left(g_i\right)\right\}, \tag{10}$$

where $sg\left(\cdot\right)$ is the stop-gradient operation, which means that g_i and g_{N+i} are treated as constant. By combining Equations 9-10, the total loss $\mathcal{L}_{NCDS}$ can be formulated as:

$$\mathcal{L}_{NCDS} = \mathcal{L}_{CE}\left(\{c_i\}_{i=1}^{N}, \{\tilde{y}_i\}_{i=1}^{N}\right) + \sum_{k=1}^{K}\left\{\lambda_1\mathcal{L}_{class}\left(\{q_{k,i}\}_{i=1}^{2N}, \{\tilde{y}_i\}_{i=1}^{2N}\right) + \lambda_2\mathcal{L}_{\text{source}}\left(\{h_{k,i}\}_{i=1}^{2N}, \{g_{k,i}\}_{i=1}^{2N}\right)\right\} \tag{11}$$

where $\mathcal{L}_{CE}$ is the cross entropy loss, which only supervises the model in the final layer. The second part, including $\mathcal{L}_{class}$ and $\mathcal{L}_{source}$, is the loss of contrastive supervision for the number of K intermediate layers. λ_1 and λ_2 are the hyperparameters.

Based on the above formulation about NCDS, our approach can be easily extended to semi-supervised learning. Assuming that dataset $\mathcal{W}$ contains labels $\mathcal{Y}$ and an unlabeled dataset $\mathcal{U}$. For labeled data, it is possible to train directly with the $\mathcal{L}_{NCDS}$ of our approach. For unlabeled data, we only train with $\mathcal{L}_{source}$ that does not require label information. The semi-supervised learning loss can be formulated as:

$$\mathcal{L}_{semi} = \mathcal{L}_{NCDS}\left(\mathcal{W}, \mathcal{Y}\right) + \mathcal{L}_{source}\left(\mathcal{U}\right). \tag{12}$$

Table 1. Evaluating image classification with different deep supervision approaches on CIFAR100.

Method	RST18	RST50	RST101	RXN50	RXN101	WRT50	WRT101	SRN18	SRN50	PRN18
Base	77.45	77.81	78.65	79.85	80.67	79.46	79.98	77.46	78.02	76.84
DSN	78.30	78.96	79.37	81.02	81.70	80.98	81.30	78.28	79.46	77.40
DKS	78.96	80.95	81.39	82.27	82.98	81.95	82.58	79.32	80.76	78.96
DHM	78.82	81.12	81.27	82.14	83.27	81.76	82.76	79.14	80.72	78.32
CDS	80.84	81.31	83.12	82.81	83.87	82.28	83.93	80.13	81.51	80.76
Ours	**82.06**	**84.96**	**85.38**	**85.14**	**85.48**	**85.49**	**85.91**	**82.13**	**84.59**	**81.94**

4 Experiment

We evaluate our method by comparing with other deep supervision methodologies [23,25,32,44] through various neural networks, including ResNet(RST) [13], ResNeXt(RXN) [38], Wide ResNet(WRT) [41], SENet(SRN) [15], PreAct ResNet(PRN) [14]. The experimental results on CIFAR100 [22] for quantitative comparison are reported in Table 1. Our approach significantly outperforms previous methods, with an average performance gain of 2.3% over CDS in terms of top-1 classification. In particular, our approach outperforms CDS by 3.65% on ResNet50. Besides, we report the evaluation results on CIFAR10 in Table 2,

Table 2. Evaluating image classification with different deep supervision approaches on CIFAR10.

Method	RST18	RST50	RST101	RXN50	RXN101	WRT50	WRT101	SRN18	SRN50	PRN18
Base	94.96	95.07	95.13	95.09	95.34	95.01	95.27	94.86	95.11	94.78
DSN	95.31	95.41	95.63	95.39	95.70	95.27	95.78	95.21	95.41	95.13
DKS	95.72	95.90	96.21	95.98	96.10	95.50	96.12	95.74	95.72	95.47
DHM	95.61	95.87	96.04	96.10	96.27	95.62	96.31	95.59	95.77	95.38
CDS	96.49	96.78	97.02	96.76	97.05	96.88	97.01	96.50	96.73	96.37
Ours	**97.01**	**97.53**	**97.51**	**97.39**	**97.56**	**97.70**	**97.61**	**96.98**	**97.26**	**97.03**

Table 3. Evaluating image classification with different deep supervision approaches on ImageNet.

Metric	Model	Baseline	DSN	DKS	DHM	CDS	**Ours**
top-1	RST18	69.21	69.54	71.32	71.29	72.85	**73.58**
	RST34	73.17	73.29	74.01	73.89	76.19	**77.30**
	RST50	75.30	75.37	76.47	76.57	78.25	**79.52**
top-5	RST18	89.01	88.87	89.20	90.06	91.30	**91.61**
	RST34	91.24	91.30	91.87	91.66	93.08	**93.68**
	RST50	92.20	92.49	93.60	93.24	93.99	**94.67**

which shows that our method improves on CDS by 0.6% in terms of average classification accuracy. Furthermore, the quantitative comparison on ImageNet [8] is reported in Table 3, Our approach achieves top-1 accuracy improvements of 0.73%, 1.11%, and 1.27% on ResNet18, ResNet34, and ResNet50, respectively.

Table 4. Ablation study of the auxiliary contrastive loss at intermediate layers on the CIFAR100 dataset with ResNet18.

Loss	accuracy / %
$\mathcal{L}_{SupCon(\text{CDS})}$	80.84
$\mathcal{L}_{SimCLR}$	78.19
$\mathcal{L}_{SimCLR}+\mathcal{L}_{class}$	79.54
$\mathcal{L}_{source}$	78.86
$\mathcal{L}_{source}+\mathcal{L}_{class_{(\text{Ours})}}$	**82.06**

Ablation Study of Auxiliary Loss We compare the classification accuracy of supervised intermediate layers when using various contrastive loss functions. Each contrastive loss is tested on the CIFAR-100 dataset with ResNet18. From the Table 4, we observe that relying solely on self-supervised contrastive loss, such as $\mathcal{L}_{SimCLR}$ and $\mathcal{L}_{source}$, to supervise intermediate layers is not effective enough. This could be attributed to the fact that the intermediate layers have not learned the similarities between the same class. In addition, simply adding the self-supervised contrastive loss and the supervised contrastive loss together (e.g. $\mathcal{L}_{SimCLR}+\mathcal{L}_{class}$) may result in a decrease in performance compared to the singular use of $\mathcal{L}_{SupCon}$. This phenomenon may be due to conflicting optimization objectives. More specifically, $\mathcal{L}_{SimCLR}$ considers same-class representations (excluding same-source representations) as negatives, while $\mathcal{L}_{class}$ considers them as positives. Thus, the optimization goal of $\mathcal{L}_{SimCLR}$ is to push these representations apart, while the optimization goal of $\mathcal{L}_{class}$ is to pull them together. Since $\mathcal{L}_{source}$ does not consider negative samples, $\mathcal{L}_{source}+\mathcal{L}_{class}$ avoids this conflict. Finally, we note that $\mathcal{L}_{source}+\mathcal{L}_{class}$ achieves the best classification performance due to the more comprehensive knowledge learned and the absence of conflicts.

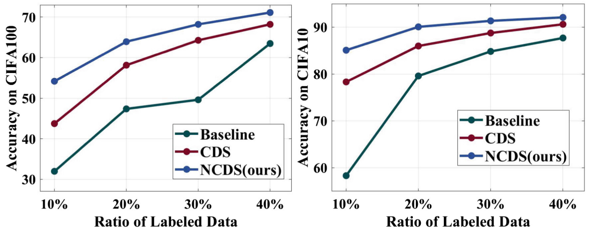


Fig. 2. Evaluation of semi-supervised training on ResNet18 for CIFAR100 and CIFAR10.

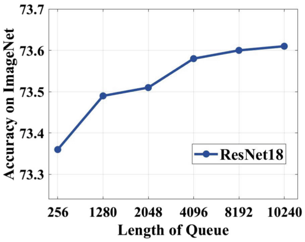
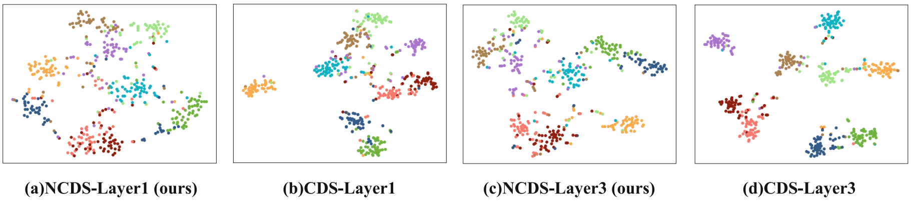


Fig. 3. Ablation study of length of queue on ResNet18 for ImageNet.



(a)NCDS-Layer1 (ours)

(b)CDS-Layer1

(c)NCDS-Layer3 (ours)

(d)CDS-Layer3

Fig. 4. t-SNE visualization of intermediate layer representations from different approaches on ResNet50 for the CIFAR100(results from the 100-epoch training out of the total of 300 epochs). For a more intuitive display of intra-class distances, we employ WCSS (Within-Cluster Sum of Squares) to quantify the intra-class distance of the representations, which measures the squared distance of all the points within a cluster to the cluster centroid. The WCSS values are 2938(a), 2409(b), 2670(c), and 2172(d), respectively. It is clear that our approach exhibits greater intra-class dispersion at shallow levels.

Semi-supervised Learning We conduct semi-supervised experiments with NCDS and CDS, using ResNet18 for CIFAR100 and CIFAR10 datasets. For each dataset, we evaluate with 10%, 20%, 30%, and 40% labels. In our approach, we train unlabeled and labeled data using $\mathcal{L}_{source}$ and $\mathcal{L}_{NCDS}$ respectively. For CDS [44], we follow their training approach, using $\mathcal{L}_{SimCLR}$ and $\mathcal{L}_{CDS}$ for unlabeled and labeled data, respectively. The experimental results are reported in Figure 2, which demonstrates that our approach outperforms CDS and Baseline at all labeled data ratios. Moreover, our approach has a greater advantage when there is less labeled data.

Ablation study of length of queue We investigate the impact of the length of queue on ImageNet with ResNet18 in Figure 3. It is observed that longer queue lengths can achieve better performance. The performance can be more affected by the queue length if it is less than the number of dataset classes. However, extremely long queues do not contribute to significant performance improvement, but they do significantly increase storage and computation requirements.

Study of t-SNE Visualization A comparative study is conducted on t-SNE visualization for NCDS and CDS projections of layer 1 and layers 3 on CIFAR100 with ResNet50, where NCDS and CDS supervise the intermediate layers with $\mathcal{L}_{source}+\mathcal{L}_{class}$ and $\mathcal{L}_{SupCon}$, respectively. Figure 4 shows the experimental results from the 100-epoch training out of a total of 300 epochs. It is observed that in the same layer, the projections of $\mathcal{L}_{NCDS}$ are more dispersed, while the projections of $\mathcal{L}_{SupCon}$ are much tighter. Compared to CDS with $\mathcal{L}_{SupCon}$, the intermediate representations of our approach have a larger intra-class distance, retaining more information unique to the individuals within the class. As pointed out in [19], increasing intra-class distance is advantageous for learning rich representations in transfer learning. Therefore, this suggests the potential for transferability and robustness in our approach.

5 Conclusion

We propose a generic training framework named NCDS that can obtain better visual representations. To learn more comprehensive knowledge, the contrastive module is designed to separate same-source loss and same-class losses. Besides, dynamic queues are designed in the intermediate layers to store same-class representations so that the model is no longer dependent on large batch sizes. Experiments with ten neural networks on mainstream datasets demonstrate that our NCDS can significantly improve the classification performance of these networks. Meanwhile, we also verify the effectiveness of our approach to semi-supervised learning. Furthermore, the larger intra-class distance of our intermediate features suggests that our approach also has the potential for transferability and robustness. In the future, we will further investigate the transferability of our approach and its potential applications in other fields.

Acknowledge. This work was supported in part by the National Key Research and Development Program of China under Grant (2023YFF1105102, 2023YFF1105105), the Major Project of the National Social Science Foundation of China (No. 21&ZD166), the National Natural Science Foundation of China (61876072), the Natural Science Foundation of Jiangsu Province (No. BK20221535) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (No. KYCX23_2438).

References

1. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Adv. Neural. Inf. Process. Syst. **33**, 9912–9924 (2020)
2. Chaitanya, K., Erdil, E., Karani, N., Konukoglu, E.: Contrastive learning of global and local features for medical image segmentation with limited annotations. Adv. Neural. Inf. Process. Syst. **33**, 12546–12558 (2020)
3. Chen, M., Fu, D.Y., Narayan, A., Zhang, M., Song, Z., Fatahalian, K., Ré, C.: Perfectly balanced: Improving transfer and robustness of supervised contrastive learning. In: International Conference on Machine Learning. pp. 3090–3122. PMLR (2022)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
5. Chen, T., Luo, C., Li, L.: Intriguing properties of contrastive losses. Adv. Neural. Inf. Process. Syst. **34**, 11834–11845 (2021)
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. Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., Feng, J.: Dual path networks. Advances in neural information processing systems **30** (2017)
8. 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)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Elsayed, G., Krishnan, D., Mobahi, H., Regan, K., Bengio, S.: Large margin deep networks for classification. Advances in neural information processing systems **31** (2018)
11. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Adv. Neural. Inf. Process. Syst. **33**, 21271–21284 (2020)
12. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
13. 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)
14. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. pp. 630–645. Springer (2016)

15. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
16. Huang, G., Chen, D., Li, T., Wu, F., Van Der Maaten, L., Weinberger, K.Q.: Multi-scale dense networks for resource efficient image classification. arXiv preprint arXiv:1703.09844 (2017)
17. 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)
18. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. pp. 646–661. Springer (2016)
19. Islam, A., Chen, C.F.R., Panda, R., Karlinsky, L., Radke, R., Feris, R.: A broad study on the transferability of visual representations with contrastive learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8845–8855 (2021)
20. Jeon, S., Min, D., Kim, S., Sohn, K.: Mining better samples for contrastive learning of temporal correspondence. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1034–1044 (2021)
21. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Adv. Neural. Inf. Process. Syst. **33**, 18661–18673 (2020)
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
23. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: Artificial intelligence and statistics. pp. 562–570. Pmlr (2015)
24. Li, C., Zeeshan Zia, M., Tran, Q.H., Yu, X., Hager, G.D., Chandraker, M.: Deep supervision with shape concepts for occlusion-aware 3d object parsing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5465–5474 (2017)
25. Li, D., Chen, Q.: Dynamic hierarchical mimicking towards consistent optimization objectives. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7642–7651 (2020)
26. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. arXiv preprint arXiv:1612.02295 (2016)
27. Reiß, S., Seibold, C., Freytag, A., Rodner, E., Stiefelhagen, R.: Every annotation counts: Multi-label deep supervision for medical image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9532–9542 (2021)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
29. Robinson, J., Sun, L., Yu, K., Batmanghelich, K., Jegelka, S., Sra, S.: Can contrastive learning avoid shortcut solutions? Adv. Neural. Inf. Process. Syst. **34**, 4974–4986 (2021)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
31. Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., Fergus, R.: Training convolutional networks with noisy labels. arXiv preprint arXiv:1406.2080 (2014)

32. Sun, D., Yao, A., Zhou, A., Zhao, H.: Deeply-supervised knowledge synergy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6997–7006 (2019)
33. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
34. Wang, L., Lee, C.Y., Tu, Z., Lazebnik, S.: Training deeper convolutional networks with deep supervision. arXiv preprint arXiv:1505.02496 (2015)
35. Wang, P., Han, K., Wei, X.S., Zhang, L., Wang, L.: Contrastive learning based hybrid networks for long-tailed image classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 943–952 (2021)
36. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3733–3742 (2018)
37. Xie, E., Ding, J., Wang, W., Zhan, X., Xu, H., Sun, P., Li, Z., Luo, P.: Detco: Unsupervised contrastive learning for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8392–8401 (2021)
38. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)
39. Xue, Y., Joshi, S., Gan, E., Chen, P.Y., Mirzasoleiman, B.: Which features are learnt by contrastive learning? on the role of simplicity bias in class collapse and feature suppression. arXiv preprint arXiv:2305.16536 (2023)
40. Ye, M., Zhang, X., Yuen, P.C., Chang, S.F.: Unsupervised embedding learning via invariant and spreading instance feature. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6210–6219 (2019)
41. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
42. Zeng, G., Yang, X., Li, J., Yu, L., Heng, P.A., Zheng, G.: 3d u-net with multi-level deep supervision: fully automatic segmentation of proximal femur in 3d mr images. In: Machine Learning in Medical Imaging: 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings 8. pp. 274–282. Springer (2017)
43. Zhang, H., Koh, J.Y., Baldridge, J., Lee, H., Yang, Y.: Cross-modal contrastive learning for text-to-image generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 833–842 (2021)
44. Zhang, L., Chen, X., Zhang, J., Dong, R., Ma, K.: Contrastive deep supervision. In: European Conference on Computer Vision. pp. 1–19. Springer (2022)
45. Zhang, L., Shi, Y., Shi, Z., Ma, K., Bao, C.: Task-oriented feature distillation. Adv. Neural. Inf. Process. Syst. **33**, 14759–14771 (2020)
46. Zhang, L., Tan, Z., Song, J., Chen, J., Bao, C., Ma, K.: Scan: A scalable neural networks framework towards compact and efficient models. Advances in Neural Information Processing Systems **32** (2019)
47. Zhang, Y., Chung, A.C.: Deep supervision with additional labels for retinal vessel segmentation task. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11. pp. 83–91. Springer (2018)
48. Zhang, Z., Zhang, X., Peng, C., Xue, X., Sun, J.: Exfuse: Enhancing feature fusion for semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 269–284 (2018)

49. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems **31** (2018)
50. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. arXiv preprint arXiv:1412.6856 (2014)

Multi-Task Interaction Network Based on a Cross-Attention Fusion Mechanism for Offline Signature Verification

Haotian Meng[1], Xiaoya Lin[1], Kurban Ubul[1,2,3](✉), and Alimjan Aysa[1,2](✉)

[1] School of Computer Science and Technology, Xinjiang University, Urumqi, China
{kurbanu,alim}@xju.edu.cn
[2] Xinjiang Multilingual Information Technology Key Laboratory, Xinjiang University, Urumqi, China
[3] Joint International Research Laboratory of Silk Road Multilingual Cognitive Computing, Urumqi, China

Abstract. Offline handwritten signature verification has been a challenging pattern recognition problem due to high intra-writer variability and inter-class similarity. In this paper, we propose a multi-task interaction network (MTI) based on a cross-attention fusion mechanism, which can dynamically focus on the critical detail differences between input signature pairs and efficiently capture contrast cues by fusing the contextual information between two images. Additionally, we designed a multi-task learning framework that enables the model to verify the authenticity of a signature while enhancing its understanding of the signature's writing style through recognition. By combining these two tasks, our network enhances overall verification accuracy by increasing the understanding of writing styles and distinguishing key detail differences between genuine and forged signatures through fine-grained comparison cues. Most current studies are based on signatures written in the same language script. Given the rich linguistic and cultural background in China's Xinjiang region, we construct a multilingual offline signature dataset containing Uyghur, Kirghiz, Kazakh, and Chinese, the first comprehensive dataset combining character-based and letter-based signatures. The verification accuracies of our method on the publicly available datasets CEDAR, BHSig-H, BHSig-B, and our Multilingual dataset Mult-Sig reach 100%, 91.19%, 94.12%, and 92.88%, respectively, and extensive experiments demonstrate the effectiveness of the proposed method and its competitiveness with current state-of-the-art techniques.

Keywords: Signature verification · Cross-attention fusion · Contrastive Interaction

1 Introduction

A handwritten signature is a personal identification method that combines textual and behavioral features with a long history and modern applications. It is

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 17–31, 2025.
https://doi.org/10.1007/978-3-031-78389-0_2

widely used in various fields such as administration, banking, contract signing, and credit card transactions. As society attaches great importance to security, the authentication security of handwritten signatures has become the focus. Handwritten signature recognition mainly consists of two tasks: 1. signature recognition(SR), i.e., to determine the identity attribution of the signature image; 2. signature verification(SV), i.e., to determine whether a pair of signature images belong to the same person. Handwritten signature verification technology is categorized online and offline based on how the signature data is captured. Online signature verification has high accuracy as dynamic features (e.g., pressure and speed) are captured directly by electronic devices. In contrast, offline signature verification is more challenging as signature images are captured by scanning or photographing, and only handwriting morphology information can be analyzed. Despite the higher accuracy of online verification, developing high-precision offline signature verification systems is of great value due to the ease of offline signature acquisition and its wide popularity in daily applications. Offline signature verification systems are categorized into Writer-Dependent and Writer-Independent strategies[11,15], the former constructing a proprietary-type model for each user and the latter permitting effective verification without a specific author by constructing a generic model. This study focuses on developing Writer-Independent offline handwritten signature verification models.

Currently, most methods focus on evaluating feature representation and similarity or distance metrics[8,11,15,19,23]. Metric learning is mainly utilized to improve the distribution of signature samples in the feature space. Such methods effectively combine the capabilities of feature learning and similarity judgment. However, metric learning usually focuses on the overall similarity or difference in optimizing the distance metric, ignoring the interaction information between a pair of signature images. This interaction information often contains vital clues to distinguish between authentic and fake signatures. For example, the traditional metric learning approach may find two signatures (A and B) very similar in overall structure and, therefore, determine that they are signatures of the same person. However, by analyzing the fine-grained interaction information, a particular letter stroke in A has a heavier start. In comparison, the corresponding stroke in B has a lighter start, or at the turn of a particular stroke, A exhibits a smooth curvature, while B appears stiff. Such detailed information may be overlooked in the overall similarity comparison, but it is essential for distinguishing genuine signatures from forged ones. In addition, most signature verification frameworks rely solely on comparing specific pairs of signatures, which makes the model mainly learn how to distinguish the difference between two signatures and fails to recognize the unique writing style of each signature, which limits the model's understanding of the writing style of signatures to some extent. In contrast, forensic verification not only compares the verified signature with the reference signature but also the signature's writing style when verifying the signature's authenticity. These unique styles are determined by the long-term writing habits of the writer, are highly personalized and extremely stable, and forgers cannot have this muscle memory in the short term. This includes the size of the letters, the slant, the way they are joined, and the overall smoothness or sharpness of the writing. Therefore, identifying the unique style of a signature is crucial to improving the generalization ability and verification accuracy of the model.

In this paper, we use a cross-attention fusion mechanism to integrate the interaction information between two images. The interaction vector is then compared with individual vectors. This approach allows the generation of distinct cues from the unique perspective of each image, which helps to distinguish the differences between signature pairs. Compared to simple distance metrics, this method generates more diverse and expressive comparison cues by comparing interaction features, effectively mitigating the problem of detailed information being overlooked in the overall similarity comparison. In addition, to address the problem that the model relies only on comparing specific signature pairs and lacks understanding of signature writing styles, we design a multi-task learning framework. The model not only learns how to verify the authenticity of signatures but also can capture and understand the personalized features of each signature by recognizing the identity, e.g., some small actions of some writers when starting and closing the pen will show some small hooks and picks in the handwriting. This enables the model to synthesize the unique writing styles of the signatures to understand and differentiate different individual signatures more comprehensively. Considering that a skilled forger may mimic the overall structure of a signature but has difficulty replicating every detailed feature, the extraction and utilization of fine-grained features are crucial for signature verification. To this end, we introduce the self-channel interaction[5] module, which focuses on exploiting the negative correlation of spatial locations to enable the model to find semantically complementary channel information, which enhances the channel features and generates more robust fine-grained feature representations. The main contributions of this paper are as follows:

1. We propose a contrastive interaction network based on a cross-attention fusion mechanism. This network can dynamically focus on the key detail differences between input signature pairs and efficiently capture the contrast cues by fusing the contextual information between two images.
2. We design a multi-task learning framework in which the model not only verifies the authenticity of signatures during training but also captures and understands the personalized features of each signature by recognizing the identity.
3. We introduce the self-channel interaction (SCI) module to enable the model to learn complementary features from related channels, thus improving the extraction of fine-grained features.
4. We constructed a multilingual offline signature dataset containing Uyghur, Kirghiz, Kazakh, and Chinese, which contains signature samples from 800 independent individuals, totaling 38,400 signature images.

2 Related Work

In the past decades, many studies have relied on manually designed features such as texture features like grayscale covariance matrix[18], local binary patterns[22], local directional pattern (LDP)[4], as well as compact correlated feature[3], surroundedness feature[7], SURF features[12], and KAZE features[13]. However, deep learning-based methods have demonstrated superior performance over traditional manual feature methods in signature verification tasks with the rise of deep learning techniques, especially the wide application of Convolutional Neural Networks (CNNs) in image processing.

Current mainstream signature verification methods are mainly based on CNN structure and metric learning; Dey et al. [2] proposed a Siamese Convolutional Network that employs contrast loss to determine whether an input pair of signatures belongs to the same person and optimizes the computed distance metric based on it. Li et al. [8] proposed a dual-channel CNN for offline signature verification, where two inputs are merged into a single dual-channel image as input to a single network; the network learns the similarity of the two inputs at the beginning and finally computes the similarity between signatures through two logarithms. Wan et al. [19] proposed an author-independent metric learning method for offline signature verification using double-triple loss. Zhu et al. [23] proposed a point-to-set (P2S) metric for offline signature verification that focuses mainly on optimizing the distance from a point to a set. However, metric learning usually focuses on the overall similarity or difference in optimizing the distance metric, ignoring the interaction information between a pair of signature images. Even if two signatures are regarded as dissimilar in the feature space, the model finds it challenging to specify the visual differences that lead to such judgments. The research field has started to explore new technical routes. Some transformer-based approaches emerged[9,16]; in addition, Soumitri et al. [1] introduced a self-supervised representation learning approach, and Ren et al. [15] proposed a graph convolutional network-based framework for handwritten signature verification. Li et al. [10] proposed a multimodal handwritten signature verification system combining static and dynamic features.

In this paper, We rethink the way of CNN structure and metric learning. Similar ideas to some of the works[2,11], we use the Siamese neural network and convolutional neural network to extract features. The differences are: 1. We utilize the interaction information between the reference and test signatures. The fusion of contextual information between the two images is achieved through the cross-attention mechanism, which is different from directly using the Euclidean distance or cosine similarity between two feature vectors to determine whether the signatures are similar, but uses the interaction information to generate adaptive contrast cues to distinguish the genuine signatures from the forged ones with more precision. 2. We design a multi-task learning framework. The model not only relies on comparing specific pairs of signatures to verify authenticity but also captures and understands the personalized features of each signature by identifying the identity. Combining the two tasks enhances the model's generalization ability and verification accuracy.

3 Method

Our model consists of two parts (shown in Fig. 1) corresponding to different tasks: the contrastive interaction module is used for signature verification (3.2). The self-channel interaction module is used for signature recognition (3.3), and we will describe in detail how each of the two parts works.

3.1 Network Architecture

Our backbone network consists of a four-layer structure; each layer comprises two convolutional layers (Conv), a batch normalization layer (BN), and a ReLU activation function. After each layer, a generalized-mean (GeM) pooling [14] is applied, which is crucial as widely used maximum or average pooling often loses critical information. We use a learnable pooling layer, which is more flexible in capturing sparse features, thus better preserving critical information in the signature image. The structure of our network is shown in Table 1.

Table 1. All dimensions are specified in the format of filters × height × width, with ReLU serving as the activation function.

Layer	Size	Kernel Parm
input	1×155×220	
conv1	64×3×3	stride = 1, pad = 1
conv1	64×3×3	stride = 1, pad = 1
gem pool1	adaptation	p=3 (trainable)
conv2	96×3×3	stride = 1, pad = 1
conv2	96×3×3	stride = 1, pad = 1
gem pool2	adaptation	p=3 (trainable)
conv3	128×3×3	stride = 1, pad = 1
conv3	128×3×3	stride = 1, pad = 1
gem pool3	adaptation	p=3 (trainable)
conv4	256×3×3	stride = 1, pad = 1
conv4	256×3×3	stride = 1, pad = 1
gem pool4	adaptation	p=3 (trainable)
gap	256	
fc	2	

3.2 Contrastive Interaction Module

First, we input reference and verification signature images into the backbone network and extract their feature maps, i.e., respectively f_1 and $\mathrm{f}_2 \in \mathrm{R}^{\mathrm{c}\times\mathrm{h}\times\mathrm{w}}$.

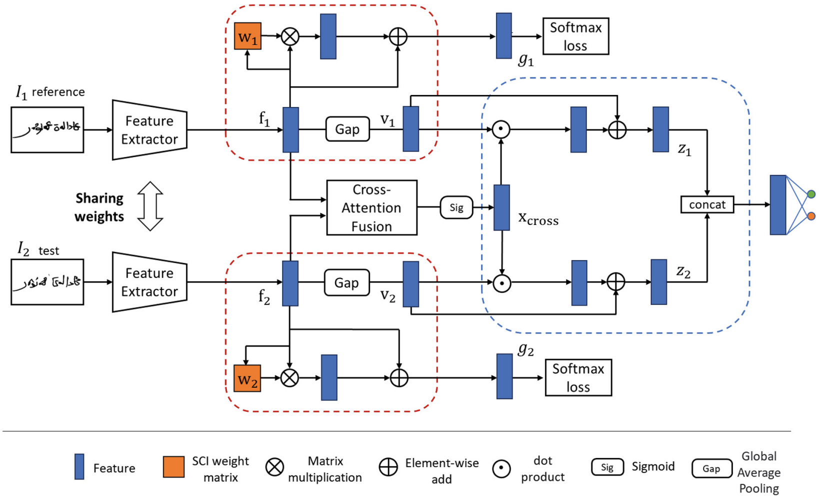


Fig. 1. Multi-Task Interaction Network. The red area indicates the self-channel interaction module for signature recognition, and the blue area indicates the contrastive interaction module for signature verification. The two tasks are performed simultaneously during training, and in the testing phase, the model only performs signature verification to determine the signature's authenticity. It should be noted that only genuine signatures are recognized; forged signatures are not involved in the recognition task. (Color figure online)

For each output feature map, we generate its query Q, key K, and value V, respectively, using a 1 × 1 convolutional layer. k, q, and v are then reshaped into a tensor $\in \mathrm{R}^{\mathrm{c} \times \mathrm{m}}, \mathrm{m} = \mathrm{w} \times \mathrm{h}$. As shown in Fig. 2, attention weights are determined by calculating the correlation between the query of the reference signature and the key of the test signature.

$$\gamma_{i,j} = \frac{\exp\left(s_{ij}\right)}{\sum_{j=1}^{m} \exp\left(s_{ij}\right)}, \text{ where } s_{ij} = k_i \otimes q_j^T \tag{1}$$

$\otimes$ denotes the matrix multiplication. s_{ij} computes the correlation between the i^{th} and j^{th} positions in k and q, respectively. Meanwhile, γ represents the attention graph, depicting the softmax normalization for each i row. This effectively allows the model to allocate more "attention" to the most critical information.

We weigh the value V to obtain a weighted feature representation using these weights.

$$r_i = \sum_{j=1}^{m} \gamma_{i,j} v_j, r_i \in \{r_1, r_2, \ldots, r_m\} \tag{2}$$

Then, r is reshaped into a tensor $\in \mathrm{R}^{c \times h \times w}$, and the obtained weighted representation of the feature map is superimposed with the original feature map to obtain the final feature representation. In addition, we swap the inputs and

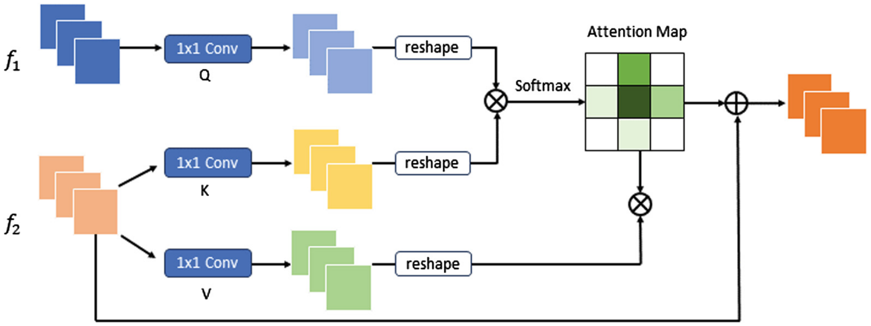


Fig. 2. Internal workings of the cross-attention module. Q, K, and V are 1×1 convolution functions.

then repeat the above computation. This step ensures that the features of Image 1 are incorporated into the contextual information of Image 2 while also allowing the features of Image 2 to contain the context of Image 1.

After completing these two rounds of computation, we join the resulting two sets of weighted features along the channel dimensions and project them through another 1x1 convolutional layer to obtain the final output feature map. Global average pooling is applied to the output feature map to obtain an interaction vector $x_{\text{cross}} \in \mathrm{R}^{\mathrm{c}}$, which contains information about the interaction between a pair of signatures and can capture high-level comparison cues. For example, a feature in one signature may differ in its corresponding position in another signature.

After obtaining the interaction vector, we propose comparing it with v_1 and v_2 (where v_1 and v_2 are derived from the feature maps f_1 and f_2 through global average pooling). The reason for this approach is that it can generate different clues from the unique perspective of each image, which helps distinguish between genuine and forged signatures.

We add a sigmoid function to normalize x_{cross} and then perform dot products with v_1 and v_2, respectively. The dot product operation measures the similarity between two vectors, thereby highlighting the channel features that are more important or different in comparison. Finally, we introduce a residual structure to enhance the original features using the discriminative clues from both images. The formula is as follows:

$$z_1 = v_1 \odot x_{\text{cross}} + v_1 \tag{3}$$

$$z_2 = v_2 \odot x_{\text{cross}} + v_2 \tag{4}$$

We concatenate the obtained attentive feature vectors z_1 and z_2 together and output a two-dimensional vector through a fully connected layer. The model is then optimized using a binary cross-entropy loss. The binary cross-entropy loss for the signature verification task we denote as L_{SV}.

$$L_{SV}(y, \hat{p}) = -[y \ln \hat{p} + (1-y)\ln(1-\hat{p})] \tag{5}$$

y=1 indicates that the signature pairs are from the same author, y=0 indicates that the signature pairs are from different authors, and p denotes the probability that the model predicts that two signatures are from the same author.

3.3 Self-Channel Interaction Module

Due to each individual's unique signature writing style, focusing only on the most discriminatory feature channels may not fully mine all available information. Inspired by Yu Gao et al.[5], we introduce the Self-Channel Interaction module since most feature channels are complementary to each other. We can extract these complementary cues by computing the relationships between these channels to better model signature writing styles.

For each signature image in the input, after CNN to get the feature mapping size c×h×w, the feature map size is reshaped as $X \in \mathrm{R}^{\mathrm{c}\times h*w}$, then the output of SCI is computed as:

$$Y = WX \in R^{c\times h*w} \tag{6}$$

Where $\mathrm{W} \in \mathrm{R}^{\mathrm{c}\times\mathrm{c}}$ denotes the weight matrix, W is calculated as follows: first, a bilinear operation is performed between X and X^T bilinear operation to obtain a bilinear matrix, representing the spatial relationship between the channels. Next, the negative sign is taken for this matrix, and the softmax function obtains the weight matrix:

$$W_{ij} = \frac{\exp\left(-XX_{ij}^T\right)}{\sum_{k=1}^{c}\exp\left(-XX_{ik}^T\right)} \tag{7}$$

Where $\sum_{\mathrm{k}=1}^{\mathrm{c}} \mathrm{W}_{\mathrm{ik}} = 1$. It is worth noting that the i^{th} channel of the output feature Y is obtained by calculating X_i interactions with all channels of X, i.e., $Y_i = W_{i1}X_1 + \ldots + W_{ic}X_c$.

According to the definition of the weight matrix W, channels with larger weights tend to complement channel semantically X_i. For example, suppose a channel focuses on capturing a signature's starting or ending strokes. In that case, channels that highlight the middle part of the signature are given a more significant weight to complement information that may be missing from the first and last parts. This mechanism focuses on exploring the totality of the signature by evaluating the complementarities and interactions between different parts.

We use the residual structure to aggregate the generated features with the original features.

$$G = \psi(Y) + X \tag{8}$$

where ψ denotes a 3×3 convolutional layer.

Ultimately, we employ a multi-class cross-entropy loss function for classification prediction of feature G generated based on Self-Channel Interaction. We denote the loss for the signature recognition task as L_{SV}. The total loss of our framework is defined as follows:

$$L = L_{SV} + L_{SR} \tag{9}$$

4 Experiments

Datasets We conducted our experiments on four datasets: CEDAR, BHSig-H, BHSig-B, and Mult-Sig.Table 2 presents the relevant information of the dataset used in this experiment, and Fig. 3 shows some examples.

CEDAR: This western offline signature dataset includes 55 users, each with 24 genuine and 24 forged signatures. Each user has 276 genuine-genuine pairs and 576 genuine-forged pairs. We randomly select five users for testing and the remaining 50 for training.

BHSig-B: This dataset contains signatures from 100 users, each with 24 genuine and 30 forged signatures. Each user has 276 genuine-genuine pairs and 720 genuine-forged pairs. We randomly select 50 users for training and 50 users for testing.

BHSig-H: This dataset includes 160 users with 24 genuine and 30 forged signatures. Each user has 276 genuine-genuine pairs and 720 genuine-forged pairs. We randomly select 100 users for training and 60 users for testing.

Mult-Sig: This multilingual dataset includes signatures in Uyghur, Kirghiz, Kazakh, and Chinese, with 200 users for each language. Each user contributes 24 genuine and 24 skillfully forged signatures, resulting in 38,400 images from 800 users. Each user has 276 genuine-genuine pairs and 576 genuine-forged pairs. We randomly select 600 users for training and 200 users for testing.

Table 2. Summary of basic information about the experimental datasets.

Dataset	Language	Composition	Number of Authors
CEDAR	English	Letters	55
BHSig-H	Hindi	Letters	160
BHSig-B	Bengali	Letters	100
Mult-Sig	Chinese, Uyghur, Kazakh, Kirghiz	Letters, Characters	800

Implementation Details We utilize bilinear interpolation to standardize the size of all images to a consistent dimension of 155×220 pixels. After resizing, we convert the images to grayscale. Moreover, the training dataset was not enhanced through data augmentation or any other methods. To keep the number of positive and negative samples balanced, we randomly selected from the negative pair of samples to balance the number of positive pairs of samples. We implemented our model based on the PyTorch platform. We used the Adam optimizer for

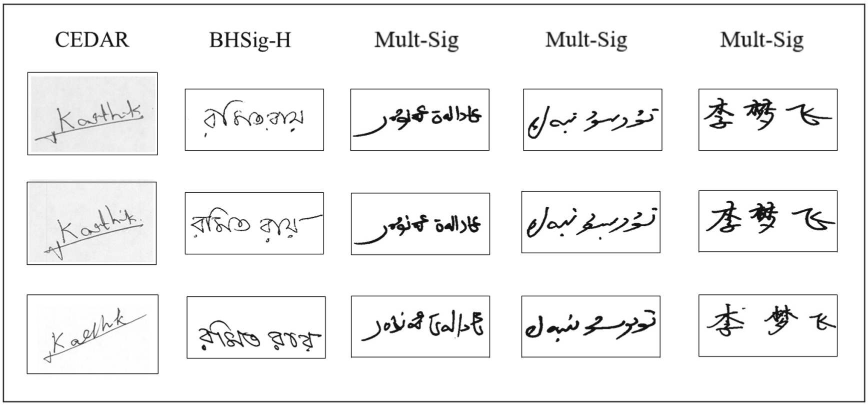


Fig. 3. Examples of CEDAR, BHSig-H, and Mult-Sig datasets. The first and second rows are genuine signatures; the third is forged signatures.

training, where the learning rate was initially set to 0.001 and tuned using a cosine annealing strategy. The training batch size of the model was set to 32. All training and testing was done on a single GTX 3090.

Evaluation metrics We quantify the performance of our method using widely recognized evaluation metrics commonly used in handwritten signature verification tasks. These include False Acceptance Rate (FAR), False Rejection Rate (FRR), Equal Error Rate (EER), and Accuracy (ACC). Where False Acceptance Rate (FAR) is the rate at which forged signatures are misdiagnosed as genuine signatures, False Rejection Rate (FRR) is the rate at which genuine signatures are misdiagnosed as forged signatures, and Equal Error Rate (EER) is the rate at which the False Acceptance Rate (FAR) and FRR are equal. The accuracy rate is the ratio of correctly predicted samples to all predicted samples.

4.1 Ablation Studies

Effects of Contrastive Interaction Module We evaluate different contrastive interaction mechanisms on the BHSig-H, BHSig-B, and Mult-Sig datasets. Detailed results are shown in Table 3. The experimental design includes a baseline algorithm without an interaction module, the proposed algorithm, and interaction vectors generated using various strategies. These strategies aim to extract mutual information from paired images, demonstrating the importance of interaction vectors in tasks such as image comparison, matching, and recognition. (I) Baseline algorithm: does not perform any interaction operation (II) Cross-attention fusion: the cross-attention fusion mechanism generates an interaction vector rich in mutual information by calculating the attentional weights of each image in an image pair concerning the other image. (III) Concatenate: The features of two images are directly connected to form a more extended vector.

(IV) Addition: Adding the features of two images through element-level addition operations, aiming at capturing the positive associations between images. (V) Subtraction: Subtracting one image's features from another using element-level subtraction operations to highlight the differences between the images. After operations (III), (IV), and (V), the resulting vectors are passed through two fully connected layers to learn the weight parameters.

Table 3. Analysis of the different interaction methods for MTI (EER%).

Interaction vector	BHSig-H	BHSig-B	Mult-Sig
Individual (Baseline)	15.78	11.19	12.57
Concat Operation	11.52	7.72	8.96
Sum Operation	13.12	8.26	9.17
Subtraction Operation	13.89	8.76	9.23
Cross-attention fusion Operation	8.81	5.88	7.12

Effects of Joint Multitask Training We explored the impact of multi-task learning strategies on the performance of the signature identification task in the BHSig-H, BHSig-B, and Mult-Sig datasets. According to the experimental results (see Table 4), training the signature recognition task in combination with the signature verification task resulted in a performance improvement compared to signature verification alone. In addition, the introduction of the self-channel interaction(SCI) module to the recognition task also observed a performance improvement. It is worth noting that there is a significant drop in performance when switching from the SCI module to the ECA (Effective Channel Attention)[20] module. This is because the ECA module while focusing on the most discriminative features, ignores other information that may be equally important. In contrast, the SCI module enhances the expressiveness of features more comprehensively by mining complementary information between channels.

Table 4. Analysis of joint multi-task training (EER%).

Model	BHSig-H	BHSig-B	Mult-Sig
SV	11.12	8.51	9.24
SV+SR	9.86	6.97	8.36
SV+SR(SCI)	8.81	5.88	7.12
SV+SR(ECA)	10.44	7.06	8.89

Table 5. Cross datasets validation results (ACC%).

Train\Test	CEDAR	BHSig-H	BHSig-B	Mult-Sig
CEDAR	100	49.05	58.12	51.12
BHSig-B	64.42	91.19	69.02	60.18
BHSig-H	58.85	66.21	94.12	62.06
Mult-Sig	70.92	75.23	76.79	92.88

4.2 Comparison with State of the Art

In our experiments, we compare our method with the following methods. As shown in Table 6, on the CEDAR dataset, we compared our method with methods such as SigNet [2], 2-Channel-2-Logit [8], MSN [21], and Surroundedness features [7], on which MTI showed excellent performance, achieving 100% accuracy, which is on par with methods such as SigNet and 2-Channel-2-Logit. This high performance may be partially attributed to the low complexity of the dataset itself. On two datasets, BHSig-H and BHSig-B, we compared our method with SigNet [2], 2C2S [16], MSN [21], 2-Channel-2-Logit [8], SURDS[1]. On the BHSig-H dataset, MTI achieves an accuracy of 91.19%, 0.51% higher than the next-best 2C2S method. On the BHSig-B dataset, MTI achieves 94.12% accuracy, 0.87% higher than the next-best 2C2S method. The results show that our proposed method outperforms all compared methods. Our method also performs more robustly in our large-scale multilingual integrated signature dataset. Meanwhile, our designed network is more suitable for sparse signature images than some classical backbone networks. The above proves the effectiveness of our method. In the future, our model can be further optimized to achieve better performance on more complex datasets. Meanwhile, our method can be generalized to similar tasks, such as image matching and fine-grained image recognition, to verify its generality and scalability.

4.3 Cross-Language Test

To analyze the generalization ability of the proposed method, cross-linguistic evaluation is performed on four independent datasets, as detailed in Table 5. When validated on cross-language datasets, a significant performance degradation is observed. This is because a model trained in a particular language will, inevitably, somewhat overfit the unique features of that language, making the model generalize poorly across languages. Our multilingual dataset achieves good cross-language performance when tested on the Bengali and Hindi datasets, which may be because Uyghur and Kirghiz are more linguistically similar to the Bengali and Hindi literature. There may be some commonalities in these languages' writing habits and styles, allowing the model to be better generalized.

Table 6. Comparison of the proposed method with the state-of-the-art methods on four signature databases (%).

Datasets	Methods	FAR	FRR	ACC	EER
CEDAR	SigNet [2]	**0**	**0**	**100**	–
	2-Channel-2-Logit [8]	–	–	**100**	**0**
	MSN [21]	3.18	0	98.40	1.63
	Surroundedness features [7]	8.33	8.33	91.67	–
	OURS	**0**	**0**	**100**	**0**
BHSig-H	SigNet [2]	15.36	15.36	84.64	15.36
	2-Channel-2-Logit [8]	–	–	86.66	13.34
	MSN [21]	17.06	**5.16**	88.88	11.31
	2C2S [16]	**8.66**	9.98	90.68	9.32
	SURDS [1]	12.01	8.98	89.50	–
	OURS	8.81	8.81	**91.19**	**8.81**
BHSig-B	SigNet [2]	13.89	13.89	86.11	13.89
	2-Channel-2-Logit [8]	–	–	88.08	11.92
	MSN [21]	10.42	6.44	91.56	8.43
	2C2S [16]	**5.37**	8.11	93.25	6.75
	SURDS [1]	19.89	**5.42**	87.34	–
	OURS	5.89	5.87	**94.12**	**5.88**
Mult-Sig	VGG16 [17]	8.46	8.46	91.54	8.46
	ResNet18 [6]	8.84	8.82	91.17	8.83
	OURS	**7.12**	**7.12**	**92.88**	**7.12**

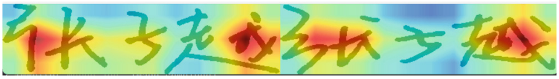


Fig. 4. Visualization results. Genuine signature on the left, skilled forgery on the right.

4.4 Visualization

We visualize the weights of the cross-attention modules, as depicted in Fig 4. The regions with warmer colors, which have higher attention weights, indicate that the model pays more attention to these areas. This highlights the visual distinctions between genuine and forged signatures, emphasizing the model's ability to discern crucial differences.

5 Conclusion

In this paper, we propose a multi-task interaction network based on a cross-attention fusion mechanism, which can dynamically focus on the crucial differences between input signature pairs and effectively capture contrast cues by integrating the contextual information from both images. Meanwhile, we design a multi-task learning framework that enables the model to verify the authenticity of a signature while recognizing it. The model can utilize the knowledge learned from the two tasks to improve the accuracy of the judgment. In the future, we can further optimize our model to achieve better performance on more complex datasets. Additionally, our approach can be extended to similar tasks, such as image matching and fine-grained image recognition, demonstrating its generality and scalability. In addition, we have collected a multilingual signature dataset covering Chinese, Uyghur, Kirghiz, and Kazakh languages in the Xinjiang region of China. To our knowledge, this is the first comprehensive dataset that integrates character-based and letter-based signatures in different linguistic contexts, laying a solid foundation for future research applicable to multilingual modeling in general-purpose scenarios.

Acknowledgements. This research was supported by the "Tianshan Talents" Science and Technology Innovation Leading Talents Project of Xinjiang Uygur Autonomous Region (2023TSYCLJ0025). In addition, it has also been supported by the National Natural Science Foundation of China (Project number: 62266044, 62061045) and the Key Laboratory of Oracle Information Processing of the Ministry of Education (OIP2021E004).

References

1. Chattopadhyay, S., Manna, S., Bhattacharya, S., Pal, U.: Surds: Self-supervised attention-guided reconstruction and dual triplet loss for writer independent offline signature verification. In: 2022 26th International Conference on Pattern Recognition (ICPR). pp. 1600–1606. IEEE (2022)
2. Dey, S., Dutta, A., Toledo, J.I., Ghosh, S.K., Lladós, J., Pal, U.: Signet: Convolutional siamese network for writer independent offline signature verification. arXiv preprint arXiv:1707.02131 (2017)
3. Dutta, A., Pal, U., Lladós, J.: Compact correlated features for writer independent signature verification. In: 2016 23rd international conference on pattern recognition (ICPR). pp. 3422–3427. IEEE (2016)
4. Ferrer, M.A., Vargas, F., Travieso, C.M., Alonso, J.B.: Signature verification using local directional pattern (ldp). In: 44th Annual 2010 IEEE International Carnahan Conference on Security Technology. pp. 336–340. IEEE (2010)
5. Gao, Y., Han, X., Wang, X., Huang, W., Scott, M.: Channel interaction networks for fine-grained image categorization. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 10818–10825 (2020)
6. 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)
7. Kumar, R., Sharma, J., Chanda, B.: Writer-independent off-line signature verification using surroundedness feature. Pattern Recogn. Lett. **33**(3), 301–308 (2012)
8. Li, C., Lin, F., Wang, Z., Yu, G., Yuan, L., Wang, H.: Deephsv: User-independent offline signature verification using two-channel cnn. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 166–171. IEEE (2019)

9. Li, H., Wei, P., Ma, Z., Li, C., Zheng, N.: Transosv: Offline signature verification with transformers. Pattern Recogn. **145**, 109882 (2024)
10. Li, Q., Wang, Z., Jin, L., Yadikar, N., Ubul, K.: Mmhsv: A multimodal handwritten signature verification fusing dynamic and static feature. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4730–4734. IEEE (2024)
11. Liu, L., Huang, L., Yin, F., Chen, Y.: Offline signature verification using a region based deep metric learning network. Pattern Recogn. **118**, 108009 (2021)
12. Malik, M.I., Liwicki, M., Dengel, A., Uchida, S., Frinken, V.: Automatic signature stability analysis and verification using local features. In: 2014 14th International Conference on Frontiers in Handwriting Recognition. pp. 621–626. IEEE (2014)
13. Okawa, M.: Kaze features via fisher vector encoding for offline signature verification. In: 2017 IEEE International Joint Conference on Biometrics (IJCB). pp. 10–15. IEEE (2017)
14. Radenović, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. IEEE Trans. Pattern Anal. Mach. Intell. **41**(7), 1655–1668 (2018)
15. Ren, C., Zhang, J., Wang, H., Shen, S.: Vision graph convolutional network for writer-independent offline signature verification. In: 2023 International Joint Conference on Neural Networks (IJCNN). pp. 1–7. IEEE (2023)
16. Ren, J.X., Xiong, Y.J., Zhan, H., Huang, B.: 2c2s: A two-channel and two-stream transformer based framework for offline signature verification. Eng. Appl. Artif. Intell. **118**, 105639 (2023)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
18. Vargas, J., Ferrer, M., Travieso, C., Alonso, J.B.: Off-line signature verification based on grey level information using texture features. Pattern Recogn. **44**(2), 375–385 (2011)
19. Wan, Q., Zou, Q.: Learning metric features for writer-independent signature verification using dual triplet loss. In: 2020 25th international conference on pattern recognition (ICPR). pp. 3853–3859. IEEE (2021)
20. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: Eca-net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11534–11542 (2020)
21. Xiong, Y.J., Cheng, S.Y.: Attention based multiple siamese network for offline signature verification. In: Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part III 16. pp. 337–349. Springer (2021)
22. Yilmaz, M.B., Yanikoglu, B., Tirkaz, C., Kholmatov, A.: Offline signature verification using classifier combination of hog and lbp features. In: 2011 international joint conference on Biometrics (IJCB). pp. 1–7. IEEE (2011)
23. Zhu, Y., Lai, S., Li, Z., Jin, L.: Point-to-set similarity based deep metric learning for offline signature verification. In: 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 282–287. IEEE (2020)

Functional Tensor Decompositions for Physics-Informed Neural Networks

Sai Karthikeya Vemuri[1,2(✉)], Tim Büchner[1], Julia Niebling[2], and Joachim Denzler[1]

[1] Computer Vision Group, Friedrich Schiller University Jena, 07743 Jena, Germany
sai.karthikeya.vemuri@uni-jena.de

[2] Institute of Data Science, German Aerospace Center, 07745 Jena, Germany

Abstract. Physics-Informed Neural Networks (PINNs) have shown continuous and increasing promise in approximating partial differential equations (PDEs), although they remain constrained by the curse of dimensionality. In this paper, we propose a generalized PINN version of the classical variable separable method. To do this, we first show that, using the universal approximation theorem, a multivariate function can be approximated by the outer product of neural networks, whose inputs are separated variables. We leverage tensor decomposition forms to separate the variables in a PINN setting. By employing Canonic Polyadic (CP), Tensor-Train (TT), and Tucker decomposition forms within the PINN framework, we create robust architectures for learning multivariate functions from separate neural networks connected by outer products. Our methodology significantly enhances the performance of PINNs, as evidenced by improved results on complex high-dimensional PDEs, including the 3d Helmholtz and 5d Poisson equations, among others. This research underscores the potential of tensor decomposition-based variably separated PINNs to surpass the state-of-the-art, offering a compelling solution to the dimensionality challenge in PDE approximation.

Keywords: Tensor Decomposition · Physics-Informed Neural Networks

1 Introduction

Employing existing physical knowledge within data-driven systems is important for scientists at the intersection of science, engineering, and machine/deep learning to enforce proper behavior within a model. Physics-informed neural networks (PINNs) [27] provide the paradigm to formulate rules of physics within the network architecture, such that the model learns from data and the underlying physics. Consequently, PINNs gained traction within the scientific machine-learning community. Mainly, PINNs are extensively applied to solve forward and inverse problems involving systems of differential equations. They are applied in various areas of science ranging from Geophysics, Structural mechanics, and

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 32–46, 2025.
https://doi.org/10.1007/978-3-031-78389-0_3

Fluid dynamics to Epidemiology [1,7,31,32]. We refer the readers to the excellent overview of PINNs by Cuomo et al. [7].

Like other numerical methods for solving partial differential equations (PDE), PINNs also suffer from the curse of dimensionality. We need n collocation points to solve a PDE accurately. In that case, the solution space explodes as n^d with each dimension d, and classical PINNs get into computational problems as neural networks struggle to resolve relevant features, often leading to erroneous solutions. Such problems and failure modes are discussed in detail in [17,22,34,35].

A classic method of solving differential equations is the variable separable method [26], where the solution is defined as the product of univariate functions. This approach is limited to a few classes of differential equations. We propose a new PINN alternative to the classical variable separable method, which can solve many arbitrary multi-dimensional PDEs irrespective of the presence of a separable form. We leverage tensor decomposition approaches and their functional forms and use individual neural networks to learn information along each dimension.

There is an increasing interest in using multiple neural networks and their blending together to form more accurate and robust PINNs. Moseley et al. [24] suggested dividing the domain of PDE solution into smaller subdomains and using individual neural networks to learn the solution within each subdomain. Haghighat et al. [10] propose an individual neural network for each variable in a mathematical model of solid mechanics consisting of five variables. Cai et al. [3], by applying PINNs to the two-phase Stefan problem, use two neural networks to model the unknown interface between two different material phases and describe the phases' two temperature distributions [18,19]. Jin et al. [14] used multiple neural networks to improve neural operators, which also come under the broad family of physics-informed machine learning, where a PDE solution operator is learned instead of solving a particular PDE. They used multiple branch and trunk nets and combined them to train Deep Neural Operators efficiently. Applying this to PINNs, Cho et al. [5] introduced Separable PINNs, where they use separate neural networks per axis, thus reducing the number of collocation points, and they leverage forward mode automatic differentiation to decrease computation and memory costs significantly.

Building upon these approaches, we introduce functional tensor decompositions as a generalized separation of variables method. We leverage several tensor decomposition forms to separate the variables in a PINN setting and approximate each decomposition component using a neural network. We show that these methods are more accurate and faster than state-of-the-art PINN architectures. Our four main contributions are as follows:

1. We extend the classical variable separable methods with PINNs by leveraging functional tensor decomposition forms, where individual neural networks learn each component of the tensor decomposition.
2. We extend the universal approximation theorem and show that any multivariate function can be approximated using the outer product of univariate neural networks, irrespective of whether a variable separable form exists.

3. We propose to use three functional tensor decomposition forms that can be combined into PINNs: Canonic-Polyadic (CP-PINN)[12], Tensor-Train (TT-PINN)[25], and Tucker decomposition[30].
4. We demonstrate that our proposed method outperforms previous state-of-the-art PINN architectures for high-dimensional PDEs and requires fewer collocation points [5]. Thus, it offers an effective means to mitigate the curse of dimensionality and provide a better representation of solutions. The code is available at https://github.com/cvjena/TensorDecompositions4PINNs

2 Theoretical Background

The classical variable separable method involves writing a multivariate function as a sum of products of univariate functions. Such forms only exist for a limited number of functions/PDEs [26]. When we decompose a multivariate function and learn each univariate function using a neural network, it is crucial to demonstrate that this approach is practical even when a separable form does not exist. This ensures the method's general applicability to all types of PDEs. To support this argument, we show in the following sections that a multi-dimensional function can be approximated using the outer products of neural networks, where a neural network represents each dimension with a sufficient rank. After that, we explain in detail how tensor decompositions are utilized in PINNs and how such architectures could mitigate the curse of dimensionality and improve the speed and accuracy of PINNs.

2.1 Universal Approximation Theorem

In this section, we revisit the classic Universal Approximation Theorem [8,13] and extend it to separable functions. We show empirically that any continuous multivariate function $f \colon \mathbb{K} \to \mathbb{R}$ within a compact bounded d-dimensional set $\mathbb{K} \in R^d$ can be approximated by the outer product of d neural networks. Each neural network is a function of a single variable $x_i (1 \leq i \leq d)$. Furthermore, we demonstrate how tensor decomposition forms can separate dimensions and how these components support solving PDEs using PINNs.

The Universal Approximation Theorem states that a feed-forward neural network with a single hidden layer can theoretically approximate any continuous function on a bounded domain with arbitrary accuracy [8,13]. Hence, given a continuous function $f : \mathbb{R}^d \to \mathbb{R}$ and for any $\epsilon > 0$, there exists a feed-forward neural network $\widehat{f}$, such that

$$\left\| f(x) - \widehat{f}(x) \right\| < \epsilon; \forall x \in \mathbb{K}. \tag{1}$$

considering that for all x inside of $\mathbb{K}$, which is a compact subset in $\mathbb{R}^d$, the inequality holds true.

Functional approximation problems, including those tackled by PINNs, involve high-dimensional spaces where the *curse of dimensionality* becomes a significant issue. Thus, significant challenges are posed in computation, the number of parameters needed, and the calculation of derivatives (an additional special case for PINNs).

A classical approach to address this *curse of dimensionality* is the sum of separable functions and is based on reconstructing a multivariate function as a product of univariate functions [5,11,26]: A d-variate function $f : \mathbb{K} \to \mathbb{R}$ can be written as

$$f(x_1, x_2, \ldots, x_d) = \sum_{j=1}^{r} \bigotimes_{i=1}^{d} g_i^j(x_i). \tag{2}$$

Where g_i^j are univariate functions, j denotes the separation rank, representing the number of terms in the function g_i. We designate the operator $\bigotimes$ as the tensor product of vector spaces defined by individual univariate functions. One of the important features of these functions is that they are not restricted from coming from a particular basis set. This is the exact point we would like to emphasize. Since they are not restricted to being unique, we propose that neural networks can approximate these functions.

We simplify the notation by rewriting the Equation (2) to omit separation rank, and the separated functions g_i contain r components corresponding to the i-th dimension. This is written as:

$$f(x_1, x_2 ...., x_d) = \bigotimes_{i=1}^{d} g_i(x_i). \tag{3}$$

Now, we use d neural networks, e.g., per dimension of the problem, to approximate the individual functions g_i. Now the tensor product of these univariate approximations is the approximation of f, denoted by $\widehat{f}$, as

$$\widehat{f}(x_1, x_2 ...., x_d) = \bigotimes_{i=1}^{d} \widehat{g}_i(x_i, \theta_i). \tag{4}$$

Where θ_i represents a neural network's trainable parameters (weights and biases). Following the above-mentioned *Universal Approximation Theorem*, we can approximate a neural network $\widehat{g}_i$ for every individual component g_i [8,13] and every $\epsilon_i \in \mathbb{R}$, such that

$$\|g_i(x_i) - \widehat{g}_i(x_i)\| < \epsilon_i; \forall x_i \in \mathbb{K}_i \wedge 1 \leq i \leq d, \tag{5}$$

where $\mathbb{K}_i$ is a compact subset of $\mathbb{R}$. The error in approximating f by $\widehat{f}$ can be now written as

$$\left\| f(x_1, x_2, \ldots, x_d) - \hat{f}(x_1, x_2, \ldots, x_d) \right\| = \left\| \bigotimes_{i=1}^{d} g_i(x_i) - \bigotimes_{i=1}^{d} \widehat{g}_i(x_i, \theta_i) \right\|. \tag{6}$$

Under certain reasonable assumptions that the univariate functional spaces are Banach in nature, and the norm is a reasonable cross-norm [11], the norm of the outer product can be written as simply the norm of products. This property is illustrated as follows:

$$\| \bigotimes_{i=1}^{d} x_i \| = \prod_{i=1}^{d} \|x_i\|. \tag{7}$$

Using this property and identity of difference of products, which is derived using mathematical induction in [11], we expand the right-hand side of Equation (6):

$$\bigotimes_{i=1}^{d} g_i - \bigotimes_{i=1}^{d} \widehat{g}_i = \sum_{j=1}^{d} \left(\prod_{k=1}^{j-1} \widehat{g}_k \right) (g_j - \widehat{g}_j) \left(\prod_{l=j+1}^{d} g_l \right). \tag{8}$$

Taking the norm and including the above equation in Equation (6)

$$\|f - \widehat{f}\| = \left\| \sum_{j=1}^{d} \left(\prod_{k=1}^{j-1} \widehat{g}_k \right) (g_j - \widehat{g}_j) \left(\prod_{l=j+1}^{d} g_l \right) \right\|. \tag{9}$$

Using the triangle inequality [28], we obtain

$$\|f - \widehat{f}\| \leq \sum_{j=1}^{d} \left\| \left(\prod_{k=1}^{j-1} \widehat{g}_k \right) (g_j - \widehat{g}_j) \left(\prod_{l=j+1}^{d} g_l \right) \right\|. \tag{10}$$

Finally, assuming the norm is sub-multiplicative, something like L, we get

$$\|f - \widehat{f}\| \leq \sum_{j=1}^{d} \left(\prod_{k=1}^{j-1} \|\widehat{g}_k\| \right) \|g_j - \widehat{g}_j\| \left(\prod_{l=j+1}^{d} \|g_l\| \right). \tag{11}$$

Simplified expression by using Equation 5

$$\|f - \widehat{f}\| \leq \sum_{j=1}^{d} \left(\prod_{k=1}^{j-1} \|\widehat{g}_k\| \right) \epsilon_j \left(\prod_{l=j+1}^{d} \|g_l\| \right). \tag{12}$$

By appropriately choosing the approximation errors ϵ_j for each univariate function and making sure that the norms don't explode (i.e., weights and gradients do not explode)[9,15], we can ensure that the total error is less than any desired ϵ. Thus, the constructed multivariate approximation using outer products of univariate functions with large enough rank is also a universal function approximator. This shows that, theoretically, we can represent any arbitrary multivariate function, regardless of the existence of variable separable form, as the outer product of neural networks. The inputs to these individual neural networks correspond to particular dimensions. The underlying ideas of triangle inequality and identity of differences are drawn from well-established theories in the field [5,11,14].

2.2 Physics-Informed Neural Networks

Physics-Informed Neural Networks (PINNs) are a class of neural networks that incorporate physical laws described by partial differential equations (PDEs) into the training process [27]. Unlike traditional neural networks that rely exclusively on data-driven learning, PINNs utilize the underlying model structure, e.g., the actual gradients in the loss function, to constrain the solution space effectively. Thereby, prior knowledge and physically consistent constraints are enforced into the learning process. Specifically, in a PINN, the loss function $\mathcal{L}$ is augmented with terms that enforce the PDE constraints. Consider a PDE of the form:

$$\mathcal{F}(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x}), \nabla^2 u(\mathbf{x}), \dots) = 0, \tag{13}$$

where $x \in \mathbb{R}^d$ represents the spatial and temporal variables, and $u(x)$ describes the solution. The loss function $\mathcal{L}$ is composed of the data loss $\mathcal{L}_{\text{data}}$ and physics loss $\mathcal{L}_{\text{physics}}$. $\mathcal{L}_{\text{data}}$ captures the error between the predicted solution $u(x;\theta)$ and the observed data. Furthermore, $\mathcal{L}_{\text{physics}}$ constrains the solution space such that the model abides by the underlying governing physics at collocation points within the domain. We formally define both as

$$\mathcal{L}_{\text{data}} = \frac{1}{N}\sum_{i=1}^{N}\left(u(\mathbf{x}_i;\theta) - u_i^{\text{data}}\right)^2, \text{ and} \tag{14}$$

$$\mathcal{L}_{\text{physics}} = \frac{1}{M}\sum_{j=1}^{M}\left(\mathcal{F}(\mathbf{x}_j, u(\mathbf{x}_j;\theta), \nabla u(\mathbf{x}_j;\theta), \nabla^2 u(\mathbf{x}_j;\theta), \dots)\right)^2. \tag{15}$$

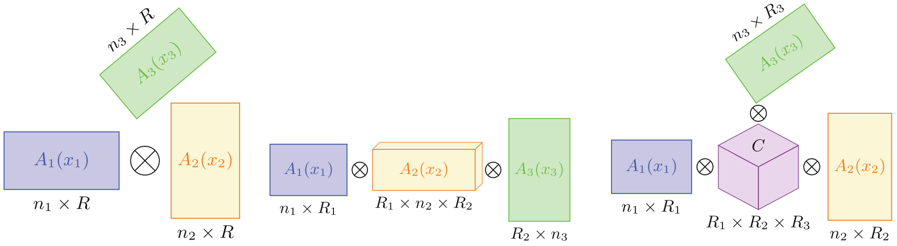
Where N and M are data and collocation points, respectively. The interplay between these two loss functions is controlled by parameter λ [23,32,34]. The combined multi-objective loss function is given as:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda \cdot \mathcal{L}_{\text{physics}}. \tag{16}$$

As stated, the collocation points refer to points inside the domain where the physics is obeyed. Generally, these are sampled uniformly in the domain to ensure the neural network learns the domain space. The curse of dimensionality manifests in PINNs as the number of collocations grows exponentially for every additional dimension. This challenges PINNs on many fronts, making them computationally expensive, and approximating solutions becomes increasingly difficult. Such failure modes are more explained in the works like [17,22,32,34,35].

PINNs can be easily seen as a special case of functional approximation, where, along with some samples, we give the underlying PDE residual(physics) from which the underlying function (solution of PDE) u needs to be approximated. We propose to represent the solution of a PINN as the outer product of univariate neural networks to separate variables. This can be seen as the PINN counterpart of the classic variable separable method. Since we have shown that the outer product of neural networks with a sufficient rank can approximate a multivariate function, this works even for cases where classical variable separable form does not exist, making a generalized separation of the variable method. The advantages of this approach over classical PINNs are as follows:

1. Requirement of fewer collocation points. While a classical PINN requires n^d collocation points to sample a d-dimensional domain, our approach needs only $n \cdot d$ points, effectively mitigating the curse of dimensionality.
2. The solution of the PDE is expressed in variable separable form, irrespective of classical variable separable form.
3. We employ individual neural networks per dimension, allowing better feature representation and avoiding potential local minima in complex problems.



(a) Canonic-Polyadic [12] (b) Tensor Train [25] (c) Tucker [30]

Fig. 1. We provide a schematic visualization for the tensor decompositions (a)-(c) on the examples for $d = 3$. The shape of the factor tensors (A) is written on the bottom of each component. Tucker [30] additionally has one core tensor C.

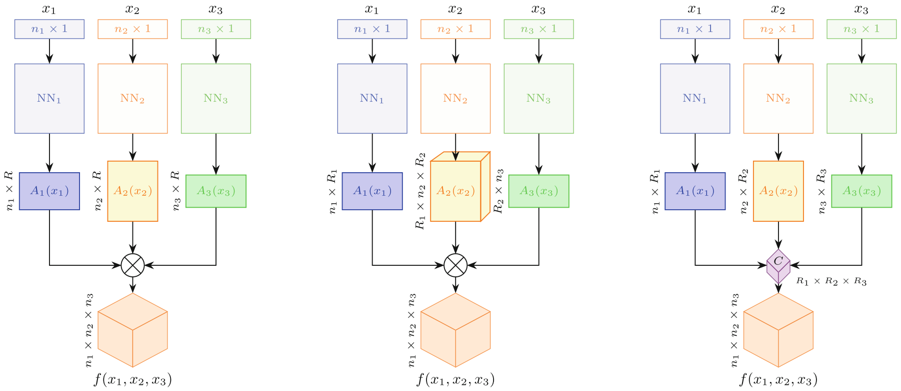
2.3 Functional Tensor Decompositions for PINNs

We leverage tensor decomposition forms to achieve the separation of variables. This approach decomposes a high-dimensional tensor into smaller components by approximating multivariate functions using outer products of univariate functions. We refer to this technique as *functional tensor decomposition*. A separate neural network is responsible for learning each component in the tensor decomposition. This work discusses three tensor decomposition forms: Canonic-Polyadic [12], Tensor-Train [25], and Tucker decompositions [30]. We provide definitions and schematics, and later the inclusion into the PINN architectures, but recommend the work of [16] for a broad overview of tensor decomposition.

Canonic-Polyadic Decomposition (CP) involves decomposing a d order tensor into d factor matrices of a specified rank R [12] similar to Separable PINN [5] as shown in Figure 1a. Mathematically, for a multi-dimensional tensor f, the CP decomposition is written as

$$f(x_1, x_2, \ldots, x_d) \approx [[A_1(x_1), A_2(x_2), \ldots, A_d(x_d)]], \tag{17}$$

where $[[\cdot]]$ denotes tensor product operation with $A_1 \in \mathbb{R}^{n_1 \times R}, \ldots, A_d \in \mathbb{R}^{n_d \times R}$ being factor matrices, for n_i points along each i-th dimension.



(a) CP-PINN (b) TT-PINN (c) Tucker-PINN

Fig. 2. Functional tensor decomposition forms within the PINN model architecture: The approximation of each component matrix based on a single variable is done with an individual neural network. These outputs are then combined as in the Canonic-Polyadic [12] (a), Tensor-Train [25] (b) or Tucker [30] (c) manner.

Tensor-Train Decomposition (TT) represents a high-dimensional tensor as a sequence of low-dimensional tensors (cores) connected in a chain (train) [25], with an example shown in Figure 1b. Similar to the matrix notation CP, we have

$$f(x_1, x_2, \ldots, x_d) \approx [[A_1(x_1), A_2(x_2), \ldots, A_d(x_d)]] \tag{18}$$

where each core tensor $A_i \in \mathbb{R}^{R_{i-1} \times n_i \times R_i}$, and $R_0 = R_d = 1$, where n_i is number of the points in the i-th dimension. Unlike CP, TT connects tensors belonging to the adjacent dimensions, making it more stable[25].

Tucker Decomposition generalizes CP by decomposing a tensor into a core tensor multiplied by matrices along each mode [30], visualized in Figure 1c. Therefore, by updating the CP matrix notation, we obtain

$$f(x_1, x_2, \ldots, x_d) \approx [[\mathcal{C}; A_1(x_1), A_2(x_2), \ldots, A_d(x_d)]] \ . \tag{19}$$

We denote $\mathcal{C} \in \mathbb{R}^{R_1 \times R_2 \times R_3 \times \ldots \times R_d}$ as the core tensor and $A_1 \in \mathbb{R}^{n_1 \times R_1}, A_2 \in \mathbb{R}^{n_2 \times R_2}, \ldots, A_d \in \mathbb{R}^{n_d \times R_d}$ are factor matrices. Unlike both CP and TT decomposition, the core connects all the dimensions, making it even more stable, with more parameters [30].

Functional tensor decomposition forms in PINNs We now use the aforementioned tensor decomposition in a PINN setup. As described earlier, we assume the solution of a PDE that needs to be solved by a PINN is decomposed into components constituting any of the tensor decomposition mentioned above, and a neural network learns each component. Therefore, we propose three architectures based on functional tensor decompositions: CP-PINN, TT-PINN, and Tucker-PINN. The schematics given in Figure 1 and 2 are for a three-dimensional PDE, but the concepts scales to arbitrary dimensions (≥ 3). For CP-PINN and Tucker-PINN, each network outputs a factor matrix of shape $n \times R$, where n still denotes the input dimension and R the desired rank of the decomposition. For this paper's scope, we consider that the ranks for all components of Tucker-PINN and TT-PINN are set to the same integer R. Additionally, for Tucker-PINN, we initialize the core tensor as an orthogonal and trainable parameter to learn the entries during training. For TT-PINN, each network outputs either the train start or end part with the shape $n \times r$ or the train middle of shape $r \times n \times r$.

3 Experiments

We solve benchmark PDEs in three dimensions and more to demonstrate the effectiveness of our tensor decomposition architectures CP-PINN, TT-PINN, and Tucker-PINN. We compare our results with the original PINN architecture [27] and other state-of-the-art PINN models [20,23,33]. Each variable is put into a four-layer feed-forward neural network with $\tanh(\cdot)$ activation functions for the proposed PINN architecture. The feature depth per layer corresponds to the rank unless specified otherwise. A network's input is collocation points along a single dimension of shape $n \times 1$, with n being the number of available points.

Our *functional tensor decomposition* models, the PDE simulation code, and experiments are created in JAX [2]. We adopt the implementation of forward gradients from [5]. The overall setup adheres to the conventional PINN framework, comprising a composite loss function (Equation 16) that combines data and PDE residual terms with no weighting, i.e., by setting $\lambda = 1$. All models are trained using Adam optimizer [15] with learning rate $1e^{-3}$ and for 50000 iterations. The performance metric is the L^2 error between predicted and simulated solutions. The tests are conducted on an NVIDIA GeForce GTX 1080 GPU, with reported relative speeds in iterations per second (IT/s).

First, we choose two three-dimensional PDE benchmarks, (2+1)d Klein-Gordon and 3d Helmholtz equation [5,21,29,37], for investigating the performance of our *functional tensor decomposition* based PINNs. Problems of this high dimension are computationally challenging for PINNs yet frequently arise in real-world applications. The boundary/initial conditions and visualizations are reported in Table 1 upper half. We compare our models against state-of-the-art methods like gradient-based PINN [27], G-PINN [36], SA-PINN [23], and Causal-PINN [33] (all implemented via PINA [6]). We omit SPINN [5] due to the same nature as CP-PINN. We ensured the solution was converged for all the benchmarks, and the training setup was as close as given in the original source. We experiment with multiple collocation points and ranks to evaluate the influence of these hyperparameters on the general model architecture.

The performance of the proposed approaches in benchmarking experiments is presented in Table 2. The results indicate that CP-PINN, TT-PINN, and Tucker-PINN demonstrate computational efficiency, requiring fewer collocation points

Table 1. We provide an overview of the four PDE experiment setups that were used to investigate the capabilities of our tensor decomposition PINNs. The plots are best viewed digitally and in color.

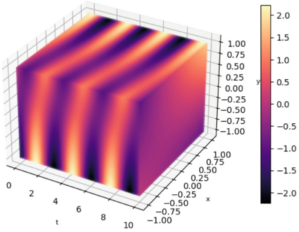
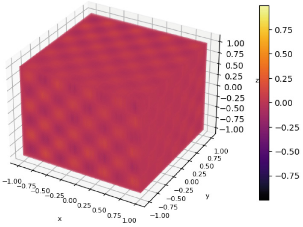
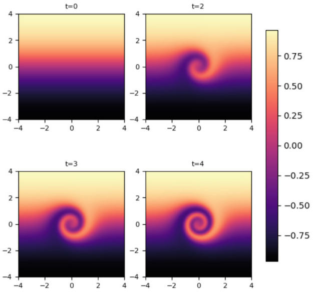
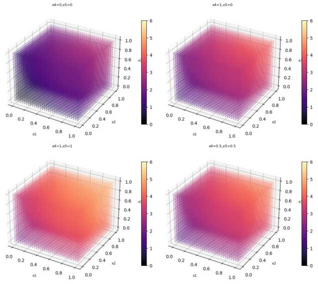
(2+1)d Klein-Gordon [5]	3d Helmholtz [29]
$f = \partial_{tt}u - \Delta u + u^2,$ $u = (x+y)\cdot\cos(2t),$ $u(x,y,0) = x + y + x\cdot y\cdot\sin(2t),$ $[x,y]\in[-1,1]^2, t\in[0,10]$	$\Delta u + k^2 u = q, x\in[-1,1]^3,$ $u(x) = 0, x\in\partial\Omega,$ $q = -(a_1\pi)^2 u - (a_2\pi)^2 u - (a_3\pi)^2 u + k^2 u,$ $u = \prod_{i=1}^{3}\sin(a_i\pi x_i)$
	
(2+1)d Flow mixing [4]	5d Poisson [37]
$0 = \frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y},$ $a(x,y) = -\frac{v_t\cdot y}{v_{\mathrm{t,max}}\cdot r},$ $b(x,y) = \frac{v_t\cdot x}{v_{\mathrm{t,max}}\cdot r},$ $v_t = \frac{\tanh(r)}{\tanh^2(r)},$ $r = \sqrt{x^2+y^2},$ $u = -\tanh(\frac{y}{2}\cos(\frac{v_t}{r\cdot v_{t,\max}}t) - \frac{x}{2}\sin(\frac{v_t}{r\cdot v_{t,\max}}t))$ with $t\in[0,4], x\in[-4,4], y\in[-4,4],$ and $v_{\mathrm{t,max}} = 0.385$	$\Delta u = -\frac{\pi^2}{4}\sum_{i=1}^{n}\sin(\frac{\pi}{2}x_i),$ $u = \sum_{i=1}^{n}\left(\sin\left(\frac{\pi}{2}x_i\right)\right),$ with $x\in[0,1], n = 5.$
	

Table 2. (2+1)d Klein Gordon Equation and 3d Helmholtz are solved by various architectures. The general trend is that a larger rank leads and a larger number of points leads to a better solution. Furthermore, we observe that our tensor decomposition PINNs are a magnitude of 10 better at solving these problems. The best performance per task is double underlined, and the second best is underlined.

PINN model	Rank	Points	$\mathcal{L}^2$ Klein-Gordon ↓	$\mathcal{L}^2$ Helmholtz ↓	Speed (IT/s) ↑
Vanilla PINN[27]	-	32^3	0.092	0.998	20
G-PINN[19]	-	32^3	0.073	0.794	16
SA-PINN[23]	-	32^3	0.095	0.920	15
Causal-PINN[33]	-	32^3	0.041	0.406	3
CP-PINN	8	64×3	0.050	0.061	364
	16	64×3	0.025	0.051	347
	32	16×3	0.069	0.063	357
	32	32×3	0.055	0.060	343
	32	64×3	$\underline{\underline{0.008}}$	$\underline{\underline{0.040}}$	327
TT-PINN	8	64×3	0.068	0.064	358
	16	64×3	0.043	0.061	353
	32	16×3	0.088	0.060	356
	32	32×3	0.079	0.055	310
	32	64×3	$\underline{0.010}$	$\underline{0.048}$	305
Tucker-PINN	8	64×3	0.061	0.079	345
	16	64×3	0.053	0.076	328
	32	16×3	0.066	0.077	301
	32	32×3	0.062	0.070	333
	32	64×3	0.019	0.057	312

and achieving higher accuracy (a factor of 10) compared to current state-of-the-art methods. A comparative analysis of CP-PINN, TT-PINN, and Tucker-PINN reveals that increasing the number of collocation points and the model's rank generally leads to improved solution accuracy. In particular, for both PDE problems, we observe that CP-PINN with a rank of 32 and 64×3 collocation points achieves the most accurate solutions. Similarly, TT-PINN achieves the second-best performance with the same rank and collocation points. We hypothesize that a low rank may result in excessive information compression, potentially leading to losing essential details in the subsequent inverse decomposition.

To further substantiate our findings, we investigate whether our *function tensor decomposition* scales to a higher dimension (> 3) effectively, as shown theoretically. Therefore, we solve 5d Poisson's Equation [37] and simulate the (2+1)d flow mixing PDE [4] to capture the intricate mixing process of two fluids, see Table 1 lower half. The other PINN architectures are not suitable to solve this task owing to their vast sampling of collocation points and slow speed

Table 3. We show the $\mathcal{L}_2$ loss for the CP-PINN, TT-PINN, and Tucker-PINN on the 5d Poisson's Equation and (2+1)d Flow mixing simulation data. The best performing scores are underlined, and each experiment has been repeated ten times (we omit the standard deviation as it has been consistent around *0.01.*)

Model	Points	Rank	CP-PINN	TT-PINN	Tucker-PINN
5d Poisson's Equation	24 × 3	6	0.097	0.048	0.040
		8	0.077	0.047	0.038
		12	0.033	0.037	0.026
(2+1)d Flow mixing	128 × 3	64	0.013	0.018	0.028

and are omitted from the results in Table 3. For the experiments, we use a fixed collocation point amount but vary the rank of the decomposition components. Surprisingly, our experiments reveal that Tucker-PINN outperforms both CP-PINN and TT-PINN, contradicting the findings in Table 2. We attribute this discrepancy to the fact that CP decomposition has fewer parameters, which may make it more challenging to find suitable rank-one approximations as dimensionality increases. Further, we observe that the (2+1)d flow mixing problem was solved only with a rank of 64 by our proposed architectures. This suggests that more collocation points may be necessary for an accurate approximation. Additionally, modifications to the neural network architecture or training protocol and extensive hyperparameter tuning could enable solutions for lower ranks than 64 but are out of the scope of this work.

4 Discussion and Conclusions

We introduce *functional tensor decomposition* based PINNs, a novel approach for solving PDEs using PINNs. Our essential contribution is extending the classical variable separable method to PINNs by leveraging function tensor decomposition forms. We show that PINNs can approximate multivariate PDEs by decomposing the solution as the outer product of smaller tensors with controlled ranks, enabling efficient and effective solutions to complex problems.

A critical insight is that such a PINN can learn any PDE irrespective of whether the variable separable form exists. We investigate three tensor decomposition forms incorporated into the general PINN framework to reduce the computational complexity, especially the *curse of dimensionaly.* The primary concept underlying our functional tensor decomposition approach for PINNs is as follows: (1) We decompose the multivariate solution of a PDE and learn each part using an individual neural network, and (2) we use this within the PINN framework, where a loss function with PDE residual and boundary conditions is optimized. Please note that a unique form of decomposition does not need to exist as we estimate the decomposition in a data-driven manner.

We conducted experiments using benchmark PDEs, such as the Klein-Gordon Equation [5] and 3d Helmholtz [29] equation, to demonstrate that our proposed methods significantly outperform existing PINNs. This distinction in accuracy is evident even when employing low numbers of collocation points and ranks. Furthermore, we substantiated our findings by checking whether our method scales to higher dimensional (> 3) PDEs, such as the 5d Poisson equation [37] or (2+1)d flow mixing [4]. We are demonstrating the ability of our method to tackle complex issues with a relatively small number of collocation points and faster speed. The results show that we effectively mitigate the *curse of dimensionality*, enabling PINNs to solve problems in even higher dimensions efficiently.

Several open questions remain unanswered and could provide further interesting research in developing PINNs for high-dimensional PDEs, even though we already outperform existing methods by a factor of ten. Our experiments indicate no best tensor decomposition form, and factors like collocation points, rank, and available physical knowledge play a significant role in overall performance. The Canonical-Polyadic decomposition [12] is the most straightforward representation among the same rank but becomes unstable in higher ranks [25]. While Tucker decomposition [30], with its increased parameter count for the same rank, does experience a curse of dimensionality due to the core tensor, albeit to a lesser extent than others. We assume that Tensor-Train [25] is a good candidate with lower dimensionality and is more stable in higher dimensions. We demonstrate the potential of *functional tensor decomposition* in enhancing PINNs. While further optimization may be possible, our findings already highlight the benefits of this method over traditional numerical approaches.

References

1. Berkhahn, S., Ehrhardt, M.: A physics-informed neural network to model covid-19 infection and hospitalization scenarios. Advances in Continuous and Discrete Models **2022**(1), 61 (2022). https://doi.org/10.1186/s13662-022-03733-5
2. Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q.: JAX: composable transformations of Python+NumPy programs (2018), http://github.com/google/jax
3. Cai, S., Wang, Z., Wang, S., Perdikaris, P., Karniadakis, G.E.: Physics-Informed Neural Networks for Heat Transfer Problems. Journal of Heat Transfer **143**(6), 060801 (04 2021https://doi.org/10.1115/1.4050542
4. Chiu, P.H., Wong, J.C., Ooi, C., Dao, M.H., Ong, Y.S.: Can-pinn: A fast physics-informed neural network based on coupled-automatic-numerical differentiation method. Comput. Methods Appl. Mech. Eng. **395**, 114909 (2022). https://doi.org/10.1016/j.cma.2022.114909
5. Cho, J., Nam, S., Yang, H., Yun, S.B., Hong, Y., Park, E.: Separable physics-informed neural networks. Advances in Neural Information Processing Systems (2023)
6. Coscia, D., Ivagnes, A., Demo, N., Rozza, G.: Physics-informed neural networks for advanced modeling. Journal of Open Source Software **8**(87), 5352 (2023)
7. Cuomo, S., Di Cola, V.S., Giampaolo, F., Rozza, G., Raissi, M., Piccialli, F.: Scientific machine learning through physics-informed neural networks: Where we are and what's next. J. Sci. Comput. **92**(3), 88 (2022). https://doi.org/10.1007/s10915-022-01939-z

8. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Math. Control Signals Systems **2**(4), 303–314 (1989). https://doi.org/10.1007/BF02551274
9. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy (13–15 May 2010), https://proceedings.mlr.press/v9/glorot10a.html
10. Haghighat, E., Raissi, M., Moure, A., Gomez, H., Juanes, R.: A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Comput. Methods Appl. Mech. Eng. **379**, 113741 (2021). https://doi.org/10.1016/j.cma.2021.113741
11. Herath, I.: Multivariate Regression using Neural Networks and Sums of Separable Functions. Ph.D. thesis, Ohio University (04 2022), http://rave.ohiolink.edu/etdc/view?acc_num=ohiou1648166101093853
12. Hitchcock, F.L.: The expression of a tensor or a polyadic as a sum of products. J. Math. Phys. **6**(1–4), 164–189 (1927). https://doi.org/10.1002/sapm192761164
13. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989). https://doi.org/10.1016/0893-6080(89)90020-8
14. Jin, P., Meng, S., Lu, L.: Mionet: Learning multiple-input operators via tensor product. SIAM J. Sci. Comput. **44**(6), A3490–A3514 (2022). https://doi.org/10.1137/22M1477751
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
16. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009). https://doi.org/10.1137/07070111X
17. Krishnapriyan, A.S., Gholami, A., Zhe, S., Kirby, R.M., Mahoney, M.W.: Characterizing possible failure modes in physics-informed neural networks (2021)
18. Lin, C., Maxey, M., Li, Z., Karniadakis, G.E.: A seamless multiscale operator neural network for inferring bubble dynamics. Journal of Fluid Mechanics **929**, A18 (2021https://doi.org/10.1017/jfm.2021.866
19. Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G.E.: Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. Nature Machine Intelligence **3**(3), 218–229 (2021). https://doi.org/10.1038/s42256-021-00302-5
20. Lu, L., Meng, X., Mao, Z., Karniadakis, G.E.: Deepxde: A deep learning library for solving differential equations (7 2019https://doi.org/10.1137/19M1274067
21. Lu, L., Meng, X., Mao, Z., Karniadakis, G.E.: DeepXDE: A deep learning library for solving differential equations. SIAM Rev. **63**(1), 208–228 (2021). https://doi.org/10.1137/19M1274067
22. Maddu, S., Sturm, D., Müller, C.L., Sbalzarini, I.F.: Inverse dirichlet weighting enables reliable training of physics informed neural networks. Machine Learning: Science and Technology **3**(1), 015026 (feb 2022https://doi.org/10.1088/2632-2153/ac3712
23. McClenny, L., Braga-Neto, U.: Self-adaptive physics-informed neural networks using a soft attention mechanism (2022)
24. Moseley, B., Markham, A., Nissen-Meyer, T.: Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations (2021)

25. Oseledets, I.V.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**(5), 2295–2317 (2011). https://doi.org/10.1137/090752286
26. Raisinghania, M.: Ordinary and Partial Differential Equations. S. Chand Publishing (1991), https://books.google.de/books?id=vaorDAAAQBAJ
27. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations (11 2017), http://arxiv.org/abs/1711.10566
28. Rudin, W.: Principles of Mathematical Analysis. International series in pure and applied mathematics, McGraw-Hill (1964), https://books.google.de/books?id=yifvAAAAMAAJ
29. Takamoto, M., Praditia, T., Leiteritz, R., MacKinlay, D., Alesiani, F., Pflüger, D., Niepert, M.: Pdebench: An extensive benchmark for scientific machine learning (2023)
30. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika **31**(3), 279–311 (1966). https://doi.org/10.1007/BF02289464
31. Vemuri, S.K., Büchner, T., Denzler, J.: Estimating soil hydraulic parameters for unsaturated flow using physics-informed neural networks. In: Franco, L., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds.) Computational Science – ICCS 2024. pp. 338–351. Springer Nature Switzerland, Cham (2024https://doi.org/10.1007/978-3-031-63759-9_37
32. Vemuri, S.K., Denzler, J.: Gradient statistics-based multi-objective optimization in physics-informed neural networks. Sensors **23**(21) (202https://doi.org/10.3390/s23218665
33. Wang, S., Sankaran, S., Perdikaris, P.: Respecting causality is all you need for training physics-informed neural networks (2022)
34. Wang, S., Teng, Y., Perdikaris, P.: Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J. Sci. Comput. **43**(5), A3055–A3081 (2021). https://doi.org/10.1137/20M1318043
35. Wang, S., Yu, X., Perdikaris, P.: When and why pinns fail to train: A neural tangent kernel perspective. Journal of Computational Physics **449**, 110768 (2022https://doi.org/10.1016/j.jcp.2021.110768
36. Yu, J., Lu, L., Meng, X., Karniadakis, G.E.: Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. Computer Methods in Applied Mechanics and Engineering **393**, 114823 (Apr 2022https://doi.org/10.1016/j.cma.2022.114823
37. Zeng, C., Burghardt, T., Gambaruto, A.M.: Feature mapping in physics-informed neural networks (pinns) (2024)

Squeeze and Hypercomplex Networks on Leaf Disease Detection

Nazmul Shahadat(✉), Anh Nguyen, and Ritika Lama

Truman State University, Kirksville, MO, USA
nazmul.ruet@gmail.com

Abstract. Detecting agricultural leaf disease is critical for crop yield and quality, where deep attention models offer promising solutions over traditional methods. This paper introduces a novel approach utilizing Squeeze-and-Hypercomplex networks (SHNets) to detect and classify leaf diseases. The existing Squeeze-and-Excitation network (SENet) enhances feature representation through channel-wise (all channels) feature re-calibration. Unlike this, Parameterized hypercomplex multiplication (PHM) based hypercomplex dense layer is used to calculate cross-channel correlations across channels. This enhances the network's representational capacity by adaptively recalibrating cross-channel feature maps and sharing weights among channels. We introduce a novel hypercomplex dense layer to inherit hypercomplex advantages in SE-based attention networks. Moreover, using hypercomplex algebra in network design enables more expressive modeling of inter-channel dependencies, capturing complex patterns in leaf imagery. Our proposed SHNet architecture was trained and evaluated on diverse leaf disease datasets, including disease categories and healthy samples. The experimental results on benchmark datasets unequivocally demonstrate the superiority of our proposed SHNet over the state-of-the-art SENet methods in terms of accuracy and computational complexity. This makes SHNet a highly suitable solution for real-time applications in precision agriculture, where the timely detection and classification of leaf diseases can significantly impact crop yield and quality.

Keywords: SHNet · Squeeze and Hypercomplex Networks · Squeeze-and-Excitation networks · Attention networks · Leaf Disease detection

1 Introduction

Agriculture is crucial for global economic growth and human survival, as all life depends on food. According to the United States Department of Agriculture (USDA), agriculture-related industries contributed roughly $1.530 trillion to the U.S. GDP, 4% of the global GDP, and more than 25% for developing countries' GDP in 2023. Every country prioritizes investing in agricultural innovation,

A. Nguyen and R. Lama—Authors contributed equally.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 47–63, 2025.
https://doi.org/10.1007/978-3-031-78389-0_4

policies, and infrastructure. Due to rapid population growth, urbanization, and climate change, fertile farmland is depleting, hindering crop growth. Innovative methods are needed to grow crops in less fertile soils. Researchers have developed various technologies to enhance crop production.

Due to poor soil minerals and unfavorable climates, crops can develop various diseases, such as spots, dead tissue, discoloration, wilting, stunted growth, and damage [21]. About 85% of plant diseases are caused by fungi and Non-infectious factors, such as nutrient deficiencies or temperature extremes [9]. According to the USDA, plant diseases cost the global economy around $220 billion annually and can cause a loss of 20-40% of global crop production. Plant pathogens and pests are responsible for up to 40% of maize, potato, rice, soybean, and wheat crop yield losses worldwide [25]. Therefore, farmers often use chemical fertilizers haphazardly, highlighting the importance of disease detection in agriculture.

Plants are susceptible to numerous diseases that can significantly impact crop health. Therefore, the diagnosis of plant diseases is crucial to mitigate these risks. Many Artificial Intelligence(AI) and Machine learning(ML) techniques, especially various convolutional neural networks (CNNs) (VGG [31], MobileNet [4,20], ResNet [12,13]), have shown exceptional abilities in diagnosing plant leaf diseases. No one has used an attention mechanism to analyze the diseases of these crops. Several attention-based networks have been introduced, including SENet [8], known for its parameter-efficient architecture. Moreover, the SE blocks recalibrate feature responses to focus on the most informative features and capture dependencies across channels. However, it uses two fully connected (FC) layers, which consumes high costs.

This research introduces a novel parameterized hypercomplex multiplication (PHM)-based FC layer that inherits all properties of SENet and introduces hypercomplex properties that provide better representational feature maps. The PHM layers require fewer computational resources, allow more efficient representations, and construct efficient and robust network using hypercomplex algebra. We analyze some major crop diseases, including rice, corn, wheat, and some others, using this better representational attention mechanism called the Squeeze-and-Hypercomplex network (SHNet). The effectiveness of our SHNet model is demonstrated experimentally on four crop disease detection datasets. Our assessments are based on parameter counts, FLOPs, and testing accuracy.

2 Literature Reviews

2.1 Rice Leaf Diseases Detection

In 2020, Matin et al. used the AlexNet technique to detect rice leaf diseases and demonstrated more than 99% accuracy [15]. In 2021, the model provided by Kathiresan et al., which was tested on a GAN-augmented dataset, achieved an average accuracy of 98.79% [11]. Bari et al. proposed a Faster R-CNN in diagnosing the three rice leaf diseases with accuracy rates of 98.09%, 98.85%, and 99.17%, respectively [3]. Mohapatra et al. used pre-trained InceptionV3 and ResNet152 and achieved a higher accuracy of 97.47% [16]. Yang et al. introduced the DHLC-FPN for the IDADP dataset and achieved 97.44% accuracy [34].

2.2 Wheat Leaf Diseases Detection

Rathore et al. developed a hybrid model called WheCNet, which achieved a validation accuracy of 98% [22]. Kumari et al. analyzed the ResNet model on the wheat leaf disease detection dataset, which attained the highest classification accuracy of 98% [14]. Saraswat et al. proposed a methodology for accurately detecting eight different types of wheat leaf disease [24] and revealed that the model beats all other models, with a classification accuracy of 98.08%.

2.3 Corn Leaf Diseases Detection

VGG16, a CNN version, is used to categorize infected and healthy leaves in a study by Subramanian et al. [31] and recorded an accuracy of 97%. Olayiwola et al. proposed a CNN-based model to identify four corn diseases with a 98.56% accuracy rate [17]. Kumar Sharma et al. utilized ResNext101, ResNext50, and Inception V3, achieving average accuracy levels of 91.59%, 88.43%, and 78.5%, respectively [13]. Yeswanth et al. proposed the ASFESRN model to analyze the PlantVillage Corn Leaf disease dataset and achieved accuracies of 99.7402%, 98.4805%, and 98.961% for X2, X4, X6 image scaling factors, respectively [35].

2.4 New Plant Leaf Diseases Data

Working on an open dataset containing 15200 photos of crop leaves, a ResNet34 was trained by Kumar et al. in 2020, which achieved 99.40% accuracy [12]. Pandian et al. presented a deep CNN with an average testing accuracy of 98.1% [18]. A new plant leaf dataset of 10,851 images of 44 different diseases was tested using a CNN-based attention model achieving an accuracy of 97.33% [36]. An article in 2023 by Binnar et al. detects leaf disease using three deep-learning models: AlexNet, MobileNet, and Inception-v3. They concluded that the MobileNet model is an excellent fit for the plant diseases dataset, with an accuracy of 97.52% [4]. Alqahtani et al. proposed the PlantRefineDet, which tested the PlantVillage data and achieved an accuracy of 99.99% [1].

3 Background Works

3.1 Residual 1D Convolutional Networks

Shahadat and Maida proposed a residual 1D CNN (RCN) to replace any 2D spatial CNN layer in a network to reduce the cost further. They replaced any block's 2D CNN with two 1D DSC layer. To use 1D DSC in 2D input, they separated the inputs into height and width axes for a size of $h \times w$. Each 1D CNN layer is applied to each input axis. The 1D DSC operation is defined in [27]. The n_{th} channel of trainable weight W is applied to the n_{th} of input X to get the n_{th} channel of the output feature map C_o. The computational cost of 1D DSC in RCN is, $Cost_{Conv1D} = h \cdot d_{out} \cdot k$, where d_{out} is the output channel counts. As the RCN block has two layers of 1D CNN, the total cost is twice the original cost. This type of network is used to avoid the vanishing gradient problem.

3.2 Quaternion Convolution Networks

[2] was the first to employ the quaternion number system in a neural network. Quaternion numbers are made up of one real and three imaginary components, expressed as, $Q = r + ix + jy + kz$, where r, x, y, and z are real numbers and i, j, and k are the imaginary vectors. The quaternion number system expands the 2D complex number system to four dimensions. A quaternion vector $Q_V = r + ix + jy + kz$ is convolved with a quaternion filter matrix $Q_F = R + iX + jY + kZ$, where R, X, Y, and Z are real-valued matrices and r, x, y, and z are real-valued vectors. This process is similar to complex convolution. The quaternion convolution (4D hypercomplex network) is defined as [29],

$$\begin{aligned} Q_F \circledast Q_V = (&Rr - Xx - Yy - Zz) \\ +i(&Rx + Xr + Yz - Zy) \\ +j(&Ry - Xz + Yr + Zx) \\ +k(&Rz + Xy - Yx + Zr). \end{aligned} \tag{1}$$

Every input vector is convolved with every kernel to expose cross-channel convolutional processes. The matrix representation of this quaternion CNN (QCNN) is defined in [29], where each kernel is shared among four convolutions, each with four input channels, which explains cross-channel weight sharing. Equation 1 shows 16 real-valued convolutions, but only four kernels are reused. Kernel reuse is how the channel weight sharing occurs [28].

3.3 Parameterized Hypercomplex Multiplication Layer

The main disadvantage of QCNN in neural networks is that it only exists in a relatively limited number of preset dimensions: 4D (Quaternions), 8D (Octonions), and 16D (Sedenions). The need to work within specific dimensionalities restricts the flexibility of neural network architectures using hypercomplex multiplication.

A Parameterized Hypercomplex Multiplication (PHM) is a generalized hypercomplex network proposed by [37]. PHM layers are more flexible than older techniques, learning multiplication rules directly from the data. The PHM FC layer transforms input x into output y, which is represented as, $y = PHM(x) = Hx + b$. Here, b represents a bias vector, and H represents a parameter matrix with dimensions $k \times d$, which is not a standard matrix but is constructed using a sum of Kronecker products. Our work uses four-dimensional PHM layer whose parameter matrix H is defined in Equation 2 [29].

Let the dimension of the PHM module be $D_{phm} = N$. The PHM operation requires that both d and k are divisible by N [29]. $\mathbf{H}$ is the sum of Kronecker products of the parameter matrices, $\mathbf{A}_i \in \mathbb{R}^{N \times N}$ and $\mathbf{S}_i \in \mathbb{R}^{k/N \times d/N}$, where $i = 1 \ldots N$, is defined as $\mathbf{H} = \sum_{i=1}^{N} A_i \otimes S_i$. Parameter reduction comes from reusing matrices A and S. The $\otimes$ is the Kronecker product. H is multiplied by the input in the dense layer. The learnable parameters for $N = 4$ are P_r, P_x, P_y, and P_z where $P \in \mathbb{R}^{1 \times 1}$. For A_i we use the hypercomplex matrix (4 dimensions), which is generated in a similar way to the vectormap convolution [28].

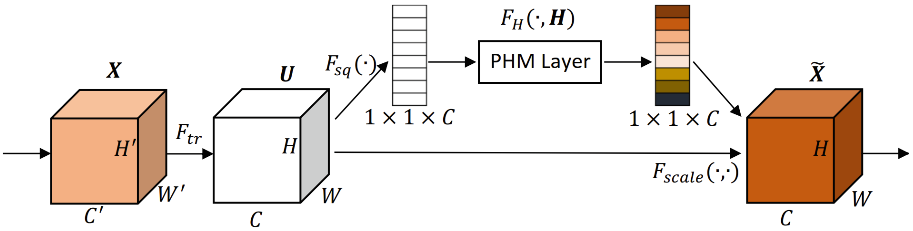


Fig. 1. Proposed Squeeze-and-Hypercomplex block.

$$
\begin{aligned}
H = {} & \underbrace{\begin{bmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{bmatrix}}_{A_1} \otimes \underbrace{\left[P_r\right]}_{S_1} + \underbrace{\begin{bmatrix} 0&-1&0&0\\ 1&0&0&0\\ 0&0&0&1\\ 0&0&-1&0 \end{bmatrix}}_{A_2} \otimes \underbrace{\left[P_x\right]}_{S_2} + \underbrace{\begin{bmatrix} 0&0&-1&0\\ 0&0&0&-1\\ 1&0&0&0\\ 0&1&0&0 \end{bmatrix}}_{A_3} \otimes \underbrace{\left[P_y\right]}_{S_3} \\
& + \underbrace{\begin{bmatrix} 0&0&0&-1\\ 0&0&1&0\\ 0&-1&0&0\\ 1&0&0&0 \end{bmatrix}}_{A_4} \otimes \underbrace{\left[P_z\right]}_{S_4} = \begin{bmatrix} P_r & -P_x & -P_y & -P_z\\ P_x & P_r & P_z & -P_y\\ P_y & -P_z & P_r & P_x\\ P_z & P_y & -P_x & P_r \end{bmatrix}
\end{aligned} \tag{2}
$$

3.4 Squeeze-and-Excitation Network

Squeeze-and-Excitation Network (SENet) are designed in a way to enhance CNNs performance by capturing channel dependencies with fewer computational costs. SENet introduces additional parameters to each channel in a convolutional block, enabling the network to adjust the importance of each feature map and weigh each channel according to its significance [8]. SENet consists of two primary operations: Squeeze and Excitation. Squeeze operation compresses spatial dimensions while retaining channel information, using global pooling operations to condense spatial information into a channel descriptor. The excitation operation utilizes the channel descriptor generated during the squeeze operation and computes a set of channel-wise scaling factors [8]. These factors determine the importance of each channels.

If $V = [v_1, v_2, \ldots, v_c]$ is a set of learned filter channels, the outputs of F_{tr} is expressed as $U = [u_1, u_2, \ldots, u_c]$, where $u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x_s$. Here, $*$ represents convolution operation, and v_c^s is a 2D spatial kernel. The output is calculated by summing across all channels, embedding channel dependencies.

SENets have the capability to dynamically adjust the importance of each feature map, leading to improved performance in various tasks such as image classification and object detection. This capability makes SENets a powerful tool for enhancing CNNs without a great computational cost.

4 Proposed Squeeze-and-Hypercomplex Network

The local receptive fields of standard convolutional filters have limitations as they can only use contextual information within a limited portion of the input. Small receptive field sizes in the lowest network tiers exacerbate this problem. Conventional methods for mapping correlations across channels assume local receptive fields and may hinder the network's ability to discover dynamic, non-linear interactions between channels. In contrast, the SE block uses global information to uncover dynamic, non-linear connections across channels, boosting the network's representative strength and speeding up learning. The SE block is designed to enhance a network's representational power by explicitly modeling the interdependencies between the channels of its convolutional features. It achieves this by using global information to emphasize informative features and selectively suppress less useful ones. The SE block enables the network to perform feature recalibration, thereby enhancing CNN performance.

- The SE blocks recalibrate feature responses to focus on the most informative features and enhance representational capacity.
- The squeeze operation uses global average pooling to reduce each feature map to a single value, compressing the spatial dimensions of each channel into a single value.
- The excitation operation uses two fully connected (FC) layers to generate channel-wise weights. The first FC layer reduces the dimensionality and captures dependencies across channels, while the second FC layer restores the original channel dimensions.
- The excitation operation result undergoes a sigmoid activation function.
- The recalibration step emphasizes important channels and diminishes less important ones, leading to improved feature representation.
- SE blocks can seamlessly enhance existing CNN architectures to learn useful features without significantly increasing computational costs.
- SE blocks add relatively few parameters and computations compared to the overall network, making them efficient in terms of resource usage.

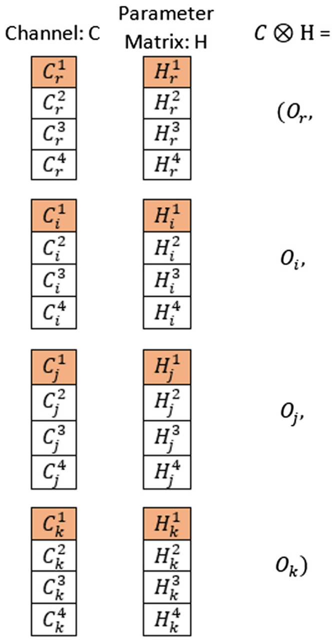

Channel: C
Parameter Matrix: H
$C \otimes$ H =
C_r^1 C_r^2 C_r^3 C_r^4
H_r^1 H_r^2 H_r^3 H_r^4
$(O_r,$
C_i^1 C_i^2 C_i^3 C_i^4
H_i^1 H_i^2 H_i^3 H_i^4
$O_i,$
C_j^1 C_j^2 C_j^3 C_j^4
H_j^1 H_j^2 H_j^3 H_j^4
$O_j,$
C_k^1 C_k^2 C_k^3 C_k^4
H_k^1 H_k^2 H_k^3 H_k^4
$O_k)$

Fig. 2. Single valued quaternion channels and hypercomplex parameter matrix are represented using four channels of real values.

However, the SE block utilizes two FC layers, which is expensive. Without an explicit mechanism, the SE block diminishes the channel counts to reduce the number of parameters. Although the SE block re-calibrates input feature maps, it cannot establish cross-channel correlations. To address these limitations, we introduce a Squeeze-and-Hypercomplex network (SHNet).

Like the Squeeze-and-Excitation network (SENet), we have two stages of network operations: Squeeze and Hypercomplex. We use the squeeze operation to capture the global context of the input feature maps, applying global average pooling to reduce the feature map to a single scalar value. For a vision input with input height H, and width W, these are explained by the following equation [8],

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{ijc} \tag{3}$$

where x_{ijc} represents the value at spatial location (i, j) in channel c and z_c is the global descriptor for channel c.

The excitation layer, mainly two FC layers, was used to apply channel-wise dependencies, emphasize the important channels, and improve feature map representation using channel recalibration. However, we use a novel PHM layer, a generalized hypercomplex network (HCNN) designed for FC layer, to import the advantages provided by FC layer (Excitation) and the advantages (cross-channel correlation concept) provided by HCNNs. The HCNNs provide a better representational feature map [30], and reduce the trainable weights by a factor of $1/N$ for an N-dimensional HCNN [37]. Our proposed SHNet architecture is depicted in Figure 1.

To apply a hypercomplex network in the dense layer, we choose a PHM-based FC layer defined in Section 3.3, where H is the hypercomplex parameter matrix (HPM) calculated using the Equation 2. We rewrite this PHM FC layer as $y = PHM(x) = HC + b$, where, x, C, b, and H, $H \in \mathbb{R}^{N \times N}$, are the input image, the single-valued input channels, the bias, and the PHM, respectively. Although our proposed PHM is generalized, we are going to explain 4D HCNN dense layer or 4D PHM layer ($N = 4$). We use permutation to construct our parameter matrix using $H = H_r, H_i, H_j, H_k$. The parameter matrix for 4D PHM layer is used to calculate the output y using,

$$\begin{aligned}
\begin{bmatrix} \mathscr{R}(C * H) \\ \mathscr{I}(C * H) \\ \mathscr{J}(C * H) \\ \mathscr{K}(C * H) \end{bmatrix} &= \begin{cases} 1 & i = 1 \\ 1 & i = j \\ 1 & j = (i + i - 1) \ mod \ N \\ -1 & \text{else} \end{cases} \odot \begin{bmatrix} H^1 & H^2 & H^3 & H^4 \\ H^4 & H^1 & H^2 & H^3 \\ H^3 & H^4 & H^1 & H^2 \\ H^2 & H^3 & H^4 & H^1 \end{bmatrix} * \begin{bmatrix} C^1 \\ C^2 \\ C^3 \\ C^4 \end{bmatrix} \\
&= \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \end{bmatrix} \odot \begin{bmatrix} H^1 & H^2 & H^3 & H^4 \\ H^4 & H^1 & H^2 & H^3 \\ H^3 & H^4 & H^1 & H^2 \\ H^2 & H^3 & H^4 & H^1 \end{bmatrix} * \begin{bmatrix} C^1 \\ C^2 \\ C^3 \\ C^4 \end{bmatrix}
\end{aligned} \tag{4}$$

where, $\odot$ is the element wise multiplication. So, the Hamiltonian product of two quaternion H (parameter matrix) and C (input channels) is defined as,

$$\begin{aligned} H \otimes C &= C^1H^1 + C^2H^2 + C^3H^3 + C^4H^4 \\ &= (C_r^1, C_i^1, C_j^1, C_k^1)(H_r^1, H_i^1, H_j^1, H_k^1) \\ &+ (C_r^2, C_i^2, C_j^2, C_k^2)(H_r^2, H_i^2, H_j^2, H_k^2) \\ &+ (C_r^3, C_i^3, C_j^3, C_k^3)(H_r^3, H_i^3, H_j^3, H_k^3) \\ &+ (C_r^4, C_i^4, C_j^4, C_k^4)(H_r^4, H_i^4, H_j^4, H_k^4) \end{aligned} \tag{5}$$

where $\otimes$ represents the Hamiltonian product [29], $H \otimes C$, and all of the other symbols in Equation 5 are quaternion numbers. The first line of Equation 5 (RHS) is also depicted in Figure 2, where the terms composing the multiplication operation shown in Equation 6 are highlighted in orange.

Furthermore, we expand the first term C^1H^1 on the right side of Equation 5 (shaded area for the input channel feature map and parameter matrix in Figure 2) by using the distributive property and grouping terms [29], defined as,

$$\begin{aligned} C^1H^1 =&(O_r, O_i, O_j, O_k) \\ =&(C_r^1, C_i^1, C_j^1, C_k^1)(H_r^1, H_i^1, H_j^1, H_k^1) \\ =&(C_r^1H_r^1 - C_i^1H_i^1 - C_j^1H_j^1 - C_k^1H_k^1, \\ &C_i^1H_r^1 + C_r^1H_i^1 + C_j^1H_k^1 - C_k^1H_j^1, \\ &C_j^1H_r^1 + C_r^1H_j^1 + C_k^1H_i^1 - C_i^1H_k^1, \\ &C_k^1H_r^1 + C_r^1H_k^1 + C_i^1H_j^1 - C_j^1H_i^1) \end{aligned} \tag{6}$$

where $\mathbf{re}(C^1H^1) = C_r^1H_r^1 - C_i^1H_i^1 - C_j^1H_j^1 - C_k^1H_k^1$, $\mathbf{i}(C^1H^1) = C_i^1H_r^1 + C_r^1H_i^1 + C_j^1H_k^1 - C_k^1H_j^1$, $\mathbf{j}(C^1H^1) = C_j^1H_r^1 + C_r^1H_j^1 + C_k^1H_i^1 - C_i^1H_k^1$, and $\mathbf{k}(C^1H^1) = C_k^1H_r^1 + C_r^1H_k^1 + C_i^1H_j^1 - C_j^1H_i^1$. It means convolving the real-valued input channel with the real-valued parameter matrix channel to obtain a real-valued scalar. Hence, the real part in Equations 5 equals O_r. Similarly, the other parts O_i, O_j, and O_k are defined as,

$$\begin{aligned} O_r =&\mathbf{r}(C^1H^1) + \mathbf{r}(C^2H^2) + \mathbf{r}(C^3H^3) + \mathbf{r}(C^4H^4) \\ O_i =&\mathbf{i}(C^1H^1) + \mathbf{i}(C^2H^2) + \mathbf{i}(C^3H^3) + \mathbf{i}(C^4H^4) \\ O_j =&\mathbf{j}(C_r^1H_r^1) + \mathbf{j}(C^2H^2) + \mathbf{j}(C^3H^3) + \mathbf{j}(C^4H^4) \\ O_k =&\mathbf{k}(C^1H^1) + \mathbf{k}(C^2H^2) + \mathbf{k}(C^3H^3) + \mathbf{k}(C^4H^4) \end{aligned} \tag{7}$$

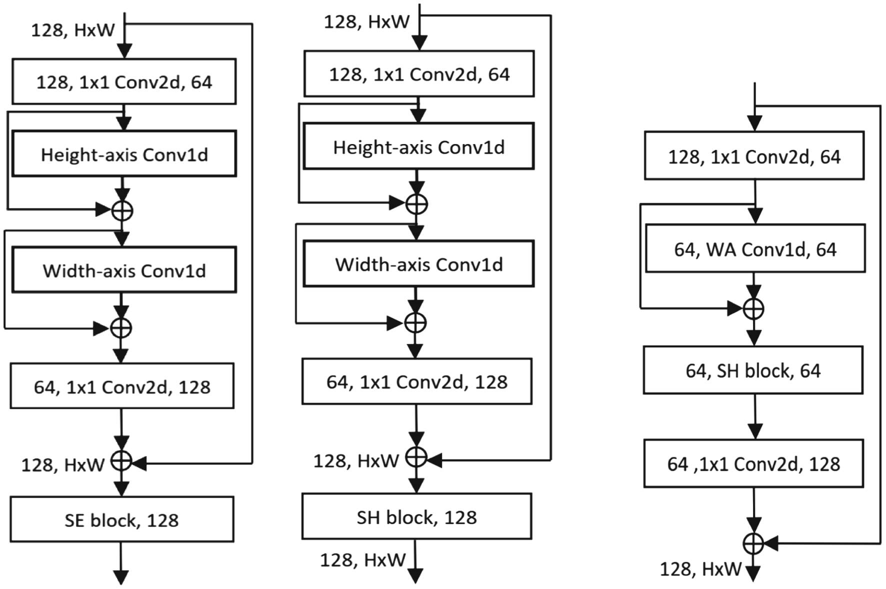
The real component (O_r) of the convolution value is defined as,

$$\begin{aligned} O_r &= C_r \otimes H_r - C_i \otimes H_i - C_j \otimes H_j - C_k \otimes H_k \\ &= C_r^1H_r^1 + C_r^2H_r^2 + C_r^3H_r^3 + C_r^4H_r^4 \\ &- C_i^1H_i^1 - C_i^2H_i^2 - C_i^3H_i^3 - C_i^4H_i^4 \\ &- C_j^1H_j^1 - C_j^2H_j^2 - C_j^3H_j^3 - C_j^4H_j^4 \\ &- C_k^1H_k^1 - C_k^2H_k^2 - C_k^3H_k^3 - C_k^4H_k^4 \end{aligned} \tag{8}$$

Similarly, we can define O_i, O_j, and O_k. Equation 6 can be reexpressed in matrix representation as,

$$\begin{bmatrix} \mathscr{R}(C^1 * H^1) \\ \mathscr{I}(C^1 * H^1) \\ \mathscr{J}(C^1 * H^1) \\ \mathscr{K}(C^1 * H^1) \end{bmatrix} = \begin{bmatrix} H_r^1 & -H_i^1 & -H_j^1 & -H_k^1 \\ H_i^1 & H_r^1 & -H_k^1 & H_j^1 \\ H_j^1 & H_k^1 & H_r^1 & -H_i^1 \\ H_k^1 & -H_j^1 & H_i^1 & H_r^1 \end{bmatrix} * \begin{bmatrix} C_r^1 \\ C_i^1 \\ C_j^1 \\ C_k^1 \end{bmatrix} \tag{9}$$

In Equation 9, each kernel channel is convolved with the corresponding image channel. Equation 6 and the graphical explanation in Figure 2 shows that each parameter of the trainable parameter matrix is used four times over the sixteen multiplications in Equations 5 and 9. Here, each input channel is convolved with all parameter matrix channels, which reveals cross-channel correlations. Parameter of the trainable parameter matrix reuse is how weight sharing occurs. [28,29] described weight sharing in the Hamilton product. The other three components of the sum in Equation 5 (i.e., C^2H^2, C^3H^3, C^4H^4) have the same structure as Equation 6, so the nature of the weight sharing is the same for all terms.



(a) SE-RCN (b) SH-RCN (c) Block with SHNet

Fig. 3. Block types. "SE" and "SH" stand for Squeeze-and-Excitation, and Squeeze-and-Hypercomplex, respectively. (a) SE layer applied after the RCN block, (b) SH layer applied after the RCN block, and (c) SHNet replaces SENet from 1D CNN with SENet found in [26].

We apply this new parameter-efficient hypercomplex FC layer to replace the FC layers in SENet and construct our novel parameter-efficient architecture called Squeeze-and-Hypercomplex network (SHNet). This SHNet architecture can provide better representational feature maps than the SENet. Also, SHNet reduces trainable parameters than the SENet in two ways: (1) the SHNet uses a hypercomplex FC layer than the two FC layers in SENet, and (2) the Hypercomplex FC layer consumes $1/N$ times fewer parameters than the real-valued FC layer. We apply our proposed SHNet block to replace the SENet from the existing parameter-efficient 1D CNN networks with SENet architecture, depicted in Figures 3a, 3b, and 3c. In these ways, any network can be constructed with our proposed SHNet architecture to boost the network performance and reduce the network's trainable parameters than the SENet.

5 EXPERIMENTAL RESULTS

5.1 Dataset Description

This paper experiments on the "Rice Leaf Disease" [23], "Wheat Leaf Disease" [10], "Corn Leaf Disease"[32], and "New Plant Leaf Disease" [33] kaggle datasets. Among these, the rice leaf disease dataset consists of 5,932 images representing four distinct types of rice leaf diseases: Bacterial blight, Blast, Brown Spot, and Tungro [23]. The Wheat Leaf Disease Dataset includes five classes: 1,658 images of healthy leaves, 1,256 images of Brown Rust disease, 939 images of Loose Smut disease, 349 images of Septoria disease, and 1,395 images of Yellow Rust disease [10]. Arun Pandian et al. contributed to the Corn Leaf Disease Dataset, a dataset for classifying corn or maize plant leaf diseases, in 2019. This dataset was based on two popular datasets: PlantVillage and PlantDoc [32]. We also constructed another corn dataset. We collected all corn disease and healthy images for four classes: 2052 images for gray spots, 2384 images for common rust, 2324 healthy images, and 2385 images for blight diseases from new plant disease dataset [33]. Moreover, the new Plant Diseases Dataset contains approximately 87K RGB images of healthy and diseased crop leaves, divided into 38 different categories [33]. We divided all of the datasets into two parts: 80% for the training set and 20% for the testing set.

Table 1. Cost comparisons of our proposed SHNet with 1D CNN (like the Figure 3c) and original SENet with 1D CNN architectures [26].

Architecture	Models	Params	FLOPs
23-1	SENet	0.33M	4.3M
	SHNet	0.3M	4.1M
23-2	SENet	1.4M	12.3M
	SHNet	1.1M	11.9M
44-1	SENet	0.58M	6.3M
	SHNet	0.54M	5.9M
44-2	SENet	2.4M	18.4M
	SHNet	2.06M	17.8M

5.2 Method

We applied our proposed SHNet to 1D CNN with SENet and constructed our block 1D CNN with SHNet. Like [26], which introduced 1D CNN with SENet, we also applied our proposed 1D CNN with SHNet block to SqueezeNext network architecture [7] and constructed our proposed network. We employed an identical training protocol as [26] for our proposed SqueezeNext (SqueezeNext block is replaced by our proposed 1D CNN with SHNet block) architectures to ensure a fair comparison. Like original work (1D CNN with SENet [26]), this work also utilizes the block multipliers "[6, 6, 8, 1]" and "[12, 12, 16, 2]", respectively, to construct 23-layer and 44-layer networks. We also analyzed these 23 and 44-layer architectures with widening factors 1 and 2, constructed 23-1, 23-2, 44-1, and 44-2 network architectures. Training was done on 64 batch sizes for all architectures. To investigate scalability, we trained the SGD (stochastic gradient descent) optimizer on the normalized input images. The images were normalized using the per-channel mean and standard deviation. All models were run using the linearly warmed-up learning for the first ten epochs from zero to 0.1 and then used cosine learning scheduling from epochs 11 to 150. The experiments were run on a workstation with an Intel(R) i9-13900K CPU, 128 GB memory, and NVIDIA RTX A6000 GPU (48GB).

Table 2. Performance evaluation on different leaf disease detection datasets using the 1D CNN with original SENet and our proposed SHNet architectures.

Dataset Type	Dataset Class	Models	Accuracy	
			SENet	SHNet
Wheat	5	23-1	96.93	98.68
		23-2	97.78	98.99
		44-1	97.27	98.87
		44-2	97.98	99.01
Rice	4	23-1	99.49	100
Corn	4	23-1	96.54	98.79
		23-2	97.31	98.89
		44-1	96.87	98.88
		44-2	97.37	98.91
Corn New Plant	4	23-1	98.91	99.29
		23-2	99.34	99.78
		44-1	99.01	99.35
		44-2	99.49	99.89
New Plant	38	23-1	99.41	99.94
		23-2	99.58	99.98
		44-1	99.47	99.96
		44-2	99.61	99.99

5.3 Result Analysis

This paper achieves a new height of performance accuracy in leaf disease detection. Our new proposed model yields impressive accuracy across multiple plant disease datasets. As previously described in section 5.1, we tested our model on wheat, rice, and corn plant datasets along with some other new plant disease detection datasets. We tested our proposed SHNet along with the original SENet model, and the experimental results are shown in Table 2. Even with fewer parameters and Flops (shown in Table 1), our novel model, when tested on the wheat leaf, was able to achieve 99.01% accuracy compared to 97.98% for SENet using 44-2 layers network. For all network architectures, our proposed SHNet performed better than the original SENet on wheat leaf data.

For the rice leaf disease dataset, our smallest model (23-1 network architecture) achieved 100% accuracy, whereas the original SENet performed 99.49. In the context of this data, it is notable that the original SENet architectures, with 23 and 44 layers and widening factors of 1 and 2, were unable to achieve the performance level demonstrated by the SHNet's 23-1 network architecture. Our SHNet was tested on corn data and corn new plant data and outperformed similarly to the wheat and rice data.

To analyze a dataset with more categories, we ran our model on a new plant disease dataset with 38 categories, where our SHNet beat the original SENet model for all network architectures; we tested our SHNet for 44 layer with a widening factor 2 achieved 99.99% performance.

Our SHNet model outperformed the SENet model across all datasets and models in this experimental analysis. In the case of cost comparison, Table 1 describes that our proposed SHNet consumes fewer parameters and FLOPs than the original SENet architecture in all the scenarios. 1D CNN with SENet and SHNet are used to evaluate these costs.

5.4 Comparison with the Literature

This section compares our new proposed model with other state-of-the-art performing models across four different datasets. SHNet outperformed other models like SVM, WheCNet, ResNet50, and DL Model by achieving an accuracy level of 99.01% in wheat leaf disease classification, shown in Table 3. Similarly, our model achieved a remarkable accuracy of 100% in the rice leaf dataset by beating the other models. Furthermore, we tested our model in the Corn leaf dataset. It achieved 99.89% accuracy, beating VGG16, ResNet, CNN, ResNext101, ASFESRN, and Ghost CNN. Finally, in the New Plant Disease dataset with 38 categories, our model once again achieved a remarkable accuracy of 99.99% compared to existing models like ResNet43, CNN, DCNN, MobileNet, and PlantRefineDet. Our SHNet model showed a similar level of accuracy as PlantRefineDet proposed by Algahtani et al. [1]. This is a remarkable level of performance achieved by our model, considering that PlantRefineDet is a computer-aided model, and our model is lightweight and mobile-embedded, which consumes only $2.06M$ parameters and $17.8M$ FLOPs. Our model's adaptability and outstanding performance, despite its lightweight design, confirms its potential in real-world scenarios.

5.5 Ablation Study

This section examines how different attention mechanisms (SE and ours SH) affect the RCN-based SqueezeNext model for detecting plant leaf diseases from the New Plant diseases dataset. We compared the RCN-based SqueezeNext 23-1 layer architectures regarding top-1 validation accuracy, parameters, and FLOPs as listed in Table 4. Without an attention model, RCN-SqueezeNext achieves an accuracy of 98.99% for $0.31M$ parameters and $4.08M$ FLOPs. Then, we utilized the SE attention layer in different network stages. The more SE attention layers are used, the better the network performs. Extending SE attention to stages 3 and 4, the model increases its accuracy from 99.07 (performance when SE is only applied to stage 4) to 99.28% and reaches 99.35% when SE is applied in

Table 3. State-of-the-art result studies on plant leaf disease detection datasets.

Category	Model	Architecture	Datasets	Accuracy
Wheat Leaf Diseases	El et al. [6]	SVM	Kaggle Data	98
	Rathore et al. [22]	WheCNet		98
	Kumari et al. [14]	ResNet50		98
	Saraswat et al. [24]	DL Model		98.08
	SH-RCN-SqueezeNext (Our)	44-2	Kaggle Data	**99.01**
Rice Leaf Diseases	Matin et al. [15]	AlexNet	Kaggle Data	99
	Pothen et al. [19]	SVM		94.6
	Kathiresan et al. [11]	RiceDenseNet	Open Sources	98.69
	Bari et al. [3]	Faster R-CNN	Open Sources	99.17
	Mohapatra et al. [16]	CNN	Open Sources	97.47
	Yang [34]	DHLC-FPN	IDADP	97.44
	SH-RCN-SqueezeNext (Our)	23-1	Kaggle Data	**100**
Corn Leaf Diseases	Subramanian et al. [31]	VGG16	Kaggle Data	97
	Olayiwola et al. [17]	CNN	Kaggle Data	98.56
	Kumar et al. [13]	ResNext101	Kaggle Data	91.59
	Yeswanth et al. [35]	ASFESRN	PlantVillage	99.74
	SH-RCN-SqueezeNext (Our)	44-1		**99.89**
New Plant Diseases	Kumar et al. [12]	ResNet34	Open Sources	99.4
	Deepalakshmi et al. [5]	CNN	Open Sources	94.5
	Pandian et al. [18]	DCNN	Open Sources	98.1
	Zamani et al. [36]	CNN	Open Sources	97.33
	Binnar et al. [4]	MobileNet	NPD	99.07
	Alqahtani et al. [1]	PlantRefineDet	PlantVillage	99.99
	SH-RCN-SqueezeNext (Our)	44-2	PlantVillage	**99.99**

stages 2, 3, and 4. When SE attention is applied to all four stages, the model achieved an accuracy of 99.41%, the highest among SE attention with $0.335M$ parameters and $4.35M$ FLOPs.

The integration of our proposed SH attention mechanism performs even better than that of the network with the SE layer. Applying SH attention at stage 4 alone leads to an accuracy of 99.77%, outperforming the performance of applying SE in all network stages. When SH attention is extended over stages 3 and 4, it yields an accuracy of 99.78%. Applying the SH attention mechanism on the last three stages and all four stages resulted in an overall accuracy of 99.91% and 99.94%. The results unequivocally show that the SH attention mechanism consistently enhances the model's performance compared to the SE mechanism, particularly when applied at multiple stages.

Table 4. Analyze New Plant diseases dataset using attention mechanisms (without attention layer, SE attention, and our proposed SHNet attention) on different stages of the RCN-based SqueezeNext 23-1 layer architecture.

Models	Attention on Network stages				Params	FLOPs	Accuracy
	Stage 1	Stage 2	Stage 3	Stage 4			
RCN-SqueezeNext					0.31M	4.08M	98.99
SE-RCN-SqueezeNext				✓	0.316M	4.14M	99.07
SE-RCN-SqueezeNext			✓	✓	0.324M	4.23M	99.28
SE-RCN-SqueezeNext		✓	✓	✓	0.333M	4.31M	99.35
SE-RCN-SqueezeNext	✓	✓	✓	✓	0.335M	4.35M	99.41
SHNet-RCN-SqueezeNext				✓	0.42M	44.9M	99.77
SHNet-RCN-SqueezeNext			✓	✓	0.42M	45M	99.78
SHNet-RCN-SqueezeNext		✓	✓	✓	0.44M	45.2M	99.91
SHNet-RCN-SqueezeNext	✓	✓	✓	✓	0.31M	4.1M	99.94

6 Conclusion

This study explores the efficacy of Squeeze-and-Hypercomplex Networks in detecting wheat, rice, corn, and new plant leaf diseases. This model exhibited superior performance in capturing complex patterns and relationships within the data. Using hypercomplex numbers, HCNNs can effectively model complex dependencies for improved disease detection accuracy, making them ideal for precise disease identification. The fusion of squeeze-and-excitation mechanisms with hypercomplex algebra provides a powerful and efficient framework for leaf disease detection, offering significant improvements in performance and scalability for related applications, and showing state-of-the-art results on some tested datasets. Our SHNet introduced cross-channel feature representation and feature recalibration, improving the model's performance.

This work's limitation is that the proposed model only tested four disease datasets. Due to machine limitations, we were unable to test some state-of-the-art datasets. Future work will focus on further optimizing these networks for other real-world applications and integrating this approach with some pre-trained network architectures.

References

1. Alqahtani, Y., Nawaz, M., Nazir, T., Javed, A., Jeribi, F., Tahir, A.: An improved deep learning approach for localization and recognition of plant leaf diseases. Expert Syst. Appl. **230**, 120717 (2023)

2. Arena, P., Fortuna, L., Occhipinti, L., Xibilia, M.G.: Neural networks for quaternion-valued function approximation. In: Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94. vol. 6, pp. 307–310. IEEE (1994)
3. Bari, B.S., Islam, M.N., Rashid, M., Hasan, M.J., Razman, M.A.M., Musa, R.M., Ab Nasir, A.F., Majeed, A.P.A.: A real-time approach of diagnosing rice leaf disease using deep learning-based faster r-cnn framework. PeerJ Computer Science **7**, e432 (2021)
4. Binnar, V., Sharma, S.: Plant leaf diseases detection using deep learning algorithms. In: Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021. pp. 217–228. Springer (2023)
5. Deepalakshmi, P., Lavanya, K., Srinivasu, P.N., et al.: Plant leaf disease detection using cnn algorithm. International Journal of Information System Modeling and Design (IJISMD) **12**(1), 1–21 (2021)
6. El-Sayed, R., Darwish, A., Hassanien, A.E.: Wheat leaf-disease detection using machine learning techniques for sustainable food quality. In: Artificial Intelligence: A Real Opportunity in the Food Industry, pp. 17–28. Springer (2022)
7. Gholami, A., Kwon, K., Wu, B., Tai, Z., Yue, X., Jin, P., Zhao, S., Keutzer, K.: Squeezenext: Hardware-aware neural network design. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 1638–1647 (2018)
8. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
9. Isleib, J.: Signs and symptoms of plant disease: Is it fungal, viral or bacterial? (Oct 2023), https://www.canr.msu.edu/news/signs_and_symptoms_of_plant_disease_is_it_fungal_viral_or_bacterial
10. jayaprakashpondy: wheat leaf disease. https://www.kaggle.com/datasets/jayaprakashpondy/wheat-leaf-disease, accessed: 2024-07-01
11. Kathiresan, G., Anirudh, M., Nagharjun, M., Karthik, R.: Disease detection in rice leaves using transfer learning techniques. In: Journal of Physics: Conference Series. vol. 1911, p. 012004. IOP Publishing (2021)
12. Kumar, V., Arora, H., Sisodia, J., et al.: Resnet-based approach for detection and classification of plant leaf diseases. In: 2020 international conference on electronics and sustainable communication systems (ICESC). pp. 495–502. IEEE (2020)
13. Kumar Sharma, N., Kalyani Immadisetty, B., Govina, A., Chandra Reddy, R., Choubey, P.: Corn leaf disease detection using resnext50, resnext101, and inception v3 deep neural networks. In: Machine Vision and Augmented Intelligence: Select Proceedings of MAI 2022, pp. 303–313. Springer (2023)
14. Kumari, N., Saini, B.: Fully automatic wheat disease detection system by using different cnn models. In: Sentiment Analysis and Deep Learning: Proceedings of ICSADL 2022, pp. 351–365. Springer (2023)
15. Matin, M.M.H., Khatun, A., Moazzam, M.G., Uddin, M.S.: An efficient disease detection technique of rice leaf using alexnet. Journal of Computer and Communications **8**(12), 49–57 (2020)
16. Mohapatra, S., Marandi, C., Sahoo, A., Mohanty, S., Tudu, K.: Rice leaf disease detection and classification using a deep neural network. In: International Conference on Computing, Communication and Learning. pp. 231–243. Springer (2022)
17. Olayiwola, J.O., Adejoju, J.A.: Maize (corn) leaf disease detection system using convolutional neural network (cnn). In: International Conference on Computational Science and Its Applications. pp. 309–321. Springer (2023)

18. Pandian, J.A., Kanchanadevi, K., Kumar, V.D., Jasińska, E., Goňo, R., Leonowicz, Z., Jasiński, M.: A five convolutional layer deep convolutional neural network for plant leaf disease detection. Electronics **11**(8), 1266 (2022)
19. Pothen, M.E., Pai, M.L.: Detection of rice leaf diseases using image processing. In: 2020 fourth international conference on computing methodologies and communication (ICCMC). pp. 424–430. IEEE (2020)
20. Ramadan, S.T.Y., Sakib, T., Haque, M.M.U., Sharmin, N., Rahman, M.M.: Generative adversarial network-based augmented rice leaf disease detection using deep learning. In: 2022 25th International Conference on Computer and Information Technology (ICCIT). pp. 976–981. IEEE (2022)
21. Randaci, A.: Common plant diseases (Nov 2021), https://earthsally.com/disease-control/common-plant-diseases.html
22. Rathore, N.P.S., Prasad, L.: Hybrid deep learning model to detect uncertain diseases in wheat leaves. Journal of Uncertain Systems **15**(03), 2241004 (2022)
23. Sankalana, N.: rice leaf disease image. https://www.kaggle.com/datasets/nirmalsankalana/rice-leaf-disease-image?resource=download, accessed: 2024-07-01
24. Saraswat, S., Batra, S., Neog, P.P., Sharma, E.L., Kumar, P.P., Pandey, A.K.: An efficient diagnostic approach for multi-class classification of wheat leaf disease using deep transfer and ensemble learning. In: 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT). pp. 544–551. IEEE (2024)
25. Savary, S., Willocquet, L., Pethybridge, S.J., Esker, P., McRoberts, N., Nelson, A.: The global burden of pathogens and pests on major food crops. Nature ecology & evolution **3**(3), 430–439 (2019)
26. Shahadat, N.: Mobile-based deep convolutional networks for malaria parasites detection from blood cell images. In: 2023 26th International Conference on Computer and Information Technology (ICCIT). pp. 1–6. IEEE (2023)
27. Shahadat, N., Maida, A.S.: Deep residual axial networks. arXiv preprint arXiv:2301.04631 (2023)
28. Shahadat, N., Maida, A.S.: Enhancing resnet image classification performance by using parameterized hypercomplex multiplication. arXiv preprint arXiv:2301.04623 (2023)
29. Shahadat, N., Maida, A.S.: Cross channel weight sharing for image classification. Image Vis. Comput. **141**, 104872 (2024)
30. Shahadat, N., Maida, A.S.: Improving axial-attention network via cross-channel weight sharing. In: The International FLAIRS Conference Proceedings. vol. 37 (2024)
31. Subramanian, M., Lv, N.P., VE, S.: Hyperparameter optimization for transfer learning of vgg16 for disease identification in corn leaves using bayesian optimization. Big Data **10**(3), 215–229 (2022)
32. Unknown6874: corn leaf disease dataset. https://www.kaggle.com/datasets/unknown6874/corn-leaf-disease-dataset, accessed: 2024-07-01
33. Vipoooool: new plant diseases dataset. https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset, accessed: 2024-07-01
34. Yang, H., Deng, X., Shen, H., Lei, Q., Zhang, S., Liu, N.: Disease detection and identification of rice leaf based on improved detection transformer. Agriculture **13**(7), 1361 (2023)
35. Yeswanth, P., Deivalakshmi, S.: Asfesrn: bridging the gap in real-time corn leaf disease detection with image super-resolution. Multimedia Syst. **30**(4), 175 (2024)

36. Zamani, A.S., Anand, L., Rane, K.P., Prabhu, P., Buttar, A.M., Pallathadka, H., Raghuvanshi, A., Dugbakie, B.N.: [retracted] performance of machine learning and image processing in plant leaf disease detection. J. Food Qual. **2022**(1), 1598796 (2022)
37. Zhang, A., Tay, Y., Zhang, S., Chan, A., Luu, A.T., Hui, S.C., Fu, J.: Beyond fully-connected layers with quaternions: Parameterization of hypercomplex multiplications with $1/n$ parameters. arXiv preprint arXiv:2102.08597 (2021)

Mangoes Ripeness Grading: Vision Based Approach

D. S. Guru and D. Nandini(✉)

Department of Studies in Computer Science, University of Mysore, Manasagangotri,
Mysore 570006, Karnataka, India
dsg@compsci.uni-mysore.ac.in, nandiniloku@gmail.com

Abstract. In this paper, we introduce a novel application domain which is first of its kind to the vision transformer based deep learning model. We propose a model for ripeness grading of mangoes using vision transformers. Our approach divides the mango image into patches, which are then linearly projected and transformed into a sequence of embeddings. To retain positional information, positional encodings are added to these patches. Additionally, for image classification, learnable class tokens are included at the start of this sequence of embeddings. The resulting sequence is passed through multiple multi-head self-attention (MSA) layers to capture both local and global dependencies and to interpret the spatial relationships among patches. Further to improve the classification performance, we explored five data augmentation strategies to synthetically induce additional data for training. Moreover, different vision transformer models are investigated with and without pre-trained weights while training of neural network. This study is demonstrated through an experimentation on a dataset of 979 images of Alphonso mango variety belonging to four classes particularly, unripen, ripened, over-ripened without internal defects and over-ripened with internal defects. The vision transformers viz., ViT_Base_16, ViT_Large_16 and ViT_Huge_14 is considered for experimentation. The results of the experimentation demonstrated that, the ViT_Huge_14 with pre-trained weight and with data augmentation gives average accuracy of 92.78%, which is better than 85.40% quoted in the existing work of mango grading using conventional machine learning on the same dataset (Raghavendra et al., 2020).

Keywords: Mango Grading · Vision Transformers · Pre-trained weights

1 Introduction

Mango ripeness grading is a critical process in the agricultural and food industry that aims to assess the maturity stage of mango fruits based on various criteria such as color, firmness, aroma, and sugar content. This grading process is essential for ensuring optimal harvesting times, determining storage conditions, and meeting market demands for mangoes at different stages of ripeness.

The objective of mango ripeness grading is to categorize mango fruits into distinct levels of ripeness, typically ranging from unripe or immature to fully ripe. Each stage

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 64–78, 2025.
https://doi.org/10.1007/978-3-031-78389-0_5

of ripeness influences not only the flavor and texture of the mango but also its shelf life and suitability for different culinary purposes and distribution channels.

Effective mango ripeness grading systems are crucial for optimizing harvest quality, minimizing post-harvest losses, and enhancing consumer satisfaction by delivering mangoes that meet desired ripeness preferences. As the demand for high-quality fruits continues to rise globally, accurate and efficient mango ripeness grading methodologies play a pivotal role in supporting sustainable agricultural practices and enhancing economic outcomes for mango growers and stakeholders across the supply chain.

Traditional mango ripeness grading typically involves manual inspection and assessment by experienced personnel based on sensory evaluation and visual inspection. Whereas, AI-driven computer vision systems analyze images of mangoes to assess color, shape, and texture, providing quantitative data for ripeness grading. This method reduces reliance on subjective human judgment.

Recently artificial intelligence (AI) has been increasingly applied to mango ripeness grading, by leveraging various techniques to automate and enhance the accuracy of this critical process, and a significant research efforts are underway in this field, considering that, significant research and development efforts have been directed towards automating the grading and sorting of mangoes using traditional machine learning methods, which depends on extracting hand crafted feature like color histogram shape texture which are not suitable for complex tasks and requires extensive analysis on features by domain experts.

Sa’ad et al., (2015) and Roomi et al., (2012) focused on attributes related to external appearances such as size, color, shape, weight, and defects. Amruta and Wakode (2021) investigated Kesar mango grading based on multiple criteria including maturity, size, and shape. Agilandeeswari et al., (2017) employed SVM classifiers for mango grading. Supekar and Wakode (2020) conducted a comprehensive analysis on mango grading based on appearance and conducted a parameter-wise survey. Khoje and Shrikant (2012) explored shape-based features such as region, contour, and wavelet. Ripeness attributes were also considered by Salunkhe et al., (2015), Chhabra et al., (2011), Nayeli et al., (2012), Nandi et al., (2014), Mansor et al., (2014), and Vyas et al., (2014). Bakar et al., (2020) focused on mango region extraction and color analysis using total soluble solids (TSS). Raghavendra et al., (2020) reported on L*a*b color feature extraction and extensively studied conventional classifiers for mango grading using a hierarchical classification method.

Over the past few years, deep learning prototypes such as convolutional neural networks (CNNs) have garnered significant recognition in the field of computer vision, due to their adeptness at autonomously learning intricate patterns from images. Recently, there has been a surge of deep learning models tailored for mango grading, as evidenced by the literature available. Bhole and Kumar (2020) introduced a transfer learning approach using a pre-trained SqueezeNet model for mango grading, achieving a reported testing accuracy of 92.27%. Zheng and Huang (2021) introduced an ultra-lightweight SqueezeNet CNN by adjusting and optimizing hyperparameters for mango grading, achieving an accuracy of 97.37%. Gururaj et al., (2022) utilized features extracted from convolutional neural networks to assess mango maturity and ripeness. Additionally, they performed mango variety classification based on shape and color features, in conjunction with traditional classifiers.

Vision transformers (ViTs) represent a recent innovation in the field of computer vision, particularly in image classification tasks. Traditionally, convolutional neural networks (CNNs) have been the dominant architecture for such tasks due to their ability to capture spatial hierarchies in images. However, transformers initially developed for natural language processing (NLP), the seminal work by Vaswani et al., (2017) presents the transformer model, a groundbreaking architecture that has fundamentally reshaped the landscape of natural language processing (NLP). This model departs from traditional sequence modelling approaches by entirely discarding the need for recurrence and convolution, relying instead on self- attention mechanisms to capture dependencies within sequences. This innovation has not only streamlined the modelling process but also significantly advanced the efficiency and performance of NLP system. Later the vision transformer (ViTs) was first introduced by Dosovitskiy et al., (2019), drawing inspiration from the successful transformer models in NLP pioneered by Devlin et al., (2019).

A vision transformer (ViTs) differs from traditional convolutional neural networks (CNNs) primarily in how it processes image data. In CNNs, each layer uses filters (convolutions) to extract local features from specific regions of the input image. These filters slide across the image, capturing features such as edges, textures, and patterns in a hierarchical manner. This approach limits the model's ability to consider relationships between distant parts of the image simultaneously.

In contrast, a vision transformer takes the entire image as input and processes it in a holistic manner using self-attention mechanisms. Self-attention allows ViTs to capture dependencies between all image patches, enabling them to learn global relationships and long-range dependencies across the entire image. By attending to all parts of the image simultaneously, ViTs can potentially better understand complex spatial relationships and capture context that may be missed by CNNs relying on localized feature extraction.

Here is a concise list of literature on Vision Transformers (ViTs) for computer vision tasks: Parez et al., (2023) introduced an optimized vision transformer approach for detecting plant diseases. Rizzo et al., (2023) noted in their survey on fruit ripeness that attention enables the acquisition of a weighted sum of input token embeddings, which can be manipulated in diverse manners. Knott et al., (2023) introduced a pre-trained vision transformer for detecting apple defects and assessing banana ripeness, achieving an accuracy of 90%. Bazi et al., (2021) proposed remote sensing image classification utilizing the vision transformer model. Yu et al., (2022) employed inception architecture and cross-channel feature learning for identifying plant diseases. Recently, deep learning algorithms such as generative adversarial networks (GANs) and Vision Transformers have become crucial in plant health monitoring, irrigation management, weed detection, and yield estimation (Dhanya et al., 2022). Khan et al. (2023) proposed a convolutional transformer for tomato grading under various conditions such as lighting, ripeness, and occlusion. Thai et al., (2022) investigated leaf disease detection using a vision transformer model, introducing a pruning algorithm to select crucial heads in each layer. Wang et al., (2022) focused on enhancing convolutional neural networks with attention and feature fusion modules to emphasize both local and global features for tomato disease detection. Shahi et al., (2022) utilized features from convolutional layers to capture high-level object-based information, integrating attention modules to highlight semantic details for fruit classification. Xiao et al., (2023) explored Swin transformers and MLPs for grading pears and apples.

It is notable from the above survey that the majority of the studies concentrate exclusively on grading unripe and ripe mangoes. However, assessing the ripeness of mangoes is essential for consumer preferences, global trade, and the food industry. Mangoes at varying stages of ripening are employed in diverse products like confections, pulps, beverages, and frozen desserts. Typically, distinguishing between ripe and unripe mangoes is relatively simple due to significant differences between classes. However, grading mangoes that are unripe, ripe, and over-ripe is challenging due to minimal variation between these stages. Therefore, our study focuses on enhancing a classification model capable of distinguishing mangoes at various ripeness stages. Over-ripe mangoes often exhibit black marks on their external surfaces, leading to assumptions of internal defects and subsequent rejection. However, many over-ripe mangoes with such marks are actually safe for consumption and free of internal defects.

Grading mango ripeness presents challenges due to subtle variations in external appearance and texture between unripe, ripe, and over-ripe stages. Currently, there is a lack of research in the literature on the application of Vision Transformers (ViTs) for mango grading. ViTs offer a promising approach by leveraging their capability to capture global dependencies and spatial relationships across the entire mango image simultaneously. This holistic view could potentially enhance the accuracy and robustness of ripeness classification compared to traditional methods. Adopting ViTs for mango ripeness grading aims to advance automated systems, improving the efficiency and reliability of fruit quality assessment.

To effectively tackle the aforementioned challenges, we propose a novel model for classifying Alphonso mangoes into four categories: unripe, ripe, over-ripe without internal defects, and over-ripe with internal defects based on vision transformers.

The key contributions of this study include:

- Expanding the initial dataset using five distinct data augmentation methods: rotation, horizontal and vertical flipping, brightness adjustment, and distortion, aimed at enhancing model performance.
- To comprehensively investigate Vision Transformers for mango grading, we have utilized all three models: ViT_Base_16, ViT_Large_16, and ViT_Huge_14. To mitigate challenges related to small dataset size and optimize computational resources, we incorporated pre-trained weights for each model variant.
- Through this extensive experimentation, we have demonstrated how data augmentation techniques such as rotation and flips contribute significantly to Vision Transformers, particularly due to their impact on patch positioning.
- To assess the effectiveness of the proposed system and explore an optimized model, the data samples are divided into two sets: one without data augmentation comprising 979 mango images across four classes, and another set with data augmentation comprising 1566 mango images across the same four classes.

- The proposed model is contrasted with other deep learning architectures, including CNNs and other pre trained network such as VGG 16, Inception V3, EffientNetB0 and ResNet50. Additionally, it is also evaluated against other existing models for mango ripeness grading.

2 Materials and Methods

2.1 Data Sets and Experimentation Setup

The mango dataset (Raghavendra et al., 2020) utilized in our study comprises 979 Alphonso mango images extracted from 230 mangoes at various ripening stages. The images have a black background and dimensions of 2267 x 1701 pixels. This dataset encompasses four classes: unripe, ripe, over-ripe without internal defects, and over-ripe with internal defects, as detailed in Table 1. To accommodate the use of pre-trained weights, our dataset undergoes preprocessing to align with the resolutions employed during pre-training. Specifically, for ViT_Base_16 and ViT_Large_16 models, the original images are resized to 224 × 224 pixels, while for the ViT_Huge_14 model, the images are resized to 518 × 518 pixels.

Table 1. Summary of the original dataset in terms of sample counts.

Category	Original Dataset
Unripe	339
Ripened	483
Over ripened without internal defects	107
Over ripened with internal defects	50
Total	979

The experimental setup for this study was conducted on Google Colab with NVIDIA-SMI 525.105.17, driver version 525.105.17, CUDA version 12.0, and GPU Tesla T4. The environment was configured with Python version 3.10.12, Torch version 2.0.1, and TensorFlow GPU 2.0.0.

2.2 Data Augmentation

Data augmentation (DA) involves expanding the original training set by applying label-preserving transformations, which can be represented as the mapping:

$$\phi : S \to A$$

Here, S denotes the original training set, and A represents the augmented set derived from S. The augmented training set is then defined as:

$$S^1 = S \cup A$$

where S^1 comprises, the original training set along with the corresponding transformations defined by ϕ. Our goal is to ensure that the augmented images maintain sufficient distinction from the originals while faithfully representing the same visual concept.

For each input image x in the original training set S, the five different images generated through transformations can be mathematically represented as:

$A(x) = \{\mathrm{rotate}(x)_{(5,10),}\ \mathrm{flip}(x)_{(LR)}, \mathrm{flip}(x)_{(TB)}, \mathrm{distort}(x)_{(4,4,8)}, \mathrm{brightness}(x)_{(0.3\text{–}1.2)}\}.$

where:

- $\mathrm{rotate}(x)_{(5,10)}$ denote images rotated by max left rotation of 5 and max right rotation of 10 degrees.
- $\mathrm{flip}(x)_{(LR)}$ and $\mathrm{flip}(x)_{(TB)}$ represent images flipped horizontally and vertically.
- distort(x) $_{(4,4,8)}$ denotes an image distorted with grid width 4, grid height 4, and magnitude 8,
- brightness(x) $_{(0.3\text{–}1.2)}$ denotes images with varying brightness factors ranging from 0.3 to 1.2.

Thus, the augmented set A for each image x in S is A(x), resulting in an augmented training set $S^1 = S \cup A$. Figure 1 depicts the pictorial representation of data augmentation for generating newly sampled mango images.

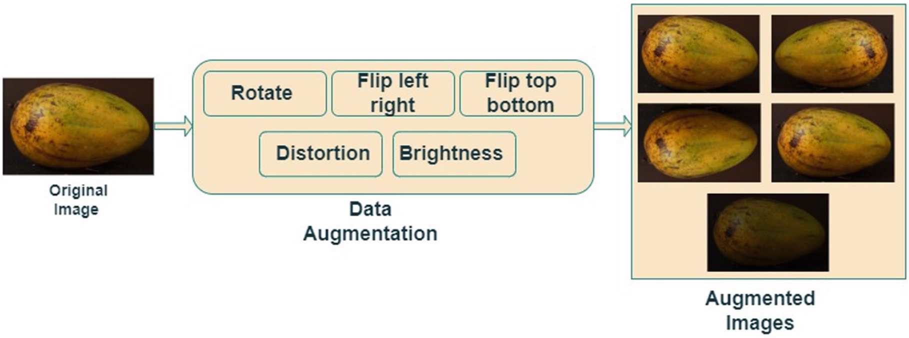


Fig. 1. Pictorial representation of data augmentation for generating newly sampled mango images.

2.3 Proposed Methodology

The proposed model, based on vision transformers, categorizes the input mango image into four classes: unripe, ripened, over-ripe without internal defects, and over-ripe with internal defects, as illustrated in Fig. 2. In this study, various configurations of vision transformer models are investigated both with and without pre-trained weights. When pre-trained weights are used, specific model weights are initialized accordingly. For instance, the ViT_Base model with a patch size of 16 would typically be trained from scratch. In addition to that, we opted to utilize pre-trained weights derived from ViT_Base_16 as well. Moreover, this research addresses challenges related to small datasets and constrained computational resources by leveraging vision transformers (ViTs) trained in a self-supervised manner.

Primarily, the proposed model divides the mango image into patches, with each patch typically being a square region of the image. For instance, if an image measures 518 x 518 pixels and the patch size is 14 x 14 pixels, the model divides both the height and width of the image to yield 1369 patches, as outlined in Eq. (1).

$$Number\ of\ patches(N) = (H * W) \div (ph * pw) \tag{1}$$

where H and W denote the height and width of the input image respectively, and ph and pw represent the patch height and patch width respectively. The stride, which is the number of pixels that the sliding window moves each time, is consistently set to 16.

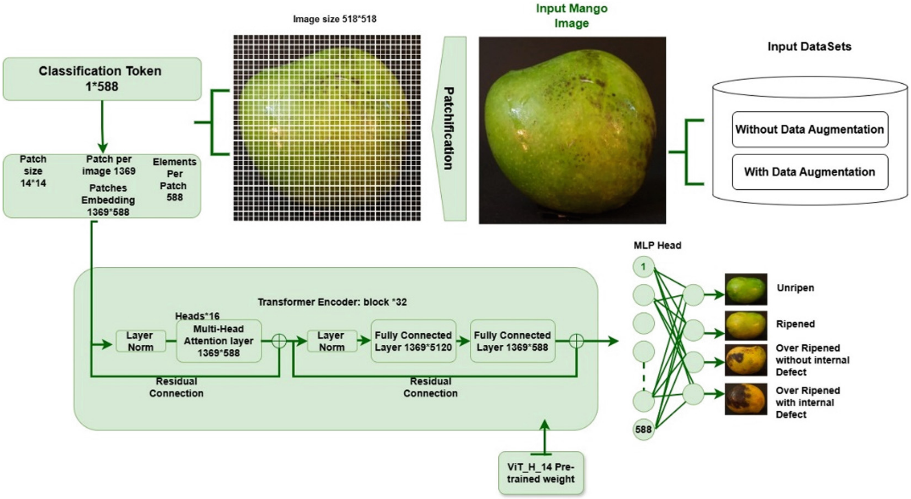


Image size 518*518
Input Mango Image
Input DataSets
Without Data Augmentation
With Data Augmentation
Patchification
Classification Token 1*588
Patch size 14*14
Patch per image 1369
Patches Embedding 1369*588
Elements Per Patch 588
Transformer Encoder: block *32
Heads*16
Layer Norm
Multi-Head Attention layer 1369*588
Layer Norm
Fully Connected Layer 1369*5120
Fully Connected Layer 1369*588
Residual Connection
Residual Connection
ViT_H_14 Pre-trained weight
MLP Head
1
588
Unripen
Ripened
Over Ripened without internal Defect
Over Ripened with internal Defect

Fig. 2. Architectural diagram of the proposed model.

In the proposed model, an image of size 518 × 518 is divided by the patch size of 14 × 14 to get 1369 number of patches. These patches are flattened from 2D to 1D vectors of size 588. Each patch vector undergoes linear transformation to generate patch embeddings. Position embeddings are then added to these embeddings to preserve the spatial sequence information of the patches. Finally, a learnable class embedding is prepended to these embeddings to facilitate image classification.

The above-mentioned task of dividing the image into patches is accomplished with the aid of the Conv2D layer; Transformer-based models have shown promising capabilities in fitting data, but there is increasing evidence suggesting they may face issues with overfitting, particularly when training data is limited (Chen et al., 2021). To tackle this challenge, we have explored integrating convolutional neural network (CNN) components into the Vision Transformer architecture. One such method involves employing Conv2D layers for patch identification within the Vision Transformer model. This strategy aims to leverage the strong spatial feature extraction capabilities of convolutional layers to improve the model's performance in vision tasks. Indeed, research indicates that using Conv2D for patch identification can significantly enhance model performance (Khan et al., 2023). Moreover, combining Transformer and convolutional elements allows for leveraging the strengths of both architectures: Transformers excel in efficient global feature extraction, while convolutional layers excel in handling local spatial features. This synergy enables the model to better capture spatial relationships within input images, leading to improved feature representation and, consequently, higher classification accuracy.

Patches are analogous to filter (kernel) size. In our context, patches and filters (feature maps) are equivalent, but we are specifying the feature map size as 14. To flatten these patches, we utilize a flatten layer, which is implemented in PyTorch as part of our architecture where we invoke and execute the task.

All these linearly projected and flattened patches, position embeddings, and class tokens are fed into the Transformer Encoder Block. The initial layer in the transformer encoder is LayerNorm (LN), which normalizes an input across its last dimension. LN aids in enhancing training efficiency and the generalization ability of the model by stabilizing the process and mitigating input variances. Following LN, the multihead self-attention (MSA) layer is constructed with specific parameters: embedding dimension of 1280 (hidden size D), utilizing 16 attention heads (hence the term "multi-head"), and dropout set to 0.

The primary role of the multihead self-attention (MSA) mechanism is to enable each patch to attend to and gather information from other patches. It captures interdependencies between patches and facilitates the model in considering the overall context of the image. MSA directs attention to different patches within the image, comprehending their relationships to capture critical information for further processing. Initially, MSA assigns three fundamental roles to each patch: Query (Q), Key (K), and Value (V). The Query (Q) represents the patch searching for other patches to attend to, while the Key (K) represents the patch being examined by others. The Value (V) holds the information or significance of the patch. MSA examines each patch and pairs it with other patches in the image, assessing their relationship by measuring the similarity between each patch's Query vector and the Key vectors of all other patches. Attention weights are computed using a softmax operation (described in Eq. 2), which determines how much attention each patch should allocate to other patches in the image. The higher the similarity score, the more closely related the patches are. This process helps MSA understand patch relationships and capture crucial information, thereby enabling the model to make more effective predictions.

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

where, d_k is the dimension of the key vector. After the multihead self-attention (MSA) layer, we employ an "Add" operation in the transformer encoder. This operation, also known as residual connections or skip connections, performs element-wise addition between the output of the previous layer and the output of the attention/feed-forward sub-layers. "Add" preserves the original information from the preceding layer while incorporating new information learned by the sub-layers. This addition also establishes shortcut paths for information flow, thereby facilitating efficient propagation of gradients during training. This mechanism effectively mitigates the vanishing gradient problem and enhances the model's learning capabilities. By combining "Add" with LayerNorm (Norm), each transformer layer promotes improved information flow, gradient propagation, and overall stability during training.

Then, MLP (multilayer perceptron) is created with three parameters: embedding dimension of 1280 (hidden size), MLP size of 5120 and dropout of 0.1 is applied after every dense layer. Each patch's output is processed through feed-forward network (FFN). This helps to capture the complex nonlinear relationship within the patches. Following this, the model has classification head which maps the output of the transformer into the desired output format.

3 Results and Discussion

In this subsection, the proposed model has been set up to function at its maximum performance through an extensive experimentation. For this purpose, the dataset has been divided into a training subset and a testing subset with 60:40 ratios empirically. Table 3 provides a detailed presentation of the average accuracies computed for all three ViT models, each with four distinct variants, across 20 trails. For each trail, these ViT models were trained over 30 epochs. Additionally, Table 3 includes the standard deviation (±SD) of the accuracies from the 20 trails for each model, offering insights into the variability and dispersion of the accuracy values around the mean. This allows for more comprehensive understanding of the model's performance consistency across different runs. In order to improve the classification performance, we have explored five data augmentation strategies to synthetically induce additional data for training and separated our dataset as; without data augmentation dataset consists of 979 mango images and with data augmentation mango dataset consists of 1566 alphonso mango images belonging to four classes specifically, unripe, ripened, over-ripened without internal defects and over-ripened with internal defects. In this study we have adopted three ViT models; and the parameters of each model is described in Table 2.

Table.2 Shows parameters specification of all three ViT models.

ViT Models	Layers	Hidden Size D	MLP Size	Heads	Params
ViT_Base_16	12	768	3072	12	86M
ViT_Large_16	24	1024	4096	16	307M
ViT_Huge_14	32	1280	5120	16	632M

Through this experimentation we could see that even though increasing the dataset from 979 images to 1566 images the accuracy is not much increased; for example, the accuracy of ViT_L_16 without data augmentation achieves 48.50%. Then if we train the same model with data augmentation, the model achieves 48.89% there is no much improvement because unlike CNNs, which can benefit from data augmentation techniques such as cropping, flipping, and rotating, vision transformers are more sensitive to the order and position of the patches.

Table.3 Accuracy obtained by all three models considering data augmentation and pre trained weights.

Vision Transformer Models	Without pre-trained weights				With pre-trained weights			
	Without Data augmentation		With Data augmentation		Without Data augmentation		With Data augmentation	
	Accuracy	± SD	Accuracy	± SD	Accuracy	± SD	Accuracy	± SD
ViT_B_16	46.6%	± 0.57	47.8%	± 0.62	89.9%	± 0.51	91.11%	± 0.49
ViT_L_16	48.50%	± 0.72	48.89%	± 0.58	91.30%	± 0.53	91.41%	± 0.59
ViT_H_14	48.72%	± 0.59	49.01%	± 0.61	92.07%	± 0.47	**92.78%**	± 0.43

The position encoding schema of proposed model plays a crucial role in its effectiveness for mango ripeness grading by facilitating accurate and context-aware feature extraction from images. This position encoding scheme has several advantages for mango ripeness grading which are listed below:

1. **Spatial Awareness**: Mangoes, exhibit distinct visual changes as they ripen, including alterations in color, texture, and size. The position embeddings in a vision transformer help the model understand the spatial relationships between different parts of the mango in the image. This spatial awareness allows the model to distinguish between relevant features such as the color changes in different areas of the mango (e.g., from green to yellow or orange) or variations in texture (e.g., smoothness to softness).
2. **Contextual Understanding**: Mango ripeness assessment requires the model to analyze the overall appearance of the mango fruit, taking into account how different parts contribute to its ripeness level. Vision transformers use position embeddings to capture both local and global context within the image. This is crucial because ripeness is often determined by a combination of visual cues spread across the entire mango rather than isolated features.
3. **Handling Different Mango Sizes and Orientations**: Mangoes can vary significantly in size and shape, and they may be presented in images with different orientations (e.g., top-down view or side view). The position embeddings allow the model to understand the spatial arrangement of pixels in the image regardless of orientation or scale. This adaptability ensures that the model can effectively learn and generalize features relevant to ripeness grading across mango images.
4. **Enhanced Feature Learning**: By incorporating position embeddings, the vision transformer can effectively learn fine-grained features that are critical for ripeness grading. For example, it can focus on specific parts of the mango that typically exhibit ripeness indicators, such as the blush on the skin or changes in texture around the stem.

One of the main challenges of training vision transformers is that, they require a lot of data to achieve good performance. Therefore, they need to see a large variety of images to learn meaningful representations. One way to address this issue is to use pre-trained models that have been trained on large-scale datasets, such as ImageNet, and then fine-tune them on the target task or domain. This can significantly reduce the training time and improve the accuracy of the vision transformer. Hence in this study initially, all the ViT models are trained from scratch and it has been observed that the accuracy was below 50%. To address this issue, this study uses the pre-trained weights of respective models that are trained on ImageNet dataset. So instead of network learns the weights during training the pre-trained weights are directly passed to model, and then it has been observed that each ViT models' performance is increased. For example, the accuracy of the ViT_H_14 model trained from the scratch is 49.01%, and then the same ViT_H_14 with vit_h_14 pre-trained weights achieve accuracy of 92.78%. Here the performance is increased by 43.77%. Hence, use of pretrained weights in proposed model significantly influence model performance and efficiency for mango ripeness grading through its transfer learning benefits and improved generalization. Overall, the novel aspect of leveraging a pre-trained model and fine-tunning it on a custom dataset lies in its ability to combine the strengths of large-scale pre-training with the specificity required for particular tasks, resulting in the models that are both efficient and highly effective in addressing diverse challenges in the application domain.

Table 4. Comparison of the proposed model with other deep learning model on the same dataset

Models	Proposed	CNN with 3 layers	Inception-V3		VGG-16		EffientNetB0		ResNet 50	
			WFT	FT	WFT	FT	WFT	FT	WFT	FT
Accuracy	**92.78**	85.1	83.4	85.4	81.0	82.2	86.2	90.4	80.3	81.6

Table 5. Comparison of the proposed model with existing models

Models	Classification Method adopted	Number of classes	Accuracy
Raghavendra et al., 2020(Existing)	Conventional Machine Learning	4	87.04%
Mansor and Othman, 2014	Fuzzy Logic	3	87%
Nandi et al., 2014	Gaussian Mixture model	4	Less than 90% in all varieties
Salunkhe and Anikert, 2015	Rule based classification	4	84.2%
Proposed	**Vision Transformers with pretrained weights**	**4**	**92.78%**

The detailed accuracies obtained by each ViT models are described in Table 3. Based on the computed accuracies, it has been observed that ViT_H_14 model with vit_h_14 pre-trained weights attained highest accuracy over rest of the ViT models. Table 4 shows the comparison of the proposed model with the other deep learning models. In table 4, for other models we have given the accuracy obtained by the pre trained architecture of the respective model with (FT) and without fine tuning (WFT). Overall, proposed model success in mango ripeness grading can be attributed to its innovative use of attention mechanisms, effective representation learning, and adaptability to different tasks through fine-tuning and customization. These factors collectively contribute to its superior performance compared to traditional CNN-based architectures. Figure 3 shows the loss and accuracy curve of the proposed model. From this curve we can see, our proposed model is learning effectively and generalizing well to unseen data. Table 5 shows the comparison of the proposed model with other existing models; it has been shown that our model out performs the existing model (Raghavendra et al., 2020) by 6.67% with the same dataset. Here, the accuracy shown against the rest of the models are the accuracies quoted in their respective works.

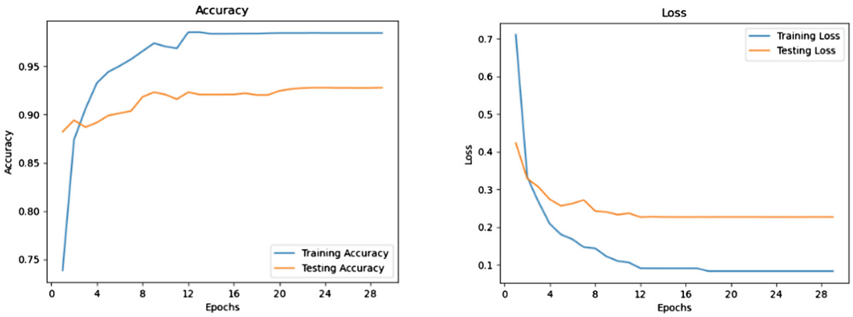


Fig. 3. Loss and Accuracy curve of the ViT_H_14 with pre-trained weights.

4 Conclusion

In this study, a successful attempt is made to explore the applicability of vision transformer for grading of mangoes. The empirical analysis conducted in this study argues that, contrary to initial expectations regarding the benefits of data augmentation for improving model performance, the Vision Transformer (ViT) does not significantly benefit from techniques such as flipping and rotating. Unlike Convolutional Neural Networks (CNNs), which can exploit these data augmentation methods effectively, ViTs are highly sensitive to the order and position of patches. Therefore, traditional data augmentation techniques do not provide substantial improvements for ViTs as they do for CNNs. Typically, achieving state-of-the-art results with Vision Transformers (ViTs) requires a substantial volume of data. These issues are addressed by using pre-trained weights and we could also see an improved performance. Compared to all three models ViT_H_14 performed better with average accuracy of 92.78%. The significance of the different vision transformer models with pre-trained weights for effective prediction of four different classes of Alphonso mango variety was brought out. Future work is on exploring the reason for still 7.22% of error in spite of using the most effective and best model available in the literature and recommending suitable modification to achieve 100% results. Further, it is planned to have a real time deployable solution for the same. Through this experimentation it is also observed that by decreasing the patch size and increasing the number of transformer layer might increase the performance of the model. So, this can also be taken up for the further experimentation.

Acknowledgments. The authors would like to thank Ministry of Social Justice and Empowerment, Government of India for the financial support. The authors also wish to acknowledge Dr.Anitha Raghavendra for her generous provision of the dataset used in this study.

References

Amruta, S., & Wakode, M. (2021) "Ripeness, Size and Shape based Automated Mango Grading using Image Processing and Machine Learning Techniques" International Journal of current engineering and technology INPRESSCO: E-ISSN 2277–4106.

Agilandeeswari, L., Prabukumar, M., & Goel, S. (2017). Automatic grading system for mangoes using multiclass SVM classifier. International Journal of Pure and Applied Mathematics, 116(23), 515-523.

Bakar, M. A., Abdullah, A. H., Rahim, N. A., Yazid, H., Saad, F. S. A., & Ahmad, K. (2020, September). Development of ripeness indicator for quality assessment of harumanis mango by using image processing technique. In IOP Conference Series: Materials Science and Engineering (Vol. 932, No. 1, p. 012087). IOP Publishing.

Bhole, V., & Kumar, A. (2020, October). Mango quality grading using deep learning technique: Perspectives from agriculture and food industry. In Proceedings of the 21st annual conference on information technology education (pp. 180–186).

Bazi, Y., Bashmal, L., Rahhal, M. M. A., Dayil, R. A., & Ajlan, N. A. (2021). Vision transformers for remote sensing image classification. Remote Sensing, 13(3), 516.

Chen, Z., Xie, L., Niu, J., Liu, X., Wei, L., & Tian, Q. (2021). Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 589–598

Chhabra, M., Gupta, A., Mehrotra, P., & Reel, S. (2012). Automated detection of fully and partially riped mango by machine vision. In Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20–22, 2011: Volume 2 (pp. 153–164). Springer India.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dhanya, V. G., Subeesh, A., Kushwaha, N. L., Vishwakarma, D. K., Kumar, T. N., Ritika, G., & Singh, A. N. (2022). Deep learning based computer vision approaches for smart agricultural applications. Artificial Intelligence in Agriculture.

Gururaj, N., Vinod, V., & Vijayakumar, K. (2022). Deep grading of mangoes using Convolutional Neural Network and Computer Vision. Multimedia Tools and Applications, 1–26.

Khan, A., Hassan, T., Shafay, M., Fahmy, I., Werghi, N., Seneviratne, L., & Hussain, I. (2023). Convolutional Transformer for Autonomous Recognition and Grading of Tomatoes Under Various Lighting, Occlusion, and Ripeness Conditions. *arXiv preprint* arXiv:2307.01530.

Khan, A., Hassan, T., Shafay, M., Fahmy, I., Werghi, N., Mudigansalage, S., & Hussain, I. (2023). Tomato maturity recognition with convolutional transformers. Scientific Reports, 13(1), 22885.

Khoje, S., & Bodhe, S. (2012). Performance comparison of Fourier transform and its derivatives as shape descriptors for mango grading. International Journal of Computer Applications, 53(3), 17-22.

Khoje, Suchitra, Bodhe, S.K., 2015. Comparative performance evaluation of fast discrete curvelet transform and color texture moments as texture features for fruit skin damage detection. Springer J. Food Sci. Technol. 52, 6914–6926.

Knott, M., Perez-Cruz, F., & Defraeye, T. (2023). Facilitated machine learning for image-based fruit quality assessment. Journal of Food Engineering, 345, 111401.

Nandi, C. S., Tudu, B., & Koley, C. (2014). Machine vision based techniques for automatic mango fruit sorting and grading based on maturity level and size. Sensing technology: current status and future trends II, 27-46

Nayeli, V. Rivera, José, J. Chanona Pérez, Reynold, F. Rebollo, José, Blasco, Georgina, C. Domínguez, de María, J.P. Flores, Israel, A. Vázquez, (2012). "Description of maturity stages of mango 'Manila' by image analysis and ripening index" CIGR-Ageng Conference www2.atb-potsdam.

Othman, M., Bakar, M. N. A., Ahmad, K. A., & Razak, T. R. (2014). Fuzzy ripening mango index using RGB colour sensor model. Researchers World, 5(2), 1.

Parez, S., Dilshad, N., Alghamdi, N. S., Alanazi, T. M., & Lee, J. W. (2023). Visual intelligence in precision agriculture: Exploring plant disease detection via efficient vision transformers. Sensors, 23(15), 6949.

Rizzo, M., Marcuzzo, M., Zangari, A., Gasparetto, A., & Albarelli, A. (2023). Fruit ripeness classification: A survey. Artificial Intelligence in Agriculture.

Raghavendra, A., Guru, D. S., Rao, M. K., & Sumithra, R. (2020). Hierarchical approach for ripeness grading of mangoes. Artificial Intelligence in Agriculture, 4, 243-252.

Supekar, A., & Wakode, M. (2020). Computer vision based automated mango grading–a review. J. Postharvest Techno, 8(1), 23-37.

Sa'ad, F. S. A., Ibrahim, M. F., Shakaff, A. M., Zakaria, A., & Abdullah, M. Z. (2015). Shape and weight grading of mangoes using visible imaging. Computers and Electronics in Agriculture, 115, 51-56.

Salunkhe, R. P., & Patil, A. A. (2015, December). Image processing for mango ripening stage detection: RGB and HSV method. In 2015 Third International Conference on Image Information Processing (ICIIP) (pp. 362–365). IEEE.

Shahi, T. B., Sitaula, C., Neupane, A., & Guo, W. (2022). Fruit classification using attention-based MobileNetV2 for industrial applications. Plos one, 17(2), e0264586.

Thai, H. T., Le, K. H., & Nguyen, N. L. T. (2023). FormerLeaf: An efficient vision transformer for Cassava Leaf Disease detection. Computers and Electronics in Agriculture, 204, 107518.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing System, 30.

Vyas, A. M., Talati, B., & Naik, S. (2014). Quality inspection and classification of mangoes using color and size features. International Journal of Computer Applications, 98(1).

Wang, Y., Chen, Y., & Wang, D. (2022). Convolution network enlightened transformer for regional crop disease classification. Electronics, 11(19), 3174.

Xiao, B., Nguyen, M., & Yan, W. Q. (2023). Fruit ripeness identification using transformers. Applied Intelligence, 1–12.

Yu, S., Xie, L., & Huang, Q. (2023). Inception convolutional vision transformers for plant disease identification. Internet of Things, 21, 100650.

Zheng, B., & Huang, T. (2021). Mango grading system based on optimized Convolutional Neural Network. *Mathematical Problems in Engineering*, *2021*, 1-11.

FedRewind: Rewinding Continual Model Exchange for Decentralized Federated Learning

Luca Palazzo(✉), Matteo Pennisi, Federica Proietto Salanitri, Giovanni Bellitto, Simone Palazzo, and Concetto Spampinato

PeRCeiVe Lab, University of Catania, Catania, Italy
lpalazzo@gmail.com
http://www.perceivelab.com/

Abstract. In this paper, we present *FedRewind*, a novel approach to decentralized federated learning that leverages model exchange among nodes to address the issue of data distribution shift. Drawing inspiration from continual learning (CL) principles and cognitive neuroscience theories for memory retention, *FedRewind* implements a decentralized routing mechanism where nodes send/receive models to/from other nodes in the federation to address spatial distribution challenges inherent in distributed learning (FL). During local training, federation nodes periodically send their models back (i.e., *rewind*) to the nodes they received them from for a limited number of iterations. This strategy reduces the distribution shift between nodes' data, leading to enhanced learning and generalization performance. We evaluate our method on multiple benchmarks, demonstrating its superiority over standard decentralized federated learning methods and those enforcing specific routing schemes within the federation. Furthermore, the combination of federated and continual learning concepts enables our method to tackle the more challenging federated continual learning task, with data shifts over both space and time, surpassing existing baselines.

Keywords: Decentralized Learning · Continual Learning · Federated Learning

1 Introduction

The proliferation of data across multiple distributed devices and locations has sparked significant interest in federated learning (FL), a paradigm that enables collaborative model training without the need to centralize data. Federated learning offers numerous benefits, including enhanced privacy and reduced communication costs. However, a fundamental challenge in FL is the non-i.i.d. (independent and identically distributed) nature of data across different nodes, which

L. Palazzo and M. Pennisi—Equal contribution.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 79–94, 2025.
https://doi.org/10.1007/978-3-031-78389-0_6

can lead to performance degradation due to data distribution shifts. This problem becomes even more pronounced in decentralized federated learning, where there is no central server to coordinate and aggregate updates, making the system less robust to heterogeneous data distributions.

Existing solutions in federated learning primarily focus on mitigating the effects of non-i.i.d. distributions through various aggregation and optimization techniques. Centralized federated learning approaches often rely on a central server to aggregate updates from all nodes, thereby smoothing out the differences in local data distributions [16,21,28]. Decentralized methods, instead, employ peer-to-peer communication and model averaging strategies to achieve consensus without a central entity [2,5,31]. While these methods have shown promise, they often fall short in addressing the dynamic nature of data distribution shifts, particularly in environments featured by strong data imbalance [32,37].

Continual learning (CL) [6,20,22], a field that addresses the problem of learning from a stream of data that changes over time, offers valuable insights for handling distribution shifts with strong imbalance. CL methods are designed to prevent *catastrophic forgetting*, which occurs when a model forgets previously learned information upon encountering new data, by maintaining knowledge across sequential learning tasks through either exposing the model to limited past experience [1,4,23,24] or regularizing model parameters [3,13,36] using previous knowledge, while learning new tasks. Although CL and FL address similar challenges, they operate in different contexts: CL deals with non-i.i.d. data over time, while FL addresses non-i.i.d. data across distributed nodes.

We here propose *FedRewind*, a novel approach that integrates continual learning concepts into federated learning to address the limitations of existing FL methods. Our method involves decentralized nodes periodically exchanging their models and sending them back to the originating nodes for a limited number of iterations during local training. This exchange mechanism, inspired by continual learning strategies, aims to prevent overfitting on local data and enhance memory retention by periodically re-exposing models to previously seen data.

FedRewind's strategy also aligns with the cognitive neuroscience principle of *testing effect*, which emphasizes the role of active recall and retrieval practice for the enhancement of long-term memory. The testing effect, in particular, demonstrates that memory retrieval processes (similar to our rewind strategy) significantly improve knowledge retention compared to simple re-exposure to information [12,25]. This phenomenon is underpinned by mechanisms such as elaborative retrieval and spreading activation, where active recall strengthens memory traces and facilitates the integration of new information into existing cognitive frameworks.

By adapting cognitive neuroscience principles and continual learning concepts to the spatial distribution challenges of FL, *FedRewind* aims to reduce the distribution shift between nodes, thus enhancing model performance and robustness. We validate our claims on multiple benchmark datasets, demonstrating how *FedRewind* leads to performance improvement over standard decentralized federated learning methods, as well as those that impose specific routing schemes within the federation. Furthermore, the combination of federated and continual learning concepts enables our method to effectively address the federated continual learning problem, where data shifts occur over both space and time, outperforming existing baselines. Our results, cumulatively, indicate that this decentralized and iterative model exchange approach offers a robust solution to the challenges posed by non-i.i.d. data in federated learning environments.

2 Related Work

Federated learning (FL) has emerged as a new paradigm within distributed machine learning, addressing the challenge of data privacy. Drawing upon the foundational work of McMahan et al. [21], FL facilitates collaborative model training while ensuring node data remains secure on their local devices.

A typical FL setting features a central server that orchestrates the learning process. This server distributes a global model to a pool of participating nodes, which use their private data for local updates on the received model. Subsequently, the nodes transmit their local updates back to the central server, which aggregates them to refine the global model. This iterative process of distributing, updating, and aggregating the model persists until a satisfactory level of convergence is achieved.The most common aggregation technique is FedAvg [21], that simply averages the local model parameters received from all nodes. More sophisticated aggregation methods have been proposed by adding a regularization term[16] or leveraging knowledge distillation[39]. Another branch of FL, namely Personalized Federated Learning, has the primary objective to improve the performances w.r.t. only the single node distribution. FedBN[17] achieves this goal by preserving the batch-norm layers of each node while FedProto[29] aggregates only the prototypes while the models are kept on each node.

While a central server simplifies the communication protocols, especially for large-scale deployments, its presence introduces specific limitations. Firstly, it creates a single point of failure, posing a vulnerability to system robustness if the server becomes unavailable. Secondly, as the number of participating nodes increases, the central server can become a bottleneck, hindering communication efficiency [18]. Finally, the very presence of a central server that aggregates data might not be desirable or even feasible in certain collaborative learning scenarios. This is particularly true for scenarios that prioritize robust privacy guarantees or involve geographically dispersed participants with limited or unreliable network connectivity.

This work investigates also decentralized federated learning, which, conversely to centralized approaches, relies on peer-to-peer communication between nodes[5,11]. This approach eliminates the single point of failure and enhances privacy guarantees, but introduces additional complexity in terms of communication protocols and achieving convergence among local models on all devices.

Since *FedRewind* leverages concepts from **continual learning (CL)**, we provide a brief overview of existing methods related to the strategies we employ to retain knowledge from past learning rounds. Continual learning [6,22] is a field of machine learning that seeks to bridge the gap between the incremental learning observed in humans and the limitations of neural networks. McCloskey and Cohen [20] identified the phenomenon of “catastrophic forgetting”, where neural networks lose previously acquired knowledge upon encountering substantial shifts in the input distribution.

To address catastrophic forgetting, various mitigation strategies have been proposed. These include the introduction of appropriate regularization terms [13,36], the development of specialized network architectures [19,26], and the use of rehearsal mechanisms that leverage a limited set of previously encountered data points [4,23,24].

FedRewind adopts a hybrid approach, combining elements of both regularization and rehearsal strategies. Unlike traditional methods, it does not use any buffer. Instead, during training rounds on local nodes, the model is periodically sent back to previous nodes for regularization. This process helps address data shifts across nodes, thereby mitigating potential forgetting.

Federated continual learning (FCL) combines the paradigms of federated learning (FL) and continual learning (CL), enabling it to address the challenge of distributed data that, at the same time, undergo continual change over time [9,33,34]. However, initial efforts in this area compromised data privacy by requiring the storage of training samples on the central server [33]. Recent advancements in FCL prioritize data privacy by advocating for the storage of only perturbed images for replay purposes [9]. FedWEIT [34], instead, tackled the problem by decomposing network weights but necessitates data replay buffers. Recently, FedSpace [27] has been developed to overcome the limitations of current methods. It leverages class prototypes within the feature space and employs contrastive learning to preserve prior knowledge and reduce divergence between the behaviors of different federation nodes.

Our *FedRewind* strategy is complementary to approaches like FedSpace, enhancing their capabilities by mitigating typical overfitting in distributed learning, particularly in cases of high class imbalance and strongly non-iid data distribution among nodes. Specifically, *FedRewind* addresses these issues by transferring models between nodes to enforce iid-ness on data, rather than relying on data storage. This method significantly reduces privacy concerns associated with storing sensitive training data. By periodically sending the model back to previous nodes, we maintain knowledge across sequential tasks and enforce regularization, thus reducing node overfitting, while enhancing both privacy (no need for data sharing) and scalability.

3 Method

In federated learning, a collection of nodes collaboratively train a shared model while keeping their data localized. This approach ensures data privacy but introduces challenges related to effective knowledge sharing across distributed

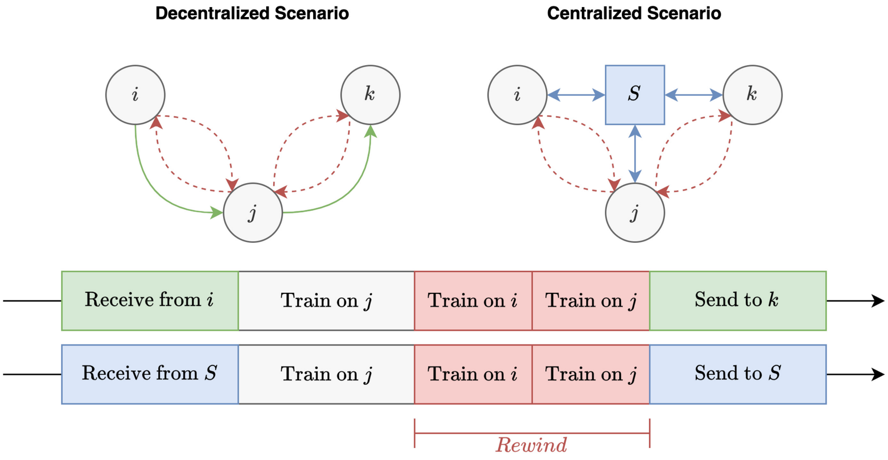


Fig. 1. Rewind strategy. The model received and trained on the current node is sent back to its source node for a brief fine-tuning. The model then returns to the node and continue its training before the start of a new federated round.

and sequential learning tasks. To address these challenges, we propose a novel "rewind" strategy. This section introduces the approach and key concepts, describes the rewind method in detail, and provides the corresponding pseudocode.

Federated Learning. Federated learning is a collaborative machine learning approach where multiple nodes (N nodes) train a shared model without centralizing their data. Each node updates its model using local data and shares the model updates rather than the data itself, ensuring data privacy.

Decentralized Federated Learning. In this framework, nodes communicate directly with each other without a central server. We consider two modes of communication:

- **Random Communication (RWT)**: Nodes select a random source node (for the incoming model) and a random destination node (for the outcoming model) for information exchange in each round.
- **Cyclic Communication (CWT)**: Each node communicates with the same predetermined source and destination nodes in every round, as described in [5].

Centralized Federated Learning. In this framework, a central server coordinates the training process. Nodes send their local model updates to the server, which aggregates them to form a global model.

Training Rounds. Defined as a block of training where all nodes complete training for E epochs. At the end of each round, model exchange across all federated nodes is carried out.

3.1 The Rewind Strategy

To improve knowledge sharing in federated learning, we introduce the "rewind" strategy, described in Fig. 1. This method involves temporarily reverting the model to a previous node to rehearse prior knowledge, thus preserving data privacy. We apply this strategy on both centralized and decentralized federated learning.

Decentralized federated learning. During each communication round, a generic node C_j receives a model M_i, parameterized by θ_i, from a source node C_i, trains it on its local dataset D_j, and forwards it to another node C_k. The standard training process on node C_j for model M_j parameterized by θ_j at round t on dataset D_j is given by:

$$M_j^{(t)} = \mathrm{Train}_{D_j,E}(M_i^{(t-1)}) = \theta_i^{(t-1)} - \eta \sum_{0}^{E-1} \nabla L(\theta_i^{(t-1)}, D_j) \tag{1}$$

where E denotes the number of epochs for a single federation round, and L is a generic loss function.

To enhance knowledge retention, we introduce a fractional computation budget parameterized by λ for retraining the model on its origin node before continuing training on the current node. This modifies the training equation as follows:

$$M_j^{(t)} = \mathrm{Train}_{D_j,\lambda\cdot E}\left(\mathrm{Train}_{D_i,\lambda\cdot E}\left(\mathrm{Train}_{D_j,(1-2\lambda)\cdot E}\left(M_i^{(t-1)}\right)\right)\right) \tag{2}$$

where λ denote the fraction of the budget allocated for rewinding.

Centralized Federated Learning. In this scenario, a central server S aggregates models received from nodes at each communication round. The model computed by the server $M_s^{(t)}$ at round t is defined as:

$$M_s^{(t)} = agg(\{M_j^{(t)} \mid j \in \{1, 2, \ldots, N\}\}) \tag{3}$$

where $agg(\cdot)$ represents a generic aggregation function employed by the server. Applying the rewind strategy, the training process for a generic node C_j is modified to:

$$M_j^{(t)} = \mathrm{Train}_{D_j,\lambda\cdot E}\left(\mathrm{Train}_{D_i,\lambda\cdot E}\left(\mathrm{Train}_{D_j,(1-2\lambda)\cdot E}\left(M_s^{(t-1)}\right)\right)\right) \tag{4}$$

By leveraging inter-node communication and the rewind strategy, federated learning-whether decentralized or centralized-can more effectively retain knowledge across different tasks. This approach ensures that the shared model benefits from the distributed data while maintaining privacy and improving performance.

The pseudo-code of the rewind strategy is reported in Algorithm 1 and Algorithm 2 respectively for the decentralized and centralized scenario.

Algorithm 1: Decentralized Federated Learning with Rewind Strategy.

```
Input: $N$ nodes, initial model $M_0$, epochs $E$, fractional budget $\lambda$
for each round $t$ in 1 to $T$ do
    for each node $C_j$ in $N$ do
        // Receive model from source node $C_i$
        $M_i^{(t-1)} \leftarrow \text{receive\_model}(C_i)$;
        // Train on current node's dataset $D_j$
        $M_{\text{intermediate}} \leftarrow \text{Train}(M_{\text{rewind}}, D_j, (1-2\lambda) \cdot E)$;
        // Rewind phase: Train on source node's dataset $D_i$
        $M_{\text{rewind}} \leftarrow \text{Train}(M_i^{(t-1)}, D_i, \lambda \cdot E)$;
        // Finish training on current node's dataset $D_j$
        $M_j^{(t)} \leftarrow \text{Train}(M_{\text{intermediate}}, D_j, \lambda \cdot E)$;
        // Send model to destination node $C_k$
        $\text{send\_model}(C_j, M_j^{(t)})$;
    end
end
```

Algorithm 2: Centralized Federated Learning with Rewind Strategy.

```
Input: $N$ nodes, initial model $M_0$, epochs $E$, fractional budget $\lambda$
for each round $t$ in 1 to $T$ do
    for each node $C_j$ in $N$ do
        // Receive aggregated model from the server
        $M_s^{(t-1)} \leftarrow \text{receive\_model}(\text{server})$;
        // Train on current node's dataset $D_j$
        $M_{\text{intermediate}} \leftarrow \text{Train}(M_{\text{rewind}}, D_j, (1-2\lambda) \cdot E)$;
        // Rewind phase: Train on previous node's dataset $D_i$
        $M_{\text{rewind}} \leftarrow \text{Train}(M_s^{(t-1)}, D_i, \lambda \cdot E)$;
        // Finish training on current node's dataset $D_j$
        $M_j^{(t)} \leftarrow \text{Train}(M_{\text{intermediate}}, D_j, \lambda \cdot E)$;
        // Send model to the server
        $\text{send\_model}(\text{server}, M_j^{(t)})$;
    end
    // Server aggregates models from all nodes
    $M_s^{(t)} \leftarrow \text{aggregate\_models}(\{M_j^{(t)} \mid j \in 1 \text{ to } N\})$;
end
```

4 Results

4.1 Federated Learning Performance

Experimental settings. To evaluate the effectiveness of *FedRewind*, we simulate different federated learning scenarios using three benchmarks (generally employed for testing FL methods), namely MNIST [8], CIFAR10 [14] and CIFAR100 [14]. Data is distributed across nodes according to a non-independent and identically distributed (non-IID) scheme. This distribution is achieved by applying a Dirichlet distribution, as in previous work [15,30,35], parameterized by α_{dir}, which serves as a measure of the degree of data heterogeneity across nodes; a lower α_{dir} value indicates a more pronounced imbalance in data distribution across nodes.

Our experimental settings include 50 communication rounds and two configurations based on the number of nodes in the federation: one with 10 nodes and another with 50 nodes. During each round, in each node, we perform local training for 10 epochs (E). For the rewind experiments, we set the rewind hyperparameter $\lambda = 0.1$. This configuration determines a training procedure where, within the given E=10 epochs, 8 epochs are dedicated to the local training on the current node's data, followed by one epoch on the previous node's data, and concluding with a final epoch on the current node's data. All experiments are carried out using the ResNet18 architecture [10], pre-trained on ImageNet [7], optimized using Stochastic Gradient Descent (SGD) with a learning rate of 0.001.

Metrics. In decentralized federated learning (FL), each node creates a distinct local model, unlike in the centralized FL paradigm, which results in a single global model at the end of the training phase. To quantify the aggregated performance and generalization capability of the federation, we propose the *Federation Accuracy (FA)* metric. This metric is calculated by testing all the node models within the federation against all the private test sets and then computing the mean accuracy. Given a federation of size N, our metric is defined as follows:

$$\text{FA} := \frac{1}{N \times N} \sum_{i=1}^{N} \sum_{j=1}^{N} \text{Acc}(M_i, D_j^{test}) \tag{5}$$

where $\text{Acc}(M_i, S_j)$ is the accuracy of model M_i on the private test set D_j^{test} of node j. Similarly we define the *Federation Fairness (FF)* that measures how much the performance of the nodes changes across the federation (e.g. the standard deviation of the accuracy of nodes):

$$\text{FF} := \sqrt{\frac{1}{(N \times N) - 1} \sum_{i=1}^{N} \sum_{j=1}^{N} (\text{Acc}(M_i, D_j^{test}) - \text{FA})^2} \tag{6}$$

Moreover, our objective is not only to improve the overall generalization across the federation, but also to enhance performance of each individual node on its private dataset. To measure this performance, we define the *Personalized Federation Accuracy (PFA)* metric as:

$$\text{PFA} := \frac{1}{N}\sum_{i=1}^{N} \text{Acc}(M_i, D_i^{test}). \tag{7}$$

These three metrics, PA, FF and PFA, allow us to capture both the generalization capabilities of the entire federation and the performance improvements from the perspective of individual nodes.

Baselines. We test our approach in combination to existing FL strategies, applying it to CWT [5], RWT, and FedAvg [21]. CWT employs a static cyclic model transfer between rounds, while RWT is our modified version of CWT, featuring random communication between nodes in each round.

We also assess our approach in two reference scenarios: the *Joint* and *Standalone* settings. The *Joint* setting represents an optimal condition where all data from the federation is consolidated and utilized for training on a single node, thereby establishing an upper bound on performance.

In contrast, the *Standalone* setting assumes that each node trains its model independently, with no communication or data sharing between nodes. This setting generally sets a lower bound on performance, particularly when the data distribution between nodes is highly non-IID.

Results. We begin our evaluation by testing FedRewind performance on the two scenarios: with 10 nodes and with 50 nodes, both under a strongly non-IID scenario with $\alpha_{dir} = 0.25$. Table 1 presents the results in terms of Federation Accuracy (FA) across the three benchmarks, showing that the *rewind* strategy consistently achieves higher accuracy. This improvement is especially pronounced in the 50-node scenario, which poses greater complexity and challenge due to its larger and more heterogeneous node distribution. Similarly, Table 2 presents the results in terms of Federation Fairness (FF), demonstrating that the implementation of the rewind strategy consistently reduces the standard deviation of the accuracy of nodes, thus enhancing the generalization capabilities of the federation.

The Personalized Federation Accuracy (PFA) results, shown in Table 3, further demonstrate the benefits of our rewind strategy at the node level. The strategy's effectiveness is evident, as it consistently enhances PFA across various datasets and federation scales. We speculate that the rewind mechanism acts as an effective regularizer, mitigating overfitting to a node's local dataset. In a non-IID setting, where the tendency to overfit to local data patterns is high, this regularization effect is crucial. By periodically rewinding and retraining with data from other nodes, the models are exposed to a more diverse data distribution, promoting more generalized representation learning. It is also noteworthy that the impact of our rewind strategy on personalization performance for FedAvg is relatively lower compared to CWT and RWT. This might be due to the aggregation step in FedAvg, which tends to smooth out the specificities of local models trained on non-IID data.

The results from the standalone setting highlight the limitations of training models in isolation, especially under non-IID conditions. As the number of nodes increases, standalone models generally perform poorly, struggling to learn representative features without the diversity of data from other nodes. This aligns with our expectations, as the standalone setting misses the collaborative benefits of federated learning.

Table 1. Federation Accuracy in a non-IID setting ($\alpha_{dir} = 0.25$) for the MNIST, CIFAR-10, and CIFAR-100 benchmarks, organized across 10 and 50 nodes.

Method	**10 Nodes**			**50 Nodes**		
	MNIST	**C10**	**C100**	**MNIST**	**C10**	**C100**
Joint	99.22	78.53	53.13	99.22	78.53	53.13
Standalone	69.19	33.23	19.07	49.91	26.85	11.08
FedAVG	95.43	51.19	39.89	87.47	53.26	37.29
↪**Rewind**	**97.59**	**59.27**	**41.06**	**91.77**	**58.84**	**39.90**
CWT	93.09	45.81	34.02	88.27	40.00	24.90
↪**Rewind**	**96.19**	**55.57**	**37.44**	**92.79**	**47.83**	**27.51**
RWT	94.93	44.16	32.08	86.66	38.75	24.42
↪**Rewind**	**97.42**	**52.34**	**36.72**	**87.01**	**45.85**	**27.40**

Table 2. Federation Fairness in a non-IID setting ($\alpha_{dir} = 0.25$) for the MNIST, CIFAR-10, and CIFAR-100 benchmarks, organized across 10 and 50 nodes.

Method	**10 Nodes**			**50 Nodes**		
	MNIST	**C10**	**C100**	**MNIST**	**C10**	**C100**
Joint	N/A	N/A	N/A	N/A	N/A	N/A
Standalone	27.02	25.47	12.15	28.97	20.59	11.08
FedAVG	3.81	13.37	10.33	13.18	11.21	5.33
↪**Rewind**	**1.50**	**12.75**	**10.09**	**9.03**	**10.10**	**5.16**
CWT	8.52	24.21	11.22	14.18	21.36	6.85
↪**Rewind**	**4.78**	**19.53**	**10.07**	**8.48**	**18.84**	**6.72**
RWT	4.27	24.95	10.84	17.54	21.26	6.75
↪**Rewind**	**1.82**	**21.98**	**10.06**	**14.68**	**19.01**	**6.69**

We further evaluated the robustness of our rewind strategy under varying degrees of data heterogeneity. Using the CIFAR-10 dataset, we measured federation accuracy with the Dirichlet coefficient (α_{dir}) ranging from 0.1, indicating

Table 3. Personalized Federation Accuracy in a non-IID setting ($\alpha_{dir} = 0.25$) for the MNIST, CIFAR-10, and CIFAR-100 benchmarks, organized across 10 and 50 nodes.

Method	**10 Nodes**			**50 Nodes**		
	MNIST	**C10**	**C100**	**MNIST**	**C10**	**C100**
Joint	99.22	78.53	53.13	99.22	78.53	53.13
Standalone	98.69	83.96	53.49	96.20	77.65	41.73
FedAVG	97.09	**56.15**	37.94	83.83	45.88	32.54
↪**Rewind**	**97.38**	54.39	**38.71**	**85.79**	**47.99**	**35.62**
CWT	94.82	38.01	30.76	86.74	38.90	24.72
↪**Rewind**	**95.65**	**48.85**	**34.34**	**93.40**	**46.29**	**26.58**
RWT	92.95	43.48	29.36	85.43	33.64	23.90
↪**Rewind**	**96.61**	**45.95**	**35.50**	**85.79**	**47.62**	**27.59**

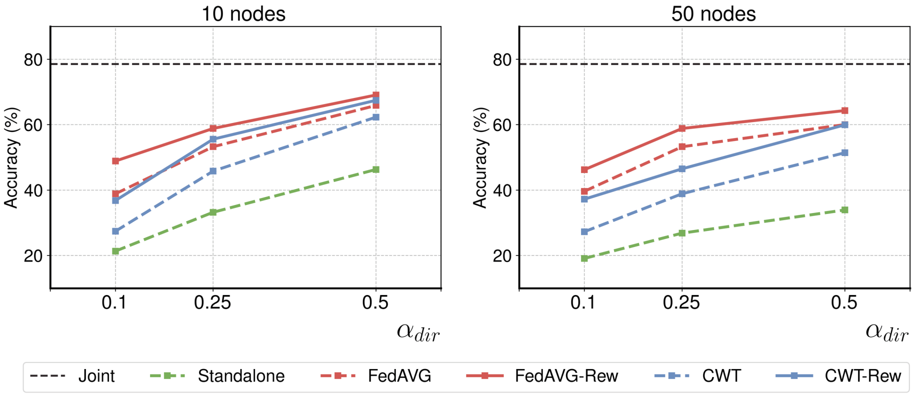


Fig. 2. Performance at different degrees of data heterogeneity (α_{dir}) on CIFAR-10 for 10 (left) and 50 (right) nodes.

extreme non-IID conditions, to 0.5, representing a less skewed data distribution among nodes. The results, shown in Fig. 2, demonstrate that our rewind strategy consistently enhances FA, with the highest gain obtained at $\alpha_{dir} = 0.1$, the most challenging setting. This suggests that the rewind strategy is particularly effective in environments with high data distribution skewness. As α_{dir} increases, *FedRewind* continues to provide benefits, though they are less pronounced. These findings collectively demonstrate that the rewind strategy is a robust method for federated learning, capable of enhancing model performance in diverse data conditions. Its consistent performance across different levels of non-IIDness, as shown in Fig. 2, suggests reliable applicability in real-world federated settings where data distributions vary widely.

We finally verify whether the enhanced performance is due to rewinding to the previous node or to any other node. Thus, we compared our rewind strategy to a random rewinding one, where the model is sent to a random node of the federation. Table 4 shows the performance of the two strategies when combined to CWT and RWT, highlighting how rewinding to the previous node in the communication chain is more effective than using a random node. It has to be noted that, in RWT, the performance increase is slightly lower because the preceding node changes at each communication round, slightly reducing the benefits of rewinding.

Table 4. Comparison between different rewinding strategies. RandRewind sends the model back to a random done instead of the previous node as Rewind does.

Method	CIFAR-10		CIFAR-100	
	10 Nodes	50 Nodes	10 Nodes	50 Nodes
CWT	45.81	40.00	34.02	24.90
↪**RandRewind**	51.59	45.76	36.58	**27.60**
↪ **Rewind**	**55.57**	**47.83**	**37.44**	27.51
RWT	44.16	38.75	32.08	24.42
↪ **RandRewind**	50.83	45.62	36.62	**27.40**
↪**Rewind**	**52.83**	**45.85**	**36.72**	27.09

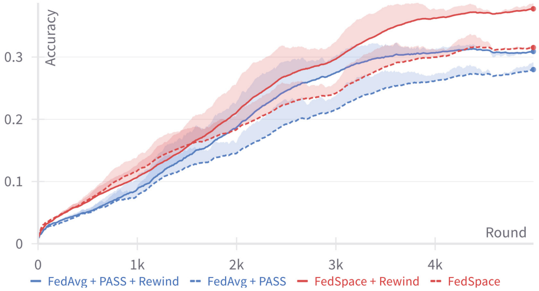


Fig. 3. Training trend of rewind strategy in AFCL

4.2 Continual Federated Learning

We also evaluate *FedRewind* within the complex context of Asynchronous Federated Continual Learning (AFCL) [27], where data is not only distributed across multiple nodes (as in federated learning) but also subject to changing distributions over time (as in continual learning). In this asynchronous setting, each node independently progresses through its continual learning tasks, creating unique

Table 5. Results of rewind strategy in AFCL

Method	Accuracy
FedAvg + PASS	29.70
↪ **Rewind**	**33.46**
FedSpace	35.53
↪ **Rewind**	**39.40**

distribution shifts at different times. We argue that the *rewind* strategy is particularly advantageous in AFCL scenarios, as rewinding on another node can mitigate the exacerbated problem of forgetting.

To test this hypothesis, we implement our strategy on top of the current state-of-the-art approach for AFCL, FedSpace [27]. We replicate their experimental setup, using CIFAR100 divided into 10 tasks of 10 classes each, and maintain the same hyperparameters, except for the number of epochs per round, and without any pretraining. Specifically, we rerun their experiments with $E = 3$ because the default value of 1 was incompatible with the rewind strategy. The experimental results and trends are detailed in Table 5 and Figure 3, respectively, where the *rewind* strategy is integrated into FedSpace. Additionally, we applied our proposed strategy to the same baseline used in [27], where PASS [38], a continual learning strategy, is adapted to the federated scenario by combining it with FedAvg. In both cases, incorporating the rewind strategy results in enhanced performance, while maintaining computational costs low.

5 Conclusion

In this paper, we introduce *FedRewind*, a novel approach that incorporates the *rewind* technique into federated learning (FL) scenarios to address challenges arising from non-i.i.d. data distributions across distributed nodes. By periodically exchanging and rewinding models among nodes, *FedRewind* mitigates issues related to overfitting on locally skewed data, which can hinder model generalizability and lead to catastrophic forgetting. This method significantly enhances performance by promoting robustness against class imbalances and improving overall model generalization, even in the complex context of Asynchronous Federated Continual Learning (AFCL).

We first validated *FedRewind* on standard federated learning scenarios, demonstrating significant improvements in performance and generalization over existing methods such as FedAVG, CWT, and RWT. Importantly, these improvements were achieved without increasing computational costs, facilitating seamless integration into existing FL frameworks. We further evaluated our approach in the more extreme context of AFCL, surpassing existing methods.

In conclusion, by integrating concepts from continual learning and leveraging cognitive neuroscience principles, *FedRewind* reduces the impact of distribution shifts, providing a robust solution to the challenges posed by non-i.i.d. data in distributed learning environments.

Acknowledgements. We acknowledge the support of the PNRR ICSC National Research Centre for High Performance Computing, Big Data and Quantum Computing (CN00000013), "Federated Learning as a Service (FLaaS) with Generative AI - GenF - Innovation Grant", under the NRRP MUR program funded by the NextGenerationEU. Matteo Pennisi is a PhD student enrolled in the National PhD in Artificial Intelligence, XXXVII cycle, course on Health and life sciences, organized by Universitá Campus Bio-Medico di Roma.

References

1. Bellitto, G., Pennisi, M., Palazzo, S., Bonicelli, L., Boschini, M., Calderara, S.: Effects of auxiliary knowledge on continual learning. In: 2022 26th International Conference on Pattern Recognition (ICPR). pp. 1357–1363. IEEE (2022)
2. Beltrán, E.T.M., Pérez, M.Q., Sánchez, P.M.S., Bernal, S.L., Bovet, G., Pérez, M.G., Pérez, G.M., Celdrán, A.H.: Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. IEEE Communications Surveys & Tutorials (2023)
3. Boschini, M., Bonicelli, L., Porrello, A., Bellitto, G., Pennisi, M., Palazzo, S., Spampinato, C., Calderara, S.: Transfer without forgetting. In: European Conference on Computer Vision. pp. 692–709. Springer (2022)
4. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark Experience for General Continual Learning: a Strong. Advances in Neural Information Processing Systems, Simple Baseline. In (2020)
5. Chang, K., Balachandar, N., Lam, C., Yi, D., Brown, J., Beers, A., Rosen, B., Rubin, D.L., Kalpathy-Cramer, J.: Distributed deep learning networks among institutions for medical imaging. J. Am. Med. Inform. Assoc. **25**(8), 945–954 (2018)
6. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
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 (2009). 10.1109/CVPR.2009.5206848
8. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Process. Mag. **29**(6), 141–142 (2012)
9. Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X., Zhu, Q.: Federated class-incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10164–10173 (2022)
10. 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 (2016)
11. Kalra, S., Wen, J., Cresswell, J.C., Volkovs, M., Tizhoosh, H.R.: Decentralized federated learning through proxy model sharing. Nat. Commun. **14**(1), 2899 (2023)
12. Karpicke, J.D., Blunt, J.R.: Retrieval practice produces more learning than elaborative studying with concept mapping. Science **331**(6018), 772–775 (2011)

13. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences (2017)
14. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images. Tech. rep, Citeseer (2009)
15. Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10713–10722 (2021)
16. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems **2**, 429–450 (2020)
17. Li, X., Jiang, M., Zhang, X., Kamp, M., Dou, Q.: Fedbn: Federated learning on non-iid features via local batch normalization. arXiv preprint arXiv:2102.07623 (2021)
18. Lian, X., et al.: Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. NeurIPS (2017)
19. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7765–7773 (2018)
20. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. Psychology of learning and motivation (1989)
21. McMahan, B., et al.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
22. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks (2019)
23. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: iCaRL: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2017)
24. Robins, A.: Catastrophic forgetting, rehearsal and pseudorehearsal. Connection Science (1995)
25. Roediger, H.L., Butler, A.C.: The critical role of retrieval practice in long-term retention. Trends Cogn. Sci. **15**(1), 20–27 (2011)
26. Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y.W., Pascanu, R., Hadsell, R.: Progress & compress: A scalable framework for continual learning. In: International Conference on Machine Learning (2018)
27. Shenaj, D., Toldo, M., Rigon, A., Zanuttigh, P.: Asynchronous federated continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5054–5062 (2023)
28. Shoham, N., et al.: Overcoming forgetting in federated learning on non-iid data. arXiv:1910.07796 (2019)
29. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning across heterogeneous clients. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8432–8440 (2022)
30. Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., Khazaeni, Y.: Federated learning with matched averaging. arXiv preprint arXiv:2002.06440 (2020)
31. Wink, T., Nochta, Z.: An approach for peer-to-peer federated learning. In: 2021 51st Annual IEEE/IFIP DSN-W (2021)

32. Yang, X., Yu, H., Gao, X., Wang, H., Zhang, J., Li, T.: Federated continual learning via knowledge fusion: A survey. IEEE Transactions on Knowledge and Data Engineering (2024)
33. Yao, X., Sun, L.: Continual local training for better initialization of federated models. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 1736–1740. IEEE (2020)
34. Yoon, J., Jeong, W., Lee, G., Yang, E., Hwang, S.J.: Federated continual learning with weighted inter-client transfer. In: International Conference on Machine Learning. pp. 12073–12086. PMLR (2021)
35. Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., Khazaeni, Y.: Bayesian nonparametric federated learning of neural networks. In: International conference on machine learning. pp. 7252–7261. PMLR (2019)
36. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning (2017)
37. Zhu, C., Xu, Z., Chen, M., Konečnỳ, J., Hard, A., Goldstein, T.: Diurnal or nocturnal? federated learning of multi-branch networks from periodically shifting distributions. In: International Conference on Learning Representations (2022)
38. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5871–5880 (2021)
39. Zhu, Z., Hong, J., Zhou, J.: Data-free knowledge distillation for heterogeneous federated learning. In: International conference on machine learning. pp. 12878–12889. PMLR (2021)

On the Relationship Between Double Descent of CNNs and Shape/Texture Bias Under Learning Process

Shun Iwase[1](✉), Shuya Takahashi[1,2,3,4], Nakamasa Inoue[2], Rio Yokota[2], Ryo Nakamura[3], Hirokatsu Kataoka[4], and Eisaku Maeda[1]

[1] Tokyo Denki University, Adachi City, Tokyo, Japan
23amj03@ms.dendai.ac.jp, maeda.e@mail.dendai.ac.jp
[2] Tokyo Institute of Technology, Meguro City, Tokyo, Japan
[3] Tenchijin Inc., Chuo City, Tokyo, Japan
[4] National Institute of Advanced Industrial Science and Technology, Tsukuba, Ibaraki, Japan

Abstract. The double descent phenomenon, which deviates from the traditional bias-variance trade-off theory, attracts considerable research attention; however, the mechanism of its occurrence is not fully understood. On the other hand, in the study of convolutional neural networks (CNNs) for image recognition, methods are proposed to quantify the bias on shape features versus texture features in images, determining which features the CNN focuses on more. In this work, we hypothesize that there is a relationship between the shape/texture bias in the learning process of CNNs and epoch-wise double descent, and we conduct verification. As a result, we discover double descent/ascent of shape/texture bias synchronized with double descent of test error under conditions where epoch-wise double descent is observed. Quantitative evaluations confirm this correlation between the test errors and the bias values from the initial decrease to the full increase in test error. Interestingly, double descent/ascent of shape/texture bias is observed in some cases even in conditions without label noise, where double descent is thought not to occur. These experimental results are considered to contribute to the understanding of the mechanisms behind the double descent phenomenon and the learning process of CNNs in image recognition.

Keywords: Double Descent · Shape/Texture Bias · Pre-training

1 Introduction

Deep learning has become an important research area with applications in many fields such as computer vision. To build high-performance models with deep learning, it is essential to prepare appropriate training datasets and determine the number of model parameters according to these datasets. Two crucial phenomena to understand in this context are underfitting[20] and overfitting[28].

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 95–109, 2025.
https://doi.org/10.1007/978-3-031-78389-0_7

Underfitting occurs even with sufficient training data if there are too few model parameters and prevents the model from learning features from the training data, leading to suboptimal performance on test data. Conversely, overfitting occurs when there is either too little training data or an excess of model parameters, causing the model to fit too closely to the training data and hindering generalization to new test data. Researchers widely recognize these phenomena as the bias-variance trade-off[24].

However, recent studies report an interesting phenomenon: when the number of model parameters becomes very large, performance can improve again after overfitting[1,2,5,21]. Belkin et al. name this phenomenon "double descent" and demonstrate it in models such as two-layer neural networks and random forests.[2] Later, Nakkiran et al. confirm this phenomenon in more practical deep neural networks such as convolutional neural networks (CNNs) for image classification.[21] Moreover, they show that in addition to increasing the number of model parameters, increasing the training epochs can also induce double descent. Many studies on double descent focus on elucidating its theoretical foundations. Particularly, research on double descent due to an increase in the number of training epochs considers the features of the data and how models learn these features [13,22,26]. However, such investigations mainly use artificial data or non-deep learning models, and studies centered on the characteristics of real data in deep learning, such as shape and texture specific to images, have rarely been conducted.

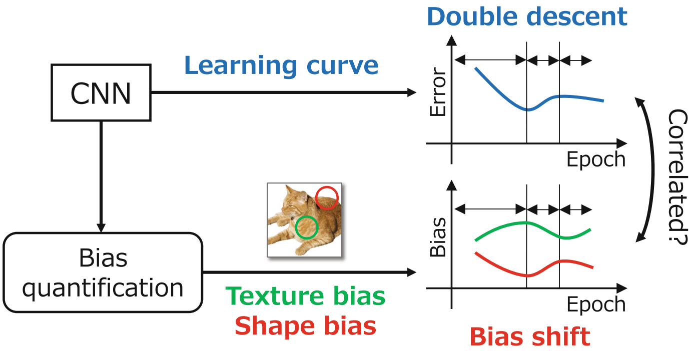


Fig. 1. Flow of analysis process presented in this paper. We train CNNs for image recognition under double descent conditions. We monitor the temporal evolution of the shape/texture bias and test error assessing the capacity of the model to interpret shapes and textures while exploring their correlation.

On the other hand, in image classification, analyses focusing on features such as shape and texture reveal that CNN models trained on ImageNet tend to be biased towards texture features [10]. It has also been found that this texture bias can be reduced using simple data augmentation and prolonged training, leading to a stronger bias towards shape features [14]. Given these characteristics of CNNs, important questions arise about the relationship between the double descent phenomenon and learning of shape and texture features with CNNs. However, a comprehensive exploration of the relationship between the learning phases for shape and texture and the various phases of double descent has not yet been conducted.

In this study, we delve into the relationship between image-specific features (shape and texture) and the double descent of CNNs. The flow of analysis is shown in Fig. 1. First, we define the period until the initial increase in test error as Phase 1, the period until it starts to decrease as Phase 2, and the subsequent period as Phase 3. Second, we assess the relationship between test error and the bias toward shape and texture during these phases. Specifically, we quantify the shape/texture bias of the CNN model by utilizing the method proposed by Islam et al. [17] during training, and compare the trajectories of the bias with the progression of double descent. Furthermore, we calculate the correlation coefficients between the test errors and the bias values in each of the three phases for a more quantitative evaluation. We also conduct ablation studies and analyses under various conditions. The contributions of the present paper are as follows:

- We conduct the first analysis of features of natural images, such as shape and texture, in the context of the double descent phenomenon in deep learning. We calculate the model's bias toward shape and texture and compare the evolution of this bias with changes in test errors throughout the learning process.
- As a result, we find that shape/texture bias and test error are often correlated in Phases 1 and 2. In Fig. 2, a strong correlation is present in Phases 1 and 2, while Phase 3 tends to show no correlation, under Nakkiran's setting. Interestingly, the inflection points in the temporal progression of the test error and the shape/texture bias almost coincide. To the best of our knowledge, we are the first to report the double descent/ascent phenomena of shape/texture bias and its synchronization with the double descent phenomena of test error.
- To better understand the phenomena, we perform ablation studies and analyses beyond Nakkiran's setting, including experiments with various CNN architectures and different noise levels. One interesting finding is that we observe double descent/ascent of shape/texture bias even without adding noise to the labels, a condition where double descent is thought not to occur (Fig. 6).

2 Related Work

2.1 Double descent

A recently discovered phenomenon called double descent [2] shows that as model complexity increases further, performance improves again. In other words, after the initial U-shaped curve (transition from underfitting to overfitting), a new phase of performance improvement appears with increased complexity. Overparameterized deep neural networks, theoretically prone to overfitting, sometimes demonstrate superior generalization performance [3,7,11]. Belkin et al. [2]

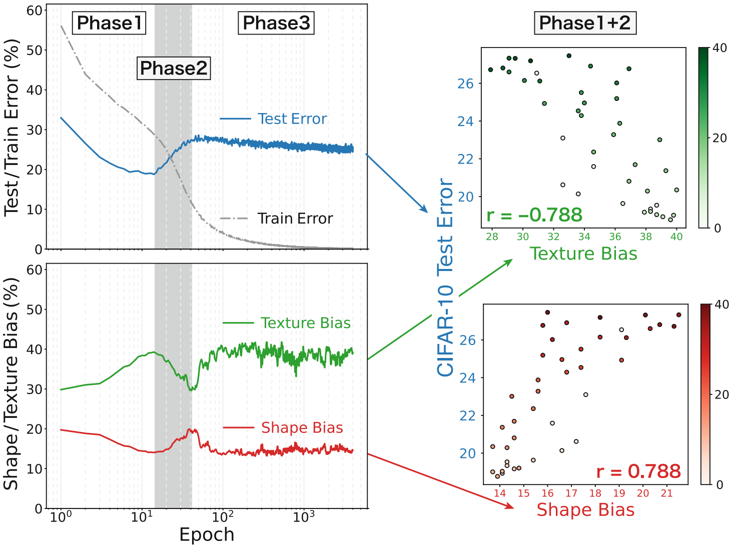


Fig. 2. Schematic overview of this study. Top left: The learning curve of the CIFAR-10 image recognition task under the setting of [21] et al, where epoch-wise double descent was observed. Test errors were divided into three phases based on their temporal differentiation. Bottom left: This records the model's shape/texture bias during the aforementioned learning process. It shows the synchronous changes between test errors and shape/texture bias. Right: A scatter plot of test error and shape/texture bias. Especially in Phase 1 and Phase 2, there is a positive correlation between test error and shape bias, and a negative correlation between test error and texture bias. In all bias visualization settings, including this one, we use a 5-term moving average to smooth the data for trend analysis.

first confirmed double descent in decision trees and two-layer neural networks. Later, Nakkiran et al. [21] showed that it also occurs in deep neural networks and with more learning epochs. Reports also indicate double descent happens with increased sparsity due to parameter pruning [12]. Double descent observed with more parameters, learning epochs, and increased sparsity is called model-wise double descent, epoch-wise double descent, and sparse double descent, respectively [12,21]. The discovery of these phenomena challenges traditional interpretations related to the design and parameter selection of models. It significantly impacts both the theory and practice of machine learning. Understanding and utilizing these phenomena could contribute to the development of more efficient and versatile machine-learning models.

Model-wise double descent. Yang et al. [29] revisited the classic theory of the bias-variance trade-off through extensive experiments. They found that while bias monotonically decreases as classification theory predicts, variance shows unimodal behavior. This combination of bias and variance suggests three typical risk curve patterns, aligning with many already reported experimental results.

Epoch-wise double descent. Several hypotheses about double descent in the learning process emerge from statistical simulation results. These hypotheses focus on the characteristics of the data. For example, Stephenson et al. [26] assume that double descent occurs due to slow yet beneficial features and show that removing the principal components of data in an ideal linear model can eliminate the double descent behavior. On the other hand, Pezeshki et al. [22] find through experiments that features learned at different scales cause double descent. Moreover, Heckel et al. [13] state that overlapping trade-offs between multiple biases and variances, due to different parts of the model learning at different epochs, trigger double descent. They demonstrate that varying learning rates across layers can mitigate double descent.

Sparse double descent. As the model's sparsity increases, meaning many parameters become zero or very small, we first observe performance improvement. However, performance declines after a certain point. Further increasing sparsity, performance improves again [12,23]. This suggests that moderate sparsity, achievable through methods like network pruning, can suppress model overfitting and enhance generalization performance.

2.2 CNN for image understanding

Geirhos et al. [10] showed that CNNs trained on ImageNet especially emphasize image textures for classification. The input images with conflicting shape and texture information into CNNs and checked whether the output matched shape-based or texture-based labels. Based on these results, they analyzed whether CNNs prioritize shape or texture in recognition. Meanwhile, Islam et al. [17] proposed a method to quantitatively determine the emphasis on shape and texture in models based on neurons' latent representations. This method allowed them to analyze which features CNNs emphasize or ignore. Furthermore, Ge et al. [9] attempted to model the human visual system and developed the Human Vision System (HVS). HVS can quantitatively evaluate which features (shape, texture, color, etc.) play the most crucial role during image classification. Our research builds on prior studies about CNNs in image understanding and double descent. We attempted to reveal the relationship between the acquisition of knowledge about texture and shape information during CNN learning and the phenomenon of double descent.

3 Correlation analysis framework of double descent and shape/texture bias

This section explains how to investigate the relationship between epoch-wise double descent in deep learning and the shape and texture features of natural images. Figure 1 outlines this method. First, we train a CNN under conditions that cause double descent and observe the progression of the learning curve. Additionally, we quantify the bias towards shape and texture features at each epoch using the method of Islam et al. [17] and similarly observe its progression

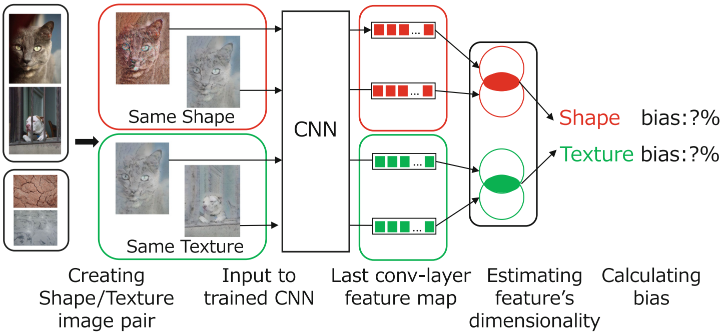

Same Shape
Same Texture
CNN
Shape bias:?%
Texture bias:?%
Creating Shape/Texture image pair
Input to trained CNN
Last conv-layer feature map
Estimating feature's dimensionality
Calculating bias

Fig. 3. Overview of Islam's method. This figure shows an overview of the process of calculating the shape and texture bias using the method of Islam et al. [17]

during training. By doing so, we compare the progression of double descent and bias. Furthermore, for quantitative evaluation, we divide double descent into three phases and assess the correlation coefficient between test error rate and shape/texture bias in each phase. The following sections explain the observation method for epoch-wise double descent, the division of its phases, and the calculation method for shape and texture bias.

3.1 How to observe double descent

To observe epoch-wise double descent, we use the conditions originally used by Nakkiran et al. [21]. They have observed double descent under various conditions. Among these conditions, we adopt the setup involving ResNet18 [11] and CIFAR-10 [19] as the baseline condition, for our study. In this condition, they added noise to the labels of the training data. This addition makes the double descent more pronounced.

3.2 Phases of learning curve with double descent

In this study, in order to analyze the relationship between double descent and the model's shape/texture bias, we divide the learning process into the following three phases based on test error. **Phase 1**: From the beginning of training until the test error reaches its minimum. **Phase 2**: From the end of Phase 1 until the test error decreases again. **Phase 3**: From the end of Phase 2 thereafter.

To determine these phases, we utilize a gradient-based method. Specifically, we monitor the test error across epochs and compute the difference Δe between consecutive epochs as $\Delta e = |e_i - e_{i+5}|$. At any given epoch i, if the difference Δe is less than or equal to a specified threshold θ, the test error has either stabilized or has improved slightly. We define the interval up to the smallest epoch number at this time as Phase 1. For Phase 2, encompasses the interval from the epoch number just after Phase 1 to the smallest epoch where the difference is less than or equal to θ. Phase 3 refers to any interval following Phase 2. In our experiments, the threshold θ is set at 0.1. In the experimental setup used, the phases are divided by the process described above because empirically the second descent of a double descent does not have a lower test error rate than the first descent.

3.3 Quantifying the shape/texture bias of the model

We estimate the number of neurons encoding shape and texture features in the final convolutional layer of CNNs using the method proposed by Islam et al. [17], and define this ratio as the model's shape and texture bias[1]. The flow for calculating shape and texture bias is shown in Fig. 3. For quantifying shape and texture bias, we use the Stylized PASCAL VOC 2012 (SVOC) dataset [8] created by the AdaIn transfer algorithm[16] from the PASCAL VOC 2012 dataset and the Describable Textures Dataset [4]. We sample image pairs with common shape and texture features from the SVOC dataset. Then, we calculate the correlation coefficients ρ_i^{shape} and $\rho_i^{texture}$ for each feature. For example, we input image pairs with common shapes into the model and obtain outputs z_i^a and z_i^b from neuron z_i. We calculate the correlation coefficient ρ_i^{shape} from these outputs z_i^a and z_i^b. We follow a similar procedure to calculate $\rho^{texture}$. We determine the proportion of neurons encoding shape and texture features (shape and texture bias) respectively, by calculating the softmax of the sum of each ρ_i^{shape} and $\rho_i^{texture}$ and the baseline value (number of neurons $|z|$). For more details on the SVOC construction method and the concept of this technique, see Islam et al. [17] .

4 Experiments

In this section, the following three sets of experiments are conducted: 1) Under the setting of Nakkiran et al. [21], where epoch-wise double descent was confirmed, we compare the progression of test error rates and shape/texture bias. We also quantitatively investigate correlations in each phase defined in Fig. 3.2. 2) We conduct ablation studies and analyses to deepen the understanding of the relationship between the double descent of test error and shape/texture bias. 3) We conduct layer-wise analyses by evaluating the shape/texture bias of each layer and visualizing the filters of the first layer.

4.1 Nakkiran's setting

Detailed settings. The ResNet18 model with weights pre-trained on ImageNet [6] is trained on CIFAR-10 [19] using label noise and data augmentation. The label noise involves randomly changing the correct label of the training data to another label with a probability of $p = 0.2$. For data augmentation, random

[1] https://github.com/islamamirul/shape_texture_neuron.

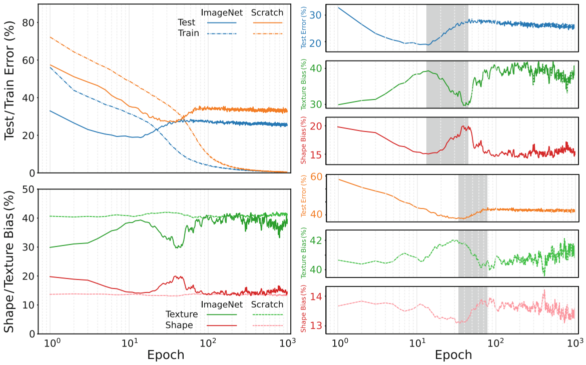


Fig. 4. Comparison of training with randomly initialized weights (Scratch) and with weights pre-trained on ImageNet (ImageNet). Left top: train and test errors (%). Left bottom: shape/texture bias (%). Right: Enlarged view.

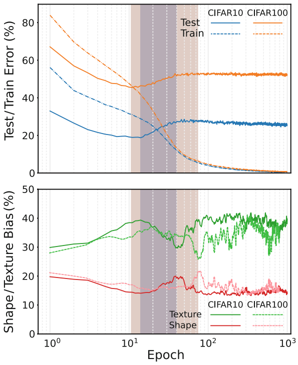


Fig. 5. Comparison of CIFAR-10 and CIFAR-100. Top: train and test errors (%). Bottom: shape/texture bias (%).

cropping and flipping are utilized. For cropping, a 4-pixel margin is added to the top, bottom, left, and right sides of the input image, and then the image is cropped to a size of 32 × 32. Flipping is applied horizontally. The batch size is set to 128. The cross-entropy loss is used as the loss function. For optimization, Adam [18] is used with a learning rate of 10^{-4}. This experimental condition is consistent with that of Nakkiran et al. [21], where an epoch-wise double descent phenomenon was observed.

Experimental results. The results comparing the progression of the test error and the shape/texture bias are shown in Fig. 2. We observed double descent phenomena of test error and double descent/ascent phenomena of shape/texture bias. Comparing the trends in the test error and the shape bias, a correlation is observed with a decrease in Phase 1, an increase in Phase 2, and a decrease again in Phase 3. There is also an inverse correlation between the test error and the texture bias. The inflection points in the temporal progression of the test error and the shape/texture bias almost coincide.

Correlation analysis in each phase. We calculated the correlation coefficients between the test error and the shape/texture bias in Phases 1, 2, and 3 for a more detailed evaluation. The results are shown in Fig. 1. The correlation coefficient r_{shape} for shape bias in Phases 1 and 2 were 0.898 and 0.771, respectively, indicating a positive correlation. Conversely, the correlation coefficient r_{texture} for texture bias in Phases 1 and 2 were −0.829 and −0.797, indicating a negative correlation. In Phase 3, these correlation coefficients were −0.026 and 0.118, respectively, showing no significant correlation. These results indicate that there are correlations in Phases 1 and 2, but not in Phase 3.

Synchronization score. To simplify the evaluation, we introduce synchronization score s defined by the average of the absolute values of the two cor-

Table 1. Correlation coefficients and synchronization scores. Phase: the three phases divided according to the method defined in 3.2. Epoch range: The start and end epoch of each phase. r_{shape}: correlation coefficients between shape bias and test error. r_{texture}: correlation coefficients between texture bias and test error. s: synchronization score.

Phase	Epoch range	r_{shape}	r_{texture}	s
Phase 1	2 - 12	0.898	-0.829	0.863
Phase 2	12 - 41	0.771	-0.797	0.784
Phase 3	41 - 1,000	-0.026	0.118	0.072

Table 2. Correlation coefficients and synchronization scores for CIFAR-10 and CIFAR-100 datasets.

Dataset	Correlation in Phase 1, 2			Correlation in Phase 3		
	r_{shape}	r_{texture}	s	r_{shape}	r_{texture}	s
CIFAR-10	0.778	-0.778	0.778	-0.026	0.118	0.072
CIFAR-100	0.133	-0.118	0.126	0.162	0.324	0.227

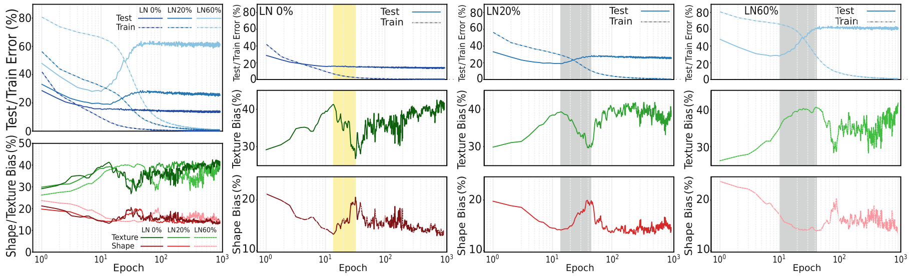


Fig. 6. Learning process under various label noise conditions. The label noise proportion p is varied at 0, 0.2, and 0.6. While double descent is not observed in test error when $p = 0$, we observed double descent/ascent phenomena in shape/texture bias.

relation coefficients, *i.e.*, $s = \frac{1}{2}(|r_{\text{shape}}| + |r_{\text{texture}}|)$. A higher score indicates a stronger synchronization between test error and shape/texture bias. In Fig. 1, strong synchronization with scores greater than 0.7 is observed in Phases 1 and 2.

4.2 Ablation studies and analyses

To better understand the relationship between the double descent of test error and the double descent/ascent of shape/texture bias, we perform ablation studies and analyses beyond Nakkiran's setting. Specifically, we conduct experiments with respect to parameter initialization, dataset selection, architecture and label noise.

Parameter initialization. To investigate the effect of parameter initialization, this experiment compares training with randomly initialized weights and with weights pre-trained on ImageNet. The results are shown in Fig. 4. We

Table 3. Correlation coefficients and synchronization scores for different label noise proportion p. Results for $p = 0$ are not reported because double descent was not detected.

p	Correlation in Phase 1, 2			Correlation in Phase 3		
	r_{shape}	r_{texture}	s	r_{shape}	r_{texture}	s
20%	0.778	−0.778	0.778	−0.026	0.118	0.072
60%	−0.560	0.598	0.579	0.122	−0.142	0.132

observed three phenomena. First, in both cases, the texture/shape bias fluctuates synchronously with the double descent phenomenon of test error, as shown in the enlarged view on the right side of the figure. Second, when random initialization is used, the absolute change of the bias is smaller. This is due to the residual effect of the Gaussian initialization. Third, when training with pre-trained weights, the transition to Phase 2 is faster. This is because the training error decreases faster. Overall, we observed similar phenomena in both cases.

Dataset. This experiment examines the effect of changing the training dataset. We report results comparing the CIFAR-10 and CIFAR-100 datasets in Fig. 5 and the quantitative evaluation in Fig. 2. We observed synchronization scores larger than 0.7 in Phase 1 and 2 on both datasets. The CIFAR-100 results show a corresponding correlation to the CIFAR-10 results: the correlation coefficients with respect to shape and texture bias were 0.778 and −0.778 for CIFAR-10, respectively; correspondingly, they were −0.689 and 0.745 for CIFAR-100. This suggests that a similar dataset shows the same synchronization between test error and shape/texture bias.

Label noise. This experiment varies the proportion of label noise at 0%, 20%, and 60%. The results are shown in Fig. 6 and Fig. 3. As the trend of the test error rate reveals, the magnitude of double descent increases as the label noise grows. However, when observing the shape/texture bias, there is a clear trend regardless of the label noise ratio. Especially in the texture bias, as the label noise increases, the shift from rising to declining seems to be delayed. This suggests that the progression of bias may slow down as label noise increases. More interestingly, when the noise proportion is 0%, double descent was not observed in test error, but we observed double descent/ascent phenomena of shape/texture bias (as colored in yellow in the figure). This indicates the possibility that there are learning phases even in parts where double descent was thought not to occur.

Architecture. This experiment examines CNN architectures other than ResNet. Specifically, we use MobileNetV2 [25], DenseNet121 [15], and Effecient-NetB0 [27]. We show the results in Fig. 7 and Fig. 4. The results reveal a moderate synchronization when using MobileNetV2 and DenseNet121. However, we observed no synchronization with EfficientNetB0. This is because there is a spike at around epoch 10 in Phase 1 that significantly reduces the synchronization score. This bias spike is currently an unknown phenomenon but could be related to the loss spike [30], suggesting room for further discussion from the perspective of the stochastic learning process.

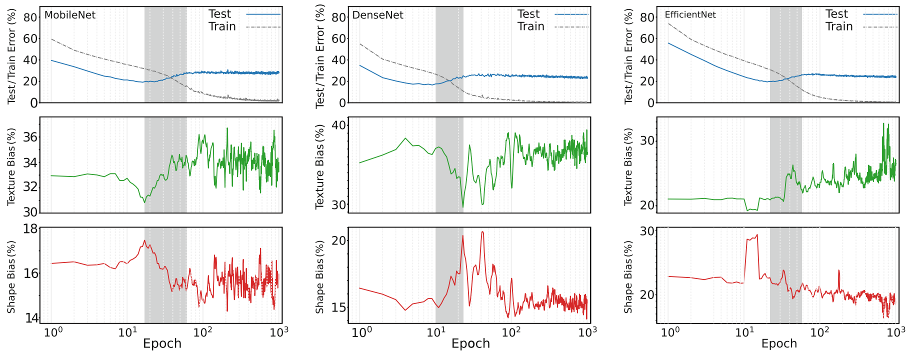


Fig. 7. Comparison of learning processes using different architectures. Left: MobileNetV2, Center: DenseNet121, Right: EfficientNetB0.

Table 4. Correlation coefficients and synchronization scores for different architectures.

Architecture	Correlation in Phase 1, 2			Correlation in Phase 3		
	r_{shape}	r_{texture}	s	r_{shape}	r_{texture}	s
MobileNet	−0.506	0.511	0.509	−0.016	0.036	0.026
DenseNet	0.326	−0.322	0.324	0.293	−0.289	0.291
EfficientNet	−0.029	0.000	0.014	0.316	−0.343	0.330

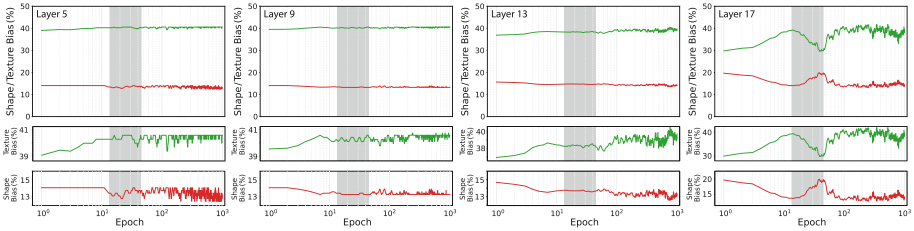
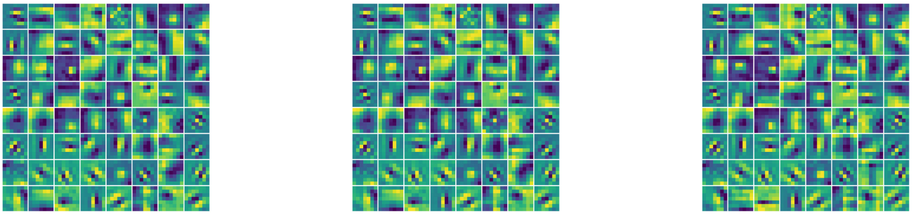


Fig. 8. The shift of biases during the learning process in each layer consisting ResNet18. Using the convolutional layers of each block consisting ResNet18, including the final convolutional layer (17th layer), biases towards shape and texture are calculated using the same method as in 3.3.

4.3 Layer-wise analyses and visualization

Here, we conduct layer-wise analyses to investigate which layers are influenced by the bias. First, we analyze the shape/texture bias for hidden layers. Second, we visualize the filters of the first convolution layer in each phase.

Shape/texture bias of hidden layers. Figure 8 shows the shape/texture bias of 5th, 9th, 13th and 17th layers of ResNet18. These results reveal that in convolutional layers, except for the 17th layer, there is no clear transition in shape/texture bias. This suggests that each layer may have unique inflection points and that the last few layers have a direct impact on test error.



(a) 13th Epoch (b) 42nd Epoch (c) 1,000th Epoch

Fig. 9. Visualization of the 1st layer in the learning process: In the setting described in Fig. 4.1, we visualize the 1st layer in the Epoch (13th, 42nd Epoch) and the 1,000th Epoch, where the double descent is divided into 3 Phases. The 1st layer at the 1,000th Epoch is visualized.

Filter visualization. Figure 9 visualizes the filters of the first convolution layer of ResNet18 at three points: the boundary between Phase 1 and Phase 2, the boundary between Phase 2 and Phase 3, and the 1,000th epoch. Although there are slight changes, the visualization confirms no significant changes in the filters. Considering the results of Fig. 4.1, this suggests that learning in shallow layers might not be affected by changes in shape/texture bias. This is consistent to the result of Fig. 8 where we observe a clear transition only on the 17th layer.

5 Discussion

In this study, we focused on shape and texture features in natural images and analyzed their relation to the double descent phenomenon. We found that under certain conditions, there is a double descent/ascent of shape/texture bias synchronized with a double descent of test error. In these conditions, there tends to be a correlation between the test error and shape/texture bias until the second descent in double descent of test error. However, this correlation disappears once the second descent begins. We discovered this trend by dividing double descent into three phases and quantitatively evaluating the correlations. In subsequent experiments, we observed the synchronization of the bias and test error only in the final convolutional layer. This observation suggests that the deeper layers of the CNN may exhibit learning trends different from those of the intermediate layers.

Previous studies on double descent have proposed the hypothesis that it might be influenced by multiple features present in data. But do features like shape and texture directly cause the double descent phenomenon of test error? If such features were truly causing this phenomenon, then double descent behavior and bias towards shape and texture should show a more clear pattern. For example, shape bias might peak first, then decrease, followed by texture bias peaking. However, in reality, shape and texture biases show an inverse correlation. Therefore, we believe that the phenomenon is more complex than it seems and that some learning tendencies causing double descent affect CNN's feature extraction tendencies, showing a correlation between double descent and bias progression.

From a practical perspective, when pre-trained on ImageNet, the possibility is suggested that test error may be minimized around epochs where bias is at its maximum or minimum. This implies that observing this bias could help determine the optimal number of training epochs. Furthermore, it was shown that factors causing double descent might also affect CNN's bias towards shape and texture features, especially in deeper layers.

In this study, we focused on the learning process of image features like shape and texture in CNNs and examined their relation to the double descent phenomenon under various conditions. There are many unexplored areas in deep learning, and double descent is one of them. This research highlights the importance of a deeper understanding of the deeper layers of deep learning, and it is considered to provide a promising direction for future research. However, given the complexity of neural network architecture, there are still unknown phenomena that need to be investigated in future research.

6 Conclusion

In this paper, inspired by previous studies on epoch-wise double descent, we focused on the relationship between image-specific features and double descent. We quantified the model's bias toward shape and texture to compare it with the test error. As a result, we discovered double descent/ascent of shape/texture bias synchronized with double descent of test error under Nakkiran's setting. Additionally, quantitative evaluations confirmed this correlation during the period from the initial decrease to the full increase in the test error. Further, we observed double descent/ascent of shape/texture bias in a condition without label noise, where double descent was thought not to occur. Layer-wise analyses deepened the understanding of the phenomena.

Limitation and future work. To make a connection to previous studies, we chose Nakkarian's setting as a starting point for investigation. We believe this was the best choice for discussing the relationship between test error and shape/texture bias because it is the simplest setting where epoch-wise double descent of test error can be reproduced. However, this remains an investigation under large-scale training as future work. As recent computer vision studies focus more on large-scale training, it would be interesting to study the scaling of double descent. Furthermore, extending bias quantification methods to modalities other than images, such as natural language and speech, is also challenging but promising to deepen the understanding of the double descent/ascent of bias in deep learning.

References

1. Advani, M.S., Saxe, A.M., Sompolinsky, H.: High-dimensional dynamics of generalization error in neural networks. Neural Netw. **132**, 428–446 (2020)
2. Belkin, M., Hsu, D., Ma, S., Mandal, S.: Reconciling modern machine-learning practice and the classical bias-variance trade-off. Proceedings of the National Academy of Sciences (PANS) **116**(32), 15849–15854 (2019)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Proc. Annual Conference on Neural Information Processing Systems (NeurIPS). vol. 33, pp. 1877–1901 (2020)
4. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , Vedaldi, A.: Describing textures in the wild. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
5. Dar, Y., Mayer, P., Luzi, L., Baraniuk, R.G.: Subspace fitting meets regression: The effects of supervision and orthonormality constraints on double descent of generalization errors. In: Proc. International Conference on Machine Learning (ICML) (2020)
6. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255 (2009)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proc. International Conference on Learning Representations (ICLR) (2021)
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
9. Ge, Y., Xiao, Y., Xu, Z., Wang, X., Itti, L.: Contributions of shape, texture, and color in visual recognition. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Proc. European Conference on Computer Vision (ECCV). pp. 369–386 (2022)
10. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In: Proc. International Conference on Learning Representations (ICLR) (2019)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
12. He, Z., Xie, Z., Zhu, Q., Qin, Z.: Sparse double descent: Where network pruning aggravates overfitting. In: Proc. International Conference on Machine Learning (ICML). pp. 8635–8659 (2022)
13. Heckel, R., Yilmaz, F.F.: Early stopping in deep networks: Double descent and how to eliminate it. In: Proc. International Conference on Learning Representations (ICLR) (2021)
14. Hermann, K.L., Chen, T., Kornblith, S.: The origins and prevalence of texture bias in convolutional neural networks. In: Larochelle, H., Ranzato, M., Hadsell,

R., Balcan, M., Lin, H. (eds.) Proc. Annual Conference on Neural Information Processing Systems (NeurIPS) (2020)

15. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2261–2269 (2017)
16. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1510–1519 (2017)
17. Islam, M.A., Kowal, M., Esser, P., Jia, S., Ommer, B., Derpanis, K.G., Bruce, N.D.B.: Shape or texture: Understanding discriminative features in cnns. In: Proc. International Conference on Learning Representations (ICLR) (2021)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) Proc. International Conference on Learning Representations (ICLR) (2015)
19. Krizhevsky, A., Hinton, G.: Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto (2009)
20. Li, Z., Liu, L., Dong, C., Shang, J.: Overfitting or underfitting? understand robustness drop in adversarial training. CoRR **abs/2010.08034** (2020)
21. Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., Sutskever, I.: Deep double descent: Where bigger models and more data hurt. J. Stat. Mech: Theory Exp. **2021**(12), 124003 (2021)
22. Pezeshki, M., Mitra, A., Bengio, Y., Lajoie, G.: Multi-scale feature learning dynamics: Insights for double descent. In: Proc. International Conference on Machine Learning (ICML). pp. 17669–17690 (2022)
23. Quétu, V., Milovanovic, M., Tartaglione, E.: Sparse double descent in vision transformers: Real or phantom threat? In: International Conference on Image Analysis and Processing (ICIAP). pp. 490–502 (2023)
24. Rajnarayan, D., Wolpert, D.: Bias-Variance Trade-offs: Novel Applications, pp. 101–110. Springer US, Boston, MA (2010). https://doi.org/10.1007/978-0-387-30164-8_75
25. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4510–4520 (2018)
26. Stephenson, C., Lee, T.: When and how epochwise double descent happens. CoRR **abs/2108.12006** (2021)
27. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: Proc. International Conference on Machine Learning (ICML). pp. 6105–6114 (2019)
28. Webb, G.I.: Overfitting, pp. 744–744. Springer US, Boston, MA (2010).https://doi.org/10.1007/978-0-387-30164-8_623
29. Yang, Z., Yu, Y., You, C., Steinhardt, J., Ma, Y.: Rethinking bias-variance tradeoff for generalization of neural networks. In: Proc. International Conference on Machine Learning (ICML). pp. 10767–10777 (2020)
30. Zhang, Z., Xu, Z.Q.J.: Loss spike in training neural networks. CoRR **abs/2305.12133** (2023)

Cystic Adenocarcinoma Segmentation Based on Multi-frequency and Multi-scale SimAM Attention

Xia Zhang[1], Jian Tan[2(✉)], Bailing Zhang[2(✉)], Guodong Xu[3(✉)], Zeyang Hu[3], Rui Wang[3], Jing Qiu Yang[3], and Chaoyi Pang[3]

[1] Zhejiang Sci-Tech University, Hangzhou, China
[2] NingboTech University, Ningbo, China
{jt,bailing.zhang}@nit.zju.edu.cn
[3] Ningbo Medical Center Lihuili Hospital, Ningbo, China
xuguodong@nbu.edu.cn, yangjingqiu@nbt.edu.cn

Abstract. Accurate medical image segmentation is crucial for early diagnosis in clinical medicine. However, neural networks for medical segmentation often overlook the combination of frequency and spatial domains, and the employed attention mechanisms treat each channel neuron equally, forming 1D or 2D weights. Such an approach fails to compute true 3D weights effectively. The SimAM paper mentions 3D attention, but its final expression formula suggests that the results are related only to the global mean and variance, without considering local information. We propose a multi-frequency attention model in multi-scale parameter-free attention (LungSSFNet) for lung segmentation in cystic adenocarcinoma datasets to address these challenges. The proposed model includes three key components: the parameter-free attention mechanism (S), the improved feature concatenation method (U), and the multi-scale, multi-frequency attention module (SSF). The U component improves upon traditional feature concatenation by providing a more effective method of capturing differences between deep semantic and shallow features. Thirdly, the SSF component is a multi-scale, multi-frequency attention module based on parameter-free 3D weights. This provides module that can capture the contours of small targets and tissue boundaries significantly. To ensure that the optimal model is not solely a result of parameter tuning, we leverage the automatic configuration module of nnU-Net to determine the parameters. These parameters will remain fixed during subsequent model evaluation. Through extensive experiments, we demonstrated that LungSSFNet consistently outperforms the state-of-the-art models by 1–2% in the segmentation of cystic adenocarcinoma. Our LungSSFNet code is available at https://github.com/zx0412/LungSSFNet.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78389-0_8.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 110–125, 2025.
https://doi.org/10.1007/978-3-031-78389-0_8

Keywords: medical image segmentation · parameter-free attention · multi-scale · multi-frequency · feature fusion · spatial attention

1 Introduction

Cancer is the principal cause of death in the world [1]. Lung cancer is the most common and among the deadliest cancers [2,3], killing more people than the bladder, brain, breast, colorectal, prostate, and stomach cancers combined [4–6]. It accounts for about 1.8 million new cases and more than 1.4 million deaths every year worldwide [1,7].

Lung cancer is often diagnosed only at critical stages. However, early diagnosis of lung cancer is fundamental to improving survival rates by enhancing treatment decisions [2,8]. It is estimated that the five-year survival rate for patients has increased by over 50% due to early diagnosis and timely treatment of lung cancer [9,10]. For prognostic imaging of lung cancer, pulmonologists and radiologists recommend examinations such as computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET) [9,11]. Both MRI and PET have limitations in detecting pulmonary nodules [12]. Specifically, MRI may miss small pulmonary metastases [3]. CT, particularly low-dose CT, can significantly reduce the incidence of advanced lung cancer, thereby markedly decreasing lung cancer mortality rates [13]. In addition to being most sensitive to small calcified pulmonary nodules [5,14,15], CT has other advantages due to its high spatial resolution including low noise and distortion, speed, non-invasiveness, cost-effectiveness, and widespread availability [9,11,16].

Lung cancer typically presents as solid masses or nodules on imaging studies, with $2 - 16\%$ of lung cancers exhibiting cavitary features [17], often presenting as thick-walled eccentric cavities. Research indicates that when the cavity wall thickness is ≤ 4 mm, 92% of the lesions are benign; when the wall thickness is ≥ 15 mm, 95% of the lesions are malignant. For wall thicknesses between 4 mm and 15 mm, 51% of the lesions are benign, and 49% are malignant [18].

Thin-walled cavitary lung cancer is a rare type of lung cancer that appears as a lesion with thin-walled cavities on chest CT scans. It is extremely rare clinically and has been scarcely reported both domestically and internationally. In the International Early Lung Cancer Action Program (I-ELCAP), only 3.7% of lung cancers were identified as thin-walled cavitary lung cancer [17]. This type of lung cancer was first reported by Womack NA in 1940 [18] and has since been variously named, including "thin-walled cavitary lung cancer," "bullous lung cancer," "lung cancer associated with cystic air spaces," "cystic lung cancer," and "cystic cavity-type lung cancer." Currently, there is no standardized terminology for this condition globally. This type of lung cancer is commonly characterized by the presence of thin-walled air cavities. These cavities can originate from pre-existing structures such as pulmonary bullae, lung cysts, or bronchiectasis. Alternatively, they may result from necrosis of the lung cancer lesion, which is expelled through the bronchi. In this paper, we collectively describe these as Cystic Adenocarcinoma. Moreover, Cystic Adenocarcinoma shares imaging characteristics with cavitary lesions, making clinical diagnosis challenging and prone to misdiagnosis or missed diagnosis [19].

In the field of lung cancer, the combination of pulmonary nodule detection and deep learning has achieved remarkable results. This approach is not only theoretically robust but also widely applied in clinical practice. However, another type of lung cancer, cystic adenocarcinoma, currently relies on traditional CT imaging interpreted by experienced physicians. Addressing this challenge, recent advances in deep learning [48,49] have paved the way for its application in medical image segmentation. In particular, U-Net has become the fundamental network structure for medical segmentation due to its skip connections, which effectively utilize both deep semantic features and shallow features.

With the widespread adoption of attention mechanisms in recent years, numerous models combining U-Net and attention mechanisms have been proposed, such as Attention U-Net [26] and Focus U-Net [47]. Nevertheless, due to the inherent characteristics of medical images, the performance of these models varies significantly across different datasets. Therefore, exploring a model with practical clinical application value remains challenging.

To fully leverage the characteristics of medical imaging, particularly the frequency domain properties of medical images [21], and to utilize multi-scale features that capture structures and details at different scales-thereby improving diagnostic accuracy, reducing noise, adapting to various imaging techniques, and meeting clinical requirements [23]-we propose the Multi-Frequency in Multi-Scale SimAM Attention (SSF). This is inspired by SimAM [20], which provides three-dimensional, parameter-free attention weights, allowing us to increase recognition accuracy while keeping the parameter count low. Additionally, to enhance the fusion of rich deep semantic features with shallow features, we introduce a SimAM-based Subtraction Unit, drawing inspiration from M^2SNet [23].

In the clinical application of cystadenocarcinoma, our methods demonstrate an approximate 3% accuracy improvement over nnUnet [28] and MADGNet [21], with our model (Lung-SSFNet).

2 Related Works

In this section, we briefly discuss representative works on network architectures and plug-and-play attention mechanisms in the context of semantic segmentation.

2.1 Model Architecture

In recent years, various advanced network architectures have been developed for CT medical segmentation. A prominent architecture is the U-Net [22], which has gained widespread popularity due to its effective encoder-decoder structure that captures context and precise localization. Following this, the nnU-Net [28] has been introduced as a self-configuring method that adapts to different datasets and tasks, achieving state-of-the-art performance. SegNet [32] and ResNet [33–35] are also notable, with the former being an encoder-decoder architecture for image segmentation, adapted for medical imaging tasks. Due to the relatively simple deployment of the nnU-Net [28] network architecture and its powerful preprocessing capabilities, which automatically generate suitable hyperparameters based on the dataset without the need for extensive tuning to find an optimal model, we adopted this network structure. Without altering the parameters, we validated the feasibility of our proposed algorithm using this framework.

2.2 Attention Mechanisms

Attention mechanisms play a crucial role in enhancing the performance of network architectures by focusing on the most relevant parts of the input data. For instance, Channel Attention (CA) mechanisms, such as Squeeze-and-Excitation Networks [36], improve representational power by emphasizing important channels. Spatial Attention (SA) mechanisms focus on spatial locations that are critical for the task, such as Non-Local Networks [37], which enhance the model's ability to capture long-range dependencies by considering the relationships between all pixel pairs. Combining both channel and spatial attention, the Convolutional Block Attention Module (CBAM) [38] and the Parameter-Free Attention Module (SimAM) [20] effectively capture rich contextual dependencies.

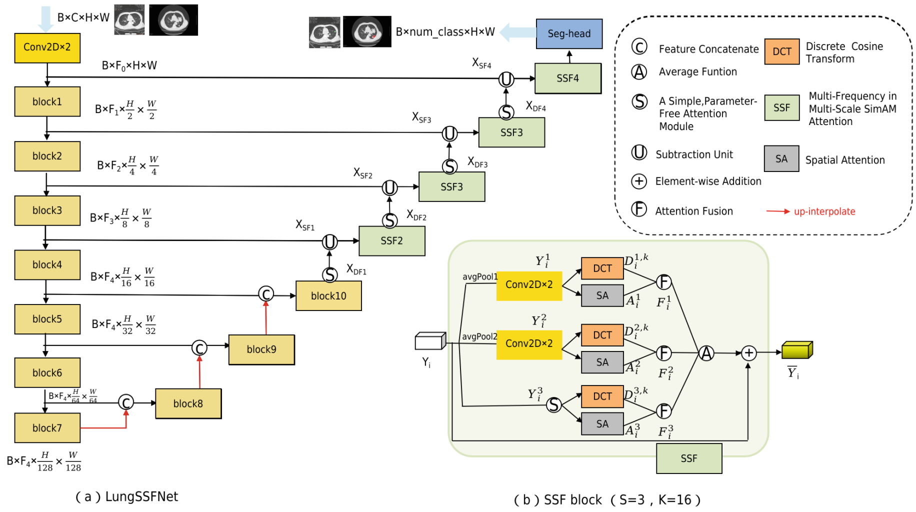
Recently, multi-scale features have been extensively applied in semantic segmentation to capture features at various resolutions, often in conjunction with attention mechanisms [39,40]. These mechanisms have demonstrated the effectiveness of enlarging the receptive field to capture multi-scale features [41–44].

Multi-frequency attention mechanisms have gained attention for their ability to leverage frequency domain information, which is crucial for capturing texture and boundary details that might be missed in the spatial domain [45,46]. For instance, MADGNet [21] proposed a generalized MFMSA block based on 2D DCT basis functions.

However, current medical image segmentation methods often focus on multi-scale and multi-frequency aspects separately, without simultaneously considering the distinct information provided by the frequency and spatial domains. To address this limitation, we integrate the frequency domain and spatial domain with multi-scale features and 3D attention [20,21,30]. Enhancing the model's ability to segment small areas with significant inter-class and intra-class variations is crucial for the clinical application of cystic adenocarcinoma.

3 Methods

Our method is divided into three parts. The first part presents the overall model architecture. The second part introduces the SimAM-based Subtraction Unit. The third part details the Multi-Frequency in Multi-Scale SimAM attention.



(a) LungSSFNet

(b) SSF block (S=3 , K=16)

Fig. 1. (a) The overall architecture of the proposed LungSSF mainly comprises the "block" and "SSF" modules. The "block" is composed of two CONV2D layers: the first CONV2D layer has a 3x3 kernel with a stride of 1, and the second has a 3x3 kernel with a stride of 2. (b) First, the "SSF" obtains multi-scale features through pooling with different convolution kernel sizes and SimAM. These multi-scale features are then represented in the spatial domain (Spatial Attention) and the frequency domain (Discrete Cosine Transform). Finally, the characterized multi-scale features are aggregated using the Attention Fusion and Average functions.

3.1 Model Architecture

Our proposed model, LungSSFNet, integrates multi-scale and multi-frequency attention mechanisms to enhance the segmentation performance of cystic adenocarcinoma. The overall architecture is depicted in Figure 1.

3.2 Fusion of Shallow Features And Deep Semantic Features Unit

Motivation: In medical image segmentation, integrating attention mechanisms can significantly enhance convolution performance [24–26]. In U-Net [22], the fusion of deep semantic features and shallow features involves a significant amount of shallow features in the concatenation operation. This leads to high computational costs with minimal contributions. M^2SNet [23] proposes a fusion method by subtracting deep semantic features from shallow features, but this approach overlooks the advantages of combining both types of features as done in U-Net. Our method incorporates *SimAM* (whose function will be explained in Subsection 3.2) after the shallow features and then fuses them with deep semantic features through the Subtraction Unit and feature concatenation. We define this process as the two operators **S** and **U** in Figure 1. The entire process can be divided into the following two steps: *1) Feature Extraction, 2) Fusion Unit.*

Feature Extraction

It has been suggested that low-level features require more computational resources due to their larger spatial resolutions but contribute less to overall performance compared to high-level features [27]. Based on this idea, we define deep semantic features and shallow features as X_{DF} and X_{SF} in Figure 1. Using nnU-Net [28] as the basic network structure, as shown in Figure 1, each $Conv2D$ consists of two 3×3 convolutions with a stride of 1, and Each block$_i$ ($i = 1, \ldots, 10$) consists of two 3×3 convolutions: the first with a stride of 1 and the second with a stride of 2. After passing through the convolution layers up to block$_{10}$, significant deep semantic features are formed.

Fusion Unit

In each convolution layer, we introduce two learnable parameters α and β. Y_i is the result formed after the operations S and U. The specific calculation formula is as follows:

$$Y_i = \alpha_i \times \mathrm{Conv}(X_i) + \beta_i \times \mathrm{Conv}(|X_{DF_i} \ominus X_{SF_i}|) \tag{1}$$

where $Cat(\cdot)$ denotes 2D convolution with a kernel size of 3, $Up(\cdot)$ denotes interpolation, and $SimAM(\cdot)$ represents parameter-free attention. Here, $\ominus$ is the element-wise subtraction operation, $|\cdot|$ calculates the absolute value, and $Conv(\cdot)$ denotes 2D convolution with a kernel size of 3.

$$X_i = \mathrm{Cat}(\mathrm{Up}(\mathrm{SimAM}(X_{DF_i})),\ X_{SF_i}) \tag{2}$$

By adding a subtraction operation to the traditional feature concatenation of X_{DF} and X_{SF}, the edge and contour information in the image can be highlighted, helping the model capture important details in the image. SimAM assigns different weights to each neuron in X_{DF}, similar to the formation of human visual features.

3.3 Multi-Frequency in Multi-Scale SimAM Attention

Motivation: Existing attention mechanisms generate 1-D or 2-D weights from feature X, which are then extended to channel or spatial attention [20]. Human visual information acquisition occurs across multiple scales, and in medical imaging, representation in the frequency domain exhibits greater variance compared to the spatial domain [21]. *SimAM*[20] leverages the latest findings in visual neuroscience, defining the energy function of neurons to form 3-D weights, creating parameter-free attention. However, from the final derived formula, it is evident that it ultimately only takes into account the global. We define this as the Multi-Frequency in Multi-Scale *SimAM* Attention (*SSF*) module to address this challenge. The overall framework of the *SSF* module is shown in Figure 1. The *SSF* module can be divided into the following three steps: *1) Multi-Scale SimAM Attention, 2) Combining Frequency And Spatial Domain Information, 3) Attention Fusion.*

Multi-Scale SimAM Attention In the human brain, certain active neurons can inhibit the activities of surrounding neurons, a phenomenon known as spatial suppression [29]. In other words, this translates to determining which neurons should be given higher priority during visual processing. The simplest approach is identifying the linear separability of the target neuron from other neurons. Based on this concept, the energy function for a neuron can be defined as follows:

$$e_t(w_t, b_t, \mathbf{y}, x_i) = (y_t - \hat{t})^2 + \frac{1}{M-1}\sum_{i=1}^{M-1}(y_i - \hat{x}_i)^2 + \lambda w_t^2 \tag{3}$$

To linearly classify the target neuron $\hat{t} = w_t t + b_t$ and other neurons $\hat{x_i} = w_t x_i + b_t$, with labels $y_t = -1$ and $y_o = 1$ respectively. Here, $\hat{t}$ and $\hat{x_i}$ are elements of the input feature map $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$ in a single channel. The optimization of w_t and b_t is performed using the Mean Squared Error (MSE) loss as the objective function. Additionally, a regularization term is included, as shown in Equation (3), where λ is the regularization constant and $M = H \times W$.

$$e_t^* = \frac{4(\sigma_t^2 + \lambda)}{(t - \mu_t)^2 + 2\sigma_t^2 + 2\lambda} \tag{4}$$

Assuming that, apart from the target neuron t, the other neurons in the channel have a consistent pixel distribution. When the energy function e_t reaches its minimum, the corresponding w_t and b_t constitute the optimal linear classification function. A smaller value of the energy function indicates greater linear separability between t and x_i, thereby signifying the higher importance of t. There is a mathematical solution for Equation (4). Consequently, the term $1/e_t$ is passed through a *sigmoid* activation function to represent the importance. Finally, this computed weight is multiplied element-wise with the input feature map to obtain the attention weight map, as shown in Equation 6.

$$Y_i^s = \text{Conv2D} \times 2\left(\text{avgPool}_s(Y_i)\right) \tag{5}$$

To generate multi-scale features from the input feature Y_i, we use pooling layers with varying kernel sizes and strides. Specifically, for each scale s, the average pooling (avgPool_s) kernel size and stride are both set to $2^{(s-1)}$, producing Y_i^s, as shown in Equation 5. However, for the case when $s = 3$, we bypass the avgPool and directly apply SimAM. In Equation 6, E is the result of e after being computed by all neurons.

$$SimAM(Y_i^s) = \sigma\left(\frac{1}{\mathbf{E}}\right) \odot \mathbf{Y_i^s}, \tag{6}$$

Combining Frequency And Spatial Domain Information

To capture image attention in the frequency domain, we utilize the approach described in [21], as illustrated in Equation 7. Here, (u_k, v_k) represent the frequency indices associated with $\mathbf{D}_i^{s,k}$. Additionally, the 2D

DCT basis images at the s-th scale branch are defined as $\mathbf{D}_{h,w}^{u_k,v_k} = \cos\left(\frac{\pi h}{H_s}\left(u_k + \frac{1}{2}\right)\right)\cos\left(\frac{\pi w}{W_s}\left(v_k + \frac{1}{2}\right)\right)$, employing a top-$K$ selection strategy as per [50]. Following this, each $\mathbf{D}_i^{s,k}$ is reduced to $\mathbf{Z}_{\text{avg}}$, $\mathbf{Z}_{\text{max}}$, and $\mathbf{Z}_{\text{min}}$ through Global Average Pooling, Global Max Pooling, and Global Min Pooling, respectively. These frequency statistics are then aggregated to form a channel attention map at the s-th scale branch. This is achieved by employing two fully-connected layers, $\mathbf{W}_1 \in \mathbb{R}^{C_s \times \frac{C_s}{r}}$ and $\mathbf{W}_2 \in \mathbb{R}^{\frac{C_s}{r} \times C_s}$, where r is the reduction ratio.

$$D_i^{s,k} = \sum_{h=0}^{H_s-1}\sum_{w=0}^{W_s-1}(Y_i^s)\mathbf{D}_{h,w}^{u_k,v_k} \tag{7}$$

To obtain image attention in the spatial domain, we refer to the method proposed in [30], as shown in Equation 8.

$$A_i^s = SA(Y_i^s) \tag{8}$$

The input feature map Y_i^s, with dimensions $b \times c \times w \times h$, undergoes two parallel pooling operations: MaxPooling (S_{max}) and AveragePooling (S_{avg}). Both operations produce feature maps of size $b \times 1 \times w \times h$. These pooled feature maps are concatenated along the channel axis to form a combined feature map, $[\mathbf{S}_{avg}; \mathbf{S}_{max}]$, with dimensions $b \times 2 \times w \times h$. This combined feature map is then convolved with a 7×7 kernel to produce a feature map of size $b \times 1 \times w \times h$. The resulting feature map is passed through a sigmoid activation function, $\sigma(\cdot)$, generating the spatial attention map, Att_{SA}, which has dimensions $b \times 1 \times w \times h$. The spatial attention map is then element-wise multiplied with the original input feature map Y_i^s, resulting in the refined feature map A_i^s with dimensions $b \times c \times w \times h$. This refined feature map A_i^s emphasizes the important spatial regions of the input feature map, enhancing the network's focus on relevant areas.As detailede in supplementary material.

Attention Fusion

The frequency and spatial domain recalibrated feature maps $D_i^{s,k}$ and A_i^s are used to determine discriminative boundary cues with different scales in the frequency and spatial domains. At this stage, we introduce two learnable parameters (α_i^s and β_i^s) for each scale branch to control the information flow between the frequency and spatial domains, respectively, as shown in Equation 9.

$$F_i^s = \text{Conv}\left(\alpha_i^s D_i^{s,k} + \beta_i^s A_i^s\right), \tag{9}$$

First, $\mathbf{F}_i^s$ is reshaped to match the shape of Y_i. Then, $\mathbf{F}$ is aggregated at different scales using the aggregation function A to examine the noise present in medical images. After aggregation, the summed result is divided by the number of scale branches, and finally, it is added to Y_i to obtain the final output $\overline{Y_i}$.

$$\overline{Y_i} = Y_i + \mathbf{A}(F_i^s,\ \text{Up}_s(F_i^s)) \tag{10}$$

4 Experiments

Our experimental description encompasses five aspects: dataset introduction, selection of evaluation metrics, experimental setup, experimental results, and ablation study results.

4.1 Datasets

The dataset used in this experiment was obtained from a public hospital, Ningbo Medical Center Lihuili Hospital, Ningbo, China, and was annotated by professional thoracic surgeons. The dataset comprises a total of 342 images, with 272 malignant cases and 70 benign cases. The dataset was divided into training, validation, and test sets in a ratio of 245:62:35. As detailede in supplementary material.

The cystic adenocarcinoma datasets exhibit significant inter-class and intra-class variations. The dataset is derived from slices obtained from 3D instruments, encompassing the Axial Plane, Coronal Plane, and Sagittal Plane. The Axial Plane can display cross-sectional images of the lungs, facilitating the observation of lung structures and lesions. Therefore, only the Axial Plane was selected during the dataset filtering process. However, since the images come from different medical devices, their shapes are inconsistent and fall into the following four categories: (512, 512), (1024, 1024), (1445, 927), and (1284, 594). As detailed in supplementary material.

4.2 Evaluation Metrics

There are many popular metrics used in different medical segmentation branches. In the field of CT medical segmentation, commonly used evaluation metrics include IoU and DSC [21,23,30,53], as well as accuracy, precision, recall, and F1_Score [51–54]. Accuracy and recall are particularly significant in clinical applications, as they are directly correlated with missed diagnoses and misdiagnoses. The IoU and DSC measure the performance of a model at the pixel level.

4.3 Experiment Settings

We run all the experiments on a workstation with Ubuntu 20.04.6 operating system, RTX4090 GPU with 24 GB memory, 62 GiB of RAM, 13th Gen Intel(R) Core(TM) i7-13700K (16 cores), and PyTorch 2.2.2 deep learning framework for implementation. In the training phase, we use the Stochastic Gradient Descent (SGD) optimizer with Nesterov momentum, an initial learning rate of 0.01, a weight decay of 3e-5, and a momentum of 0.99. The optimizer is combined with a polynomial decay learning rate scheduler to adjust the learning rate during training. The batch size is empirically set to 7, considering the memory capacity of the GPU. All the experiment networks are trained for 500 epochs to ensure model convergence. Moreover, we employ the Binary Cross-Entropy Loss (BCE) and soft dice loss function, which is a well-known loss function in medical image segmentation, to optimize the training process of the proposed LungSFFNet.

4.4 Comparison With SOTA Models

As shown in Table 1, LungSSFNet achieved the highest segmentation performance in various clinical environments compared to other models. Specifically, when compared to nnU-Net [28], which achieved the second-highest segmentation performance in most modalities, LungSSFNet improved the DSC and mIoU by an average of 1.7% and 1.14%, respectively, in the malignant classification. In the benign classification, LungSSFNet improved the DSC and mIoU by an average of 0.74% and 0.42%, respectively. Additionally, compared to MADGNet, which uses Multi-Frequency in Multi-Scale Attention, LungSSFNet improved the DSC and mIoU by an average of 1.0%.

Table 1. Segmentation results in model performance

Model	**Label**	**IoU**	**DSC**
Unet[22]	malignant	76.04	70.59
	benign	89.62	88.23
M^2SNet[23]	malignant	69.69	62.15
	benign	84.34	83.23
TANet[31]	malignant	74.63	68.68
	benign	89.83	88.57
MADGNet[21]	malignant	79.67	74.19
	benign	90.25	89.33
nnUnet[28]	malignant	79.91	73.54
	benign	92.61	91.30
LungSSFNet(ours)	malignant	**81.05**	**75.15**
	benign	**93.03**	**92.04**

As shown in Table 2, LungSSFNet achieved the highest segmentation performance across various models. In particular, compared to nnU-Net, which achieved the second-highest segmentation performance, LungSSFNet improved the accuracy (Acc) and F1-Score (F_S) by 2.84% and 1.88%, respectively. Additionally, when compared to M^2SNet, which uses multi-frequency attention, LungSSFNet improved both precision (Pre) and recall (Rec) by 3.85% and 3.30%. To evaluate the model's performance, we tested each model on the same dataset. As a result, LungSSFNet showed the highest performance with a precision (Pre) of 100.00%. Compared to TANet, the performance gap was significant. Nevertheless, LungSSFNet exhibits significant improvement in all metrics. These results indicate that models which do not consider Multi-Frequency in Multi-Scale SimAM Attention dimensions simultaneously cannot comprehend intricate anatomical knowledge.

Table 2. Segmentation results in clinical application

Model	**Acc**	**Pre**	**Rec**	**F_S**
Unet[22]	93.93	96.15	96.15	96.15
M^2SNet[23]	93.93	100.00	92.85	96.29
TANet[31]	90.90	92.30	96.00	94.11
MADGNet[21]	93.75	96.15	96.15	96.15
nnUnet[28]	93.93	96.15	96.15	96.15
LungSSFNet(ours)	**96.77**	**100.00**	**96.15**	**98.03**

Table 3. Ablation experiments of different attention mechanisms in the same framework

Model	**Label**	**IoU**	**DSC**
lungCBAM	malignant	78.42	71.79
	benign	92.61	91.13
LungSimAM	malignant	79.72	72.29
	benign	92.73	91.56
LungSSF1	malignant	71.48	66.66
	benign	90.14	89.13
LungSSF2	malignant	78.99	72.12
	benign	93.00	91.97
LungSSFNet(finally)	malignant	**81.05**	**75.15**
	benign	**93.03**	**92.04**

4.5 Ablation Study On LungSSFNet

We adopted nnU-Net [28] as the network architecture and named it "Lung". Subsequently, we experimented with attention mechanisms such as CBAM [38] and SimAM [20]. As shown in Table 3, the results indicate that LungSimAM slightly outperforms LungCBAM. Due to the significant advancements in the application of multi-scale and multi-frequency techniques in the medical segmentation field [21,23,30,43], we combined Multi-Frequency in Multi-Scale SimAM Attention with the "LungSimAM" network architecture, resulting in LungSSF1. However, the performance significantly declined. The core idea in [54] is that deep semantic information extracted from deeper convolutional layers is more abundant compared to shallow features extracted from shallower layers. However, the traditional U-Net network architecture wastes a considerable amount of computational resources on shallow features, which provide minimal image characteristics. By following this guiding principle and further ablation, we finally obtained LungSSFNet. In LungSSF2, the SSF module includes 4 blocks, and in LungSSF1, the SSF module includes 7 blocks. We found that increasing the number of SSF blocks decreases the model performance. Through further ablation, we concluded that having 4 SSF modules is optimal.

5 Conclusion

In conclusion, we would like to propose a novel medical image segmentation model called LungSSFNet, which can be used in clinical settings. One of the core components of our algorithm is the construction of Multi-Frequency and Multi-Scale SimAM Attention, which we refer to as the SSF block. Since SimAM primarily considers global features without incorporating local information into the attention weights, we integrated a spatial attention module that focuses on local features within the SSF block, termed the SA block. Through rigorous experiments, and leveraging the preprocessing capabilities of nnU-Net, we demonstrated that without tuning the parameters, the accuracy of our approach in clinical settings surpassed the state-of-the-art by 1-2 percentage points.

References

1. Prabhu, S., Prasad, K., Robles-Kelly, A., Lu, X.: AI-based carcinoma detection and classification using histopathological images: A systematic review. Comput. Biol. Medicine. **142**, 105209 (2022)
2. Monkam, P., Qi, S., Ma, H., Gao, W., Yao, Y.-D., Qian, W.: Detection and Classification of Pulmonary Nodules Using Convolutional Neural Networks: A Survey. IEEE Access. **7**, 78075–78091 (2019)
3. Naik, A., Edla, D.R.: Lung Nodule Classification on Computed Tomography Images Using Deep Learning. Wirel. Pers. Commun. **116**(1), 655–690 (2021)
4. Winkels, M., Cohen, T.S.: Pulmonary nodule detection in CT scans with equivariant CNNs. Medical Image Anal. **55**, 15–26 (2019)
5. Cao, W., Wu, R., Cao, G., He, Z.: A Comprehensive Review of Computer-Aided Diagnosis of Pulmonary Nodules Based on Computed Tomography Scans. IEEE Access. **8**, 154007–154023 (2020)
6. Sori, W.J., Jiang, F., Godana, A.W., Liu, S., Jobir, G.D.: DFD-Net: lung cancer detection from denoised CT scan image using deep learning. Frontiers Comput. Sci. **15**(2), 152701 (2021)
7. Pang, S., Meng, F., Wang, X., Wang, J., Song, T., Wang, X., Cheng, X.: VGG16-T: A Novel Deep Convolutional Neural Network with Boosting to Identify Pathological Type of Lung Cancer in Early Stage by CT Images. Int. J. Comput. Intell. Syst. **13**(1), 771–780 (2020)
8. Abid, M.M.N., Zia, T., Ghafoor, M., Windridge, D.: Multi-view Convolutional Recurrent Neural Networks for Lung Cancer Nodule Identification. Neurocomputing **453**, 299–311 (2021)

9. Zhang, G., Jiang, S., Yang, Z., Gong, L., Ma, X., Zhou, Z., Bao, C., Liu, Q.: Automatic nodule detection for lung cancer in CT images: A review. Comput. Biol. Medicine. **103**, 287–300 (2018)
10. Halder, A., Dey, D., Sadhu, A.K.: Lung Nodule Detection from Feature Engineering to Deep Learning in Thoracic CT Images: a Comprehensive Review. J. Digit. Imaging **33**(3), 655–677 (2020)
11. Zhang, G., Yang, Z., Gong, L., Jiang, S., Wang, L., Cao, X., Wei, L., Zhang, H., Liu, Z. An Appraisal of Nodule Diagnosis for Lung Cancer in CT Images. J. Medical Syst. 43(7), 181:1–181:18 (2019)
12. Thakur, S.K., Singh, D.P., Choudhary, J. Lung cancer identification: a review on detection and classification. Cancer and Metastasis Reviews. 39(3), 989–998 (2020). Springer
13. Detterbeck, F.C., Mazzone, P.J., Naidich, D.P., Bach, P.B. Screening for lung cancer: diagnosis and management of lung cancer: American College of Chest Physicians evidence-based clinical practice guidelines. Chest. 143(5), e78S–e92S (2013). Elsevier
14. Adiraju, R.V., Elias, S. A survey on lung CT datasets and research trends. Research on Biomedical Engineering. **37**(2), 403–418 (2021). Springer
15. Yu, H., Li, J., Zhang, L., Cao, Y., Yu, X., Sun, J. Design of lung nodules segmentation and recognition algorithm based on deep learning. BMC bioinformatics. 22, 1–21 (2021). Springer
16. Alakwaa, W., Nassef, M., Badr, A. Lung cancer detection and classification with 3D convolutional neural network (3D-CNN). International Journal of Advanced Computer Science and Applications. 8(8) (2017). Science and Information (SAI) Organization Limited
17. Opoka, L., Szturmowicz, M., Oniszh, K., Korzybski, D., Podgajny, Z., Blasińska-Przerwa, K., Szołkowska, M., Bestry, I.: CT imaging features of thin-walled cavitary squamous cell lung cancer. Advances in Respiratory Medicine. **87**(2), 114–117 (2019)
18. Womack, N.A., Graham, E.A. Epithelial metaplasia in congenital cystic disease of the lung: Its possible relation to carcinoma of the bronchus. The American Journal of Pathology. (5), 645 (1941). American Society for Investigative Pathology
19. Woodring, J.H., Fried, A.M., Chuang, V.P. Solitary cavities of the lung: diagnostic implications of cavity wall thickness. American Journal of Roentgenology. 135(6), 1269–1271 (1980). Am Roentgen Ray Soc
20. Yang, L., Zhang, R.-Y., Li, L., Xie, X. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 11863–11874. PMLR (2021)
21. Nam, J.-H., Syazwany, N.S., Kim, S.J., Lee, S.-C. Modality-agnostic Domain Generalizable Medical Image Segmentation by Multi-Frequency in Multi-Scale Attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11480–11491 (2024)
22. Ronneberger, O., Fischer, P., Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells III, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III. Lecture Notes in Computer Science, vol. 9351, pp. 234–241. Springer (2015)

23. Zhao, X., Jia, H., Pang, Y., Lv, L., Tian, F., Zhang, L., Sun, W., Lu, H. M^{2}SNet: Multi-scale in Multi-scale Subtraction Network for Medical Image Segmentation. CoRR. abs/2303.10894 (2023)
24. Schlemper, J., Oktay, O., Schaap, M., Heinrich, M.P., Kainz, B., Glocker, B., Rueckert, D.: Attention gated networks: Learning to leverage salient regions in medical images. Medical Image Anal. **53**, 197–207 (2019)
25. Gao, Y., Huang, R., Chen, M., Wang, Z., Deng, J., Chen, Y., Yang, Y., Zhang, J., Tao, C., Li, H. FocusNet: Imbalanced Large and Small Organ Segmentation with an End-to-End Deep Neural Network for Head and Neck CT Images. CoRR. abs/1907.12056 (2019)
26. Oktay, O., Schlemper, J., Le Folgoc, L., Lee, M.C.H., Heinrich, M.P., Misawa, K., Mori, K., McDonagh, S.G., Hammerla, N.Y., Kainz, B., Glocker, B., Rueckert, D. Attention U-Net: Learning Where to Look for the Pancreas. CoRR. abs/1804.03999 (2018)
27. Wu, Z., Su, L., Huang, Q. Cascaded Partial Decoder for Fast and Accurate Salient Object Detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 3907–3916. Computer Vision Foundation / IEEE (2019)
28. Isensee, F., Jaeger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods. 18(2), 203–211 (2021). Nature Publishing Group
29. Webb, B.S., Dhruv, N.T., Solomon, S.G., Tailby, C., Lennie, P. Early and late mechanisms of surround suppression in striate cortex of macaque. Journal of Neuroscience. 25(50), 11666–11675 (2005). Soc Neuroscience
30. Hettihewa, K., Kobchaisawat, T., Tanpowpong, N., Chalidabhongse, T.H. MANet: a multi-attention network for automatic liver tumor segmentation in computed tomography (CT) imaging. Scientific Reports. 13(1), 20098 (2023). Nature Publishing Group UK London
31. Li, Y., Yang, J., Ni, J., Elazab, A., Wu, J. TA-Net: Triple attention network for medical image segmentation. Computers in Biology and Medicine. 137, 104836 (2021). Elsevier
32. Badrinarayanan, V., Kendall, A., Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. CoRR. abs/1511.00561 (2015)
33. Gao, S., Cheng, M.-M., Zhao, K., Zhang, X., Yang, M.-H., Torr, P.H.S. Res2Net: A New Multi-scale Backbone Architecture. CoRR. abs/1904.01169 (2019)
34. He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. CoRR. abs/1512.03385 (2015)
35. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Mueller, J., Manmatha, R., Li, M., Smola, A.J. ResNeSt: Split-Attention Networks. CoRR. abs/2004.08955 (2020)
36. Hu, J., Shen, L., Sun, G. Squeeze-and-Excitation Networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 7132–7141. Computer Vision Foundation / IEEE Computer Society (2018)
37. Wang, X., Girshick, R.B., Gupta, A., He, K. Non-Local Neural Networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 7794–7803. Computer Vision Foundation / IEEE Computer Society (2018)
38. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S. CBAM: Convolutional Block Attention Module. CoRR. abs/1807.06521 (2018)

39. Sagar, A. DMSANet: Dual Multi Scale Attention Network. In: Sclaroff, S., Distante, C., Leo, M., Farinella, G.M., Tombari, F. (eds.) Image Analysis and Processing - ICIAP 2022 - 21st International Conference, Lecce, Italy, May 23-27, 2022, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13231, pp. 633–645. Springer (2022)
40. Xu, Q., Ma, Z., He, N., Duan, W.: DCSAU-Net: A deeper and more compact split-attention U-Net for medical image segmentation. Comput. Biol. Medicine. **154**, 106626 (2023)
41. Liu, S., Huang, D., Wang, Y. Receptive Field Block Net for Accurate and Fast Object Detection. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI. Lecture Notes in Computer Science, vol. 11215, pp. 404–419. Springer (2018)
42. Tan, M., Pang, R., Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, pp. 10778–10787. Computer Vision Foundation / IEEE (2020)
43. Su, R., Zhang, D., Liu, J., Cheng, C. MSU-Net: Multi-scale U-Net for 2D medical image segmentation. Frontiers in Genetics. 12, 639930 (2021). Frontiers Media SA
44. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125 (2017)
45. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. IEEE Trans. Comput. **100**(1), 90–93 (1974)
46. Shensa, M.J.: The discrete wavelet transform: wedding the a trous and Mallat algorithms. IEEE Trans. Signal Process. **40**(10), 2464–2482 (1992)
47. Yeung, M., Sala, E., Schönlieb, C.-B., Rundo, L.: Focus U-Net: A novel dual attention-gated CNN for polyp segmentation during colonoscopy. Comput. Biol. Medicine. **137**, 104815 (2021)
48. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
49. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. Attention is All you Need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 5998–6008 (2017)
50. Qin, Z., Zhang, P., Wu, F., Li, X. FcaNet: Frequency Channel Attention Networks. CoRR. abs/2012.11879 (2020)
51. Changhez, J., James, S., Jamala, F., Khan, S., Khan, M.Z., Gul, S., Zainab, I. Evaluating the Efficacy and Accuracy of AI-Assisted Diagnostic Techniques in Endometrial Carcinoma: A Systematic Review. Cureus. 16(5) (2024). Cureus
52. Hu, R., Li, H., Horng, H., Thomasian, N.M., Jiao, Z., Zhu, C., Zou, B., Bai, H.X. Automated machine learning for differentiation of hepatocellular carcinoma from intrahepatic cholangiocarcinoma on multiphasic MRI. Scientific reports. 12(1), 7924 (2022). Nature Publishing Group UK London
53. Abdelrahim, M., Saiko, M., Maeda, N., Hossain, E., Alkandari, A., Subramaniam, S., Parra-Blanco, A., Sanchez-Yague, A., Coron, E., Repici, A. Development and validation of artificial neural networks model for detection of Barrett's neoplasia: a multicenter pragmatic nonrandomized trial (with video). Gastrointestinal Endoscopy. 97(3), 422–434 (2023). Elsevier

54. Cao, K., Xia, Y., Yao, J., Han, X., Lambert, L., Zhang, T., Tang, W., Jin, G., Jiang, H., Fang, X. Large-scale pancreatic cancer detection via non-contrast CT and deep learning. Nature medicine. 29(12), 3033–3043 (2023). Nature Publishing Group US New York
55. Fan, D.-P., Ji, G.-P., Zhou, T., Chen, G., Fu, H., Shen, J., Shao, L. PraNet: Parallel Reverse Attention Network for Polyp Segmentation. CoRR.abs/2006.11392 (2020)

MSDNet: A Multi-scale Dense Network for Chip Surface Defect Segmentation

Xiaoyang Yu , Ziyi Zhu, Guanwen Zhang(✉), and Wei Zhou

Northwestern Polytechnical University, Xi'an, China
guanwen.zh@nwpu.edu.cn

Abstract. Chip surface defect detection is a critical research task and a vital component of integrated circuit quality inspection. With advancements in artificial intelligence, deep learning-based defect detection methods have become prevalent. However, due to the complex morphology of chip defects and their susceptibility to environmental background interference, precisely detecting micro-scale and multi-scale chip defects in large-scale, high-resolution images remains a significant challenge. In this paper, we propose a Multi-Scale Dense Network (MSDNet) for chip surface defect segmentation. The proposed MSDNet utilizes an encoder-decoder framework, incorporating multi-scale convolution modules, attention modules, and a dense node module to enhance defect segmentation performance. Additionally, we construct a chip defect dataset and conduct extensive experimental verifications. The experimental results demonstrate that the proposed MSDNet achieves an 85.01% defect segmentation accuracy on the chip defect dataset. Compared to the baseline, our proposed MSDNet significantly improves defect segmentation performance and ensures more precise segmentation of defect details.

Keywords: Chip surface defect detection · Defect segmentation · Deep learning · Multi-scale Features · Dense connection

1 Introduction

In the field of computer vision, detecting defects on chip surfaces is a critical research task and a fundamental aspect of integrated circuit quality inspection. Chips are essential components in numerous electronic products. Due to factors such as production techniques, equipment, and materials, defects often appear on chip surfaces. These defects can lead to performance degradation and functional abnormalities, consequently impacting the overall product performance. Identifying chip surface defects not only ensures chip quality but also enhances the reliability and stability of electronic products. Therefore, the detection of chip surface defects holds significant value.

Microscope-based chip defect detection methods [21] demand significant manpower and time, rendering them inefficient. Traditional image processing-based

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 126–139, 2025.
https://doi.org/10.1007/978-3-031-78389-0_9

defect detection methods [6,22] rely on manually designed features and are limited by factors such as lighting and environment conditions. These methods exhibit poor generalization performance and are unsuitable for complex industrial scenarios. In contrast, deep learning methods [4,8,9] enable autonomous feature learning, mitigating performance degradation caused by human factors. These methods have been increasingly adopted in industrial production, greatly improving the efficiency and accuracy of chip defect detection.

Current chip defect detection methods based on deep learning predominantly utilize convolutional neural networks (CNNs) to learn defect characteristics and determine locations using detection boxes, such as SSD [14] and YOLOv3 [18]. However, these approaches face challenges in accurately determining defect size, shape, and other characteristics. Segmentation models, such as FCN [15] and PSPNet [28], can achieve defect segmentation at the pixel level. Nevertheless, these methods have limitations in segmenting multi-scale chip surface defects. Firstly, as the network depth increases, it is difficult for these models to capture detailed information from feature maps. These networks often overlook tiny defects, resulting in poor performance when segmenting microscale defects that occupy only a few pixels. Additionally, some chip defects exhibit a high degree of similarity to the background. When defect characteristics are not pronounced, these methods tend to miss edges and details, leading to inaccurate segmentation. Furthermore, due to the lack of connections between different layers, existing segmentation networks cannot effectively leverage contextual information, resulting in poor segmentation of multi-scale defects.

In this paper, we propose a Multi-Scale Dense Network (MSDNet) based on an encoder-decoder architecture specifically designed for industrial chip surface defect segmentation. The network fully utilizes features at different scales, captures both global and detailed information, and demonstrates powerful feature learning capabilities.

Our contributions are as follows:

- We introduce multi-scale convolution modules in the encoding and decoding processes, resolving the issue of large size differences between distinct defect samples, preserving detailed information on feature maps, and enhancing the network's ability to detect microscale defects.
- We incorporate attention modules in the multi-scale network to address blurring and information loss, facilitating the detection of defect edges and detailed information.
- We propose a dense connection network between the encoder and decoder, where feature maps at different levels are extracted and fused through node modules, enhancing the network's capacity to capture global context information and facilitating feature learning and extraction across different scales.
- We construct a chip defect segmentation dataset, annotate the defects at the pixel level, and conduct a series of experiments on the dataset. The segmentation performance of our method on the chip defect dataset surpasses that of general segmentation models.

2 Related Works

As deep learning continues to evolve, convolutional neural networks (CNNs) have progressively taken precedence in surface defect detection. Based on application requirements, surface defect detection can be divided into three stages: defect classification, defect localization, and defect segmentation.

2.1 Defect Classification

Defect classification involves categorizing defects into different classes based on their characteristics. Xie et al. [25] developed a two-level hierarchical CNN system for sewer defect classification, utilizing two CNNs to distinguish defect images from normal ones and further classify defect images into specific categories. Lin et al. [10] proposed a method employing ResNet and hierarchical clustering to classify six types of defects, significantly reducing the misclassification of similar defects. Cheon et al. [2] constructed a CNN-based classification method for wafer surface damage, combining CNN and K-NN to classify both known and unknown defects.

2.2 Defect Detection

Defect localization aims to determine the precise location of defects using algorithms like Faster R-CNN [19] and YOLO [17]. Fang et al. [3] utilized Cascade R-CNN as the foundational framework, introducing a lightweight attention module to enhance feature extraction capabilities, multiple cascade head networks to improve the quality of region proposals, and Mix Non-Maximum Suppression to reduce detection redundancy and improve detection efficiency. Wang et al. [23] developed the MPSD model for mobile phone surface defect detection. This model, based on Faster R-CNN, integrates a feature pyramid network with ResNet-101 as the feature extraction network to capture small defect features, and replaces the ROI pooling layer with the ROI Align layer to reduce quantization bias. Hu et al. [5] enhanced YOLOv4 with CSP-ResNetSt and Bi-SimAM-FPN to identify small-scale defects in complex backgrounds.

2.3 Defect Segmentation

Defect segmentation involves separating defects from the background at the pixel level, with typical algorithms including U-Net [20] and SegNet [1]. Lin et al. [11] proposed CAM-UNet for anomalous image segmentation, employing an encoder-decoder architecture with skip connections and VGG-16 as the backbone. This approach emphasizes defect information by using class activation maps and feedback refinement. Yang et al. [26] designed a two-stage network called SIL-Net. In the first stage, an ULF module was added to SSD for defect identification and localization. In the second stage, a PGS composed of a principal component growth algorithm and adaptive thresholding was used for defect segmentation.

Liu et al. [13] developed the TAS^2-Net framework for surface defect segmentation. They enhanced and expanded defect samples using a GAN network, utilized the MFE module and triple context attention module to extract multi-level defect features, and employed the FCM module to fuse contextual information.

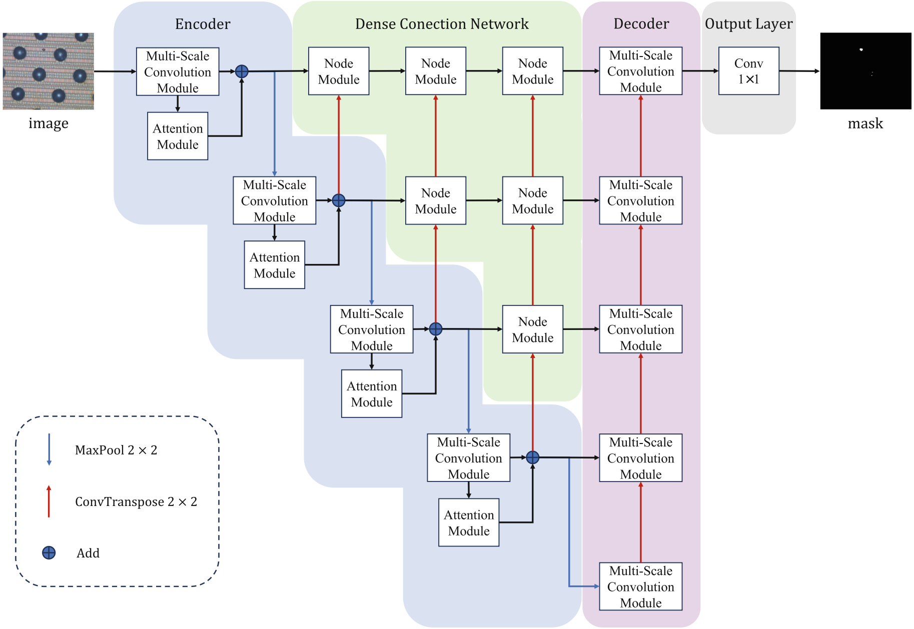


Fig. 1. The overview architecture of the proposed Multi-Scale Dense Network (MSD-Net).

3 Method

In this paper, we propose a novel multi-scale dense network (MSDNet) for defect segmentation, which incorporates multi-scale convolution modules, attention modules, and a dense connection network composed of node modules. This section provides a detailed exposition of the chip surface defect segmentation model, including the overall scheme and each constituent component of the framework.

3.1 Architecture

We propose MSDNet for chip surface defect detection and pixel-level defect segmentation. As illustrated in Fig. 1, the network incorporates an encoder, decoder, dense connection network, and output layer. The encoder consists of multi-scale convolution modules and attention modules, while the decoder only includes multi-scale modules. The dense connection network is composed of node modules, and the output layer comprises a convolution layer with a kernel size of 1 × 1.

The encoding process begins with the input image, producing rough feature maps via the multi-scale convolution module. Refined feature maps are generated by passing through the attention module and adding the results to the corresponding pixels of the original feature maps. Subsequently, downsampling reduces the size of the feature maps by half. After repeating the above process four times, the decoding process is performed. The decoder comprises five multi-scale convolution modules and four upsampling processes. The feature maps are expanded to twice their size through deconvolution. Finally, the network maps the feature vector to the output layer using a 1 × 1 convolution layer. During the encoding and decoding processes, a densely connected network is constructed to better integrate information between different layers, promoting cross-layer information integration. In this module, the high-dimensional feature maps expand to twice their size through deconvolution and concatenate with low-dimensional feature maps. These combined feature maps are then input into the node module together for feature extraction and fusion.

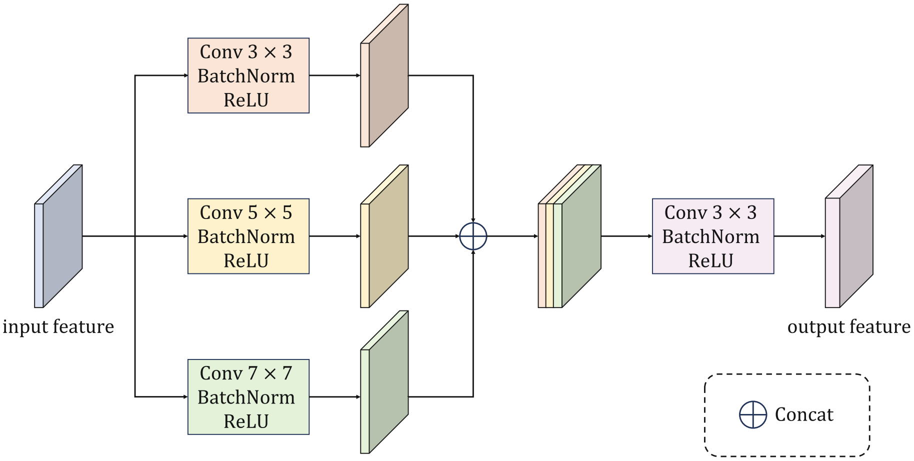


Fig. 2. The structure of the multi-scale convolution module is designed to capture features at various scales, enhancing the model's ability to detect defects of different sizes and shapes.

3.2 Multi-scale Convolution Module

Given the diversity of chip surface defects and the significant size differences between samples, the proposed multi-scale convolution module serves as the fundamental block in the encoding and decoding processes. As shown in Fig. 2, this module comprises parallel convolution layers with kernel sizes of 3 × 3, 5 × 5, and 7 × 7. Utilizing multiple parallel branches, convolution kernels of different sizes enable feature extraction and processing of information at various scales. Larger kernels capture global information, while smaller kernels focus on local details. The features of different scales are fused through concatenation, and semantic information and feature representation are enhanced through an additional 3 × 3 convolution. This structure helps the network adapt to targets of varying scales and sizes, enhancing its effective recognition and segmentation of objects of different sizes. Additionally, as the depth of the network increases and the number of convolutional layers grows, this module preserves the detailed information on the feature maps, benefiting the segmentation of small-scale defects.

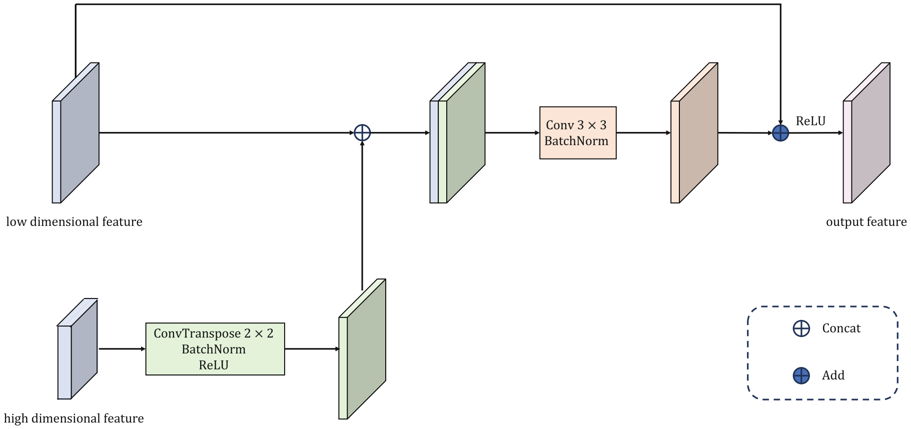


Fig. 3. The structure of the node module is designed to facilitate efficient feature extraction and fusion within the dense connection network.

3.3 Node Module

Due to the lack of information correlation between different levels of the image segmentation model based on the encoder-decoder framework, the network cannot effectively learn global context information. To address this issue and enhance global context learning, we propose a dense connection network composed of node modules to fuse feature maps in the encoding and decoding processes. Specifically, the high-dimensional feature maps, which have been downsampled and convolved, are doubled in size through deconvolution. These enlarged feature maps are then concatenated with the low-dimensional feature maps along the channel dimension and input into the node module for feature extraction and fusion. As shown in Fig. 3, the node module comprises convolution and residual connections. The feature maps, after channel fusion, undergo a 3×3 convolution, followed by pixel-wise addition with the original feature maps to produce the output feature maps. This structure leverages multi-branch networks and feature maps at different levels, obtaining broader context information through multi-level feature extraction and fusion. The resulting richer contextual information and more discriminative features improve the model's ability to represent both global and detailed information, thereby enhancing the performance of defect segmentation in complex scenes.

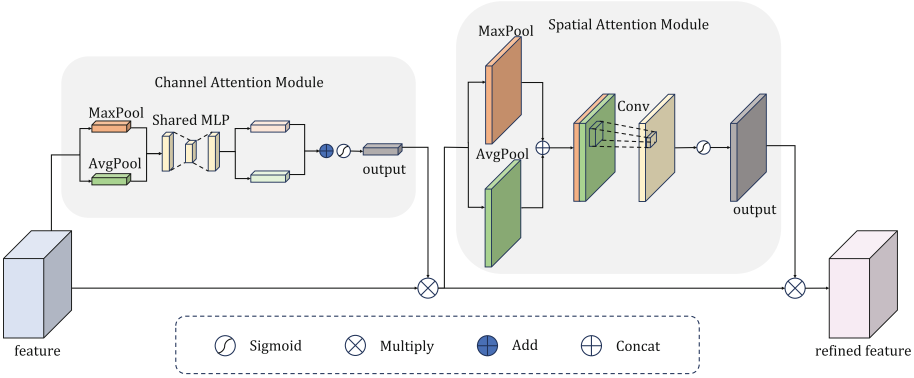


Fig. 4. The structure of the attention module is designed to selectively emphasize important features while suppressing less relevant information, thereby enhancing the network's focus on critical regions.

3.4 Attention Module

During the encoding process, information compression and loss often result in blurred key details such as defect boundaries. To address this issue, we employ the Convolutional Block Attention Module (CBAM) [24], which combines spatial and channel attention modules (as shown in Fig. 4). Feature maps from the multi-scale convolution module first pass through the channel attention module, producing weighted results. These results then pass through the spatial attention module, resulting in the final feature maps. The attention module directs the network to focus on defect information by increasing the weight of defect areas, suppressing responses in the background and irrelevant areas, and enhancing the network's ability to perceive defects.

3.5 Loss Function

Binary Cross Entropy (BCE) [27] loss is a loss function commonly used in binary classification. It is defined as

$$L_{\mathrm{BCE}} = -\frac{1}{N}\sum_{i}^{N}(y_i log(\sigma(y_i^*)) + (1-y_i)log(1-\sigma(y_i^*))) \tag{1}$$

where y_i represents the true label, y_i^* represents the precited labels, and N represents the number of samples. σ is the sigmoid activation function.

Dice loss [16] is a loss function that measures the similarity between the prediction results and the true labels. It is widely used in image segmentation. The Dice loss is defined as:

$$L_{\text{Dice}} = 1 - \frac{2\sum_i^N (y_i y_i^* + smooth)}{\sum_i^N y_i + \sum_i^N y_i^* + smooth} \tag{2}$$

where *smooth* is an adjustable parameter introduced to avoid the denominator being zero.

Focal loss [12] introduces coefficient factors to adjust the contribution of samples to the loss, aiming to improve the learning process, particularly for challenging samples. This adjustment helps alleviate issues related to the imbalance between positive and negative samples. The Focal loss is defined as:

$$L_{\text{Focal}} = -\alpha(1 - p_t)^{\gamma} log(p_t) \tag{3}$$

where α is the balance factor, which controls the balance of positive and negative samples. γ is the focus parameter, used to adjust the weight of the challenging samples. $p_t = exp(-L_{\text{BCE}})$ represents the probability of easy samples.

We employ a weighted combination of BCE loss, Dice loss, and Focal loss as the loss function to heighten the model's focus on challenging samples and address the problems present in the chip defect dataset, including imbalances in positive and negative samples and the non-uniform distribution of defect sizes. The loss function is defined as:

$$L = \beta_1 L_{\text{BCE}} + \beta_2 L_{\text{Dice}} + \beta_3 L_{\text{Focal}} \tag{4}$$

where β_1, β_2, and β_3 are the weight parameters, which can be adjusted.

4 Experiments

4.1 Dataset

We create a chip defect dataset comprising a total of 2,500 chip defect images, each with dimensions of 1224 × 1024 pixels. This dataset encompasses foreign matter defects present on the chip substrate.

With the development of technology, intelligent annotation software has greatly improved the efficiency of semantic annotation of images. In the defective image annotation phase, we first use an annotation tool [7] to mark defects on chip images, producing a preliminary mask. Then, manual modifications are applied at the pixel level to refine the mask, resulting in the final ground truth. Fig. 5 provides illustrative examples from the chip defect dataset.

In the experiment, we use the constructed chip surface defect dataset to verify our method. The dataset is divided into a training set and a test set at a ratio of 8:2. Specifically, 2,000 images are allocated to the training set, and the test set comprises 500 images.

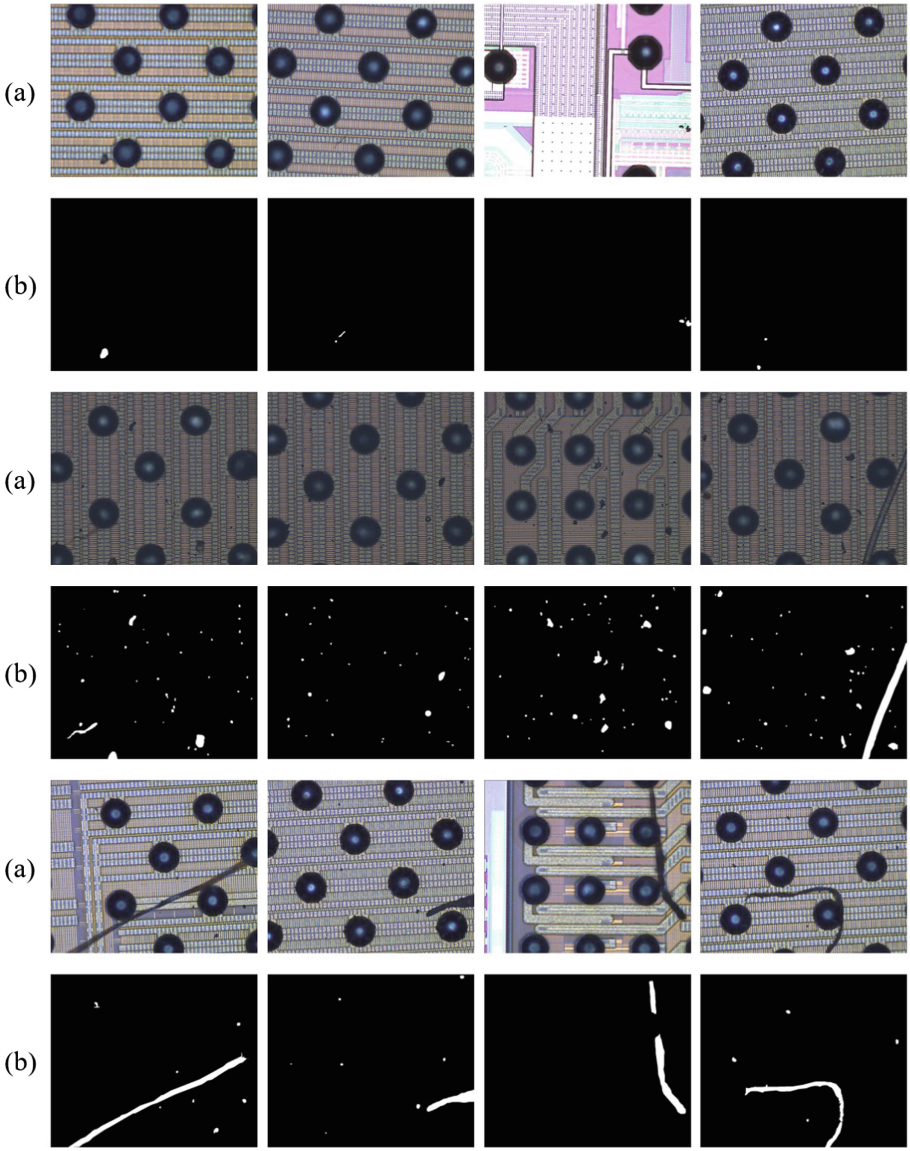


Fig. 5. Examples of the chip defect dataset. (a) Defect Image and (b) Ground Truth.

4.2 Implementation Details

The algorithm is implemented using PyTorch. The hardware configuration employed is the NVIDIA GeForce RTX 4090, and the software environment is Ubuntu 20.04 with CUDA version 12.2. The total number of training epochs is 150 with a batch size of 1. The initial learning rate is set at 0.01, and a fixed step decay strategy is adopted for learning rate decay. Specifically, the learning rate diminishes to 90% of its original value every 100 epochs. The optimizer adopts stochastic gradient descent (SGD) with a momentum coefficient of 0.9. The value of the smoothing parameter is e^{-8}. The value of α is 0.25, and γ is 2. The adjustable parameters β_1, β_2, and β_3 within the loss function are set to 0.4, 0.6, and 0.6, respectively.

4.3 Experimental Results

We compare our proposed MSDNet with existing methods on the chip defect dataset. As illustrated in Table 1, the experimental results reveal the superior performance of our MSDNet in chip defect segmentation. Specifically, when the confidence level exceeds 0.9, our MSDNet, verified using the chip defect dataset, exhibits improvements in segmentation accuracy by 44.35%, 6.43%, 6.16%, and 5.70%, compared to DeepLabv1, SegNet, Mobile-Unet, and U-Net, respectively. Additionally, the mIOU shows improvements of 33.76%, 4.68%, 2.92%, and 0.80%, respectively.

Furthermore, examples from the comparative experiment results, as shown in Fig. 6, demonstrate that our MSDNet achieves more accurate segmentation in the presence of multi-scale complex defects. Both large-size defects and tiny defects that only occupy a few pixels are segmented effectively. Moreover, compared to other networks, our model exhibits superior accuracy in extracting detailed defect information.

Table 1. Comparative experimental results. mAP stands for mean average precision, and mIOU stands for mean intersection over union.

Model	mAP(%) ↑	mIOU(%) ↑
DeepLabv1	40.66	39.39
SegNet	78.58	68.47
Mobile-UNet	78.85	70.23
U-Net	79.31	72.35
Ours	**85.01**	**73.15**

4.4 Ablation Study

We perform a comparative analysis of the baseline model, models with each of the three key modules individually (the multi-scale convolution modules, attention modules, and the dense connection network composed of node modules), and our full model (MSDNet).

The results are presented in Table 2. The experimental findings indicate that, in terms of chip defect segmentation performance, the collective utilization of all

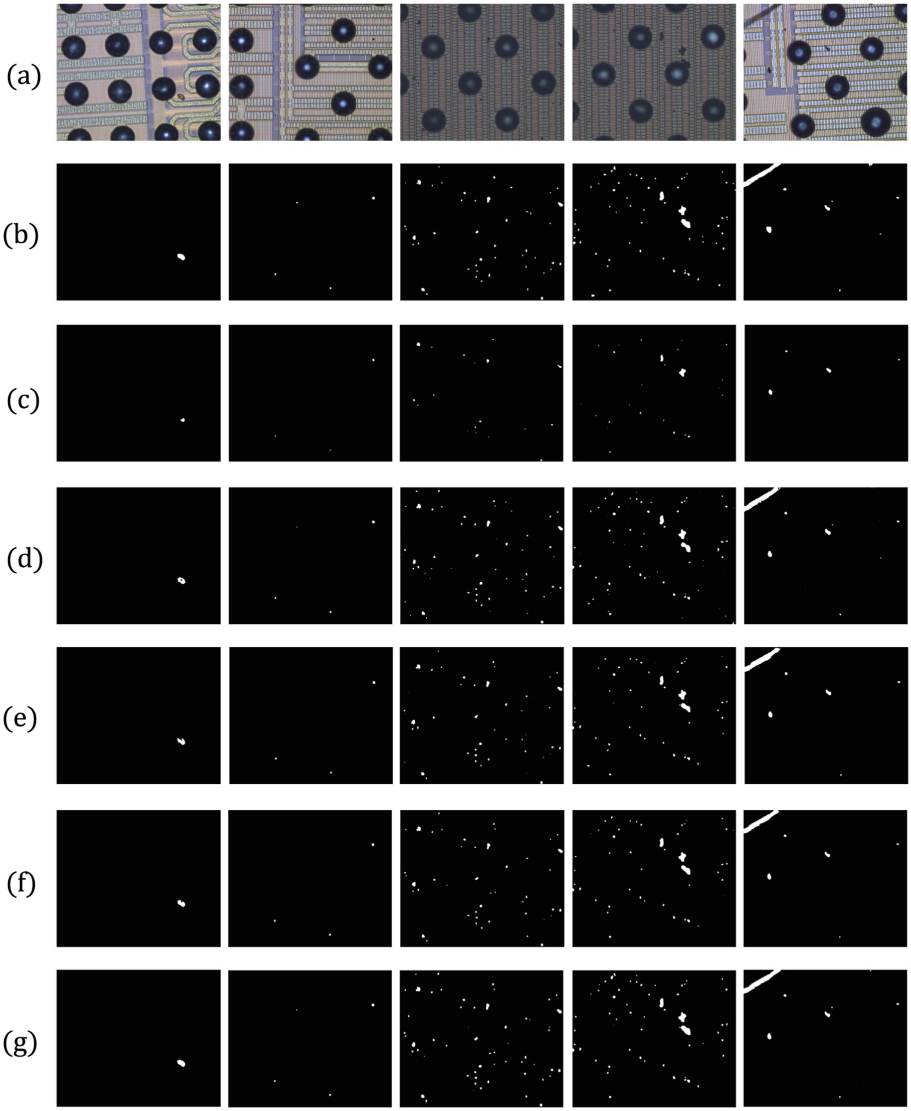


Fig. 6. Examples comparing the experimental results of different models. (a) Defect Image, (b) Ground Truth, (c)-(f) The prediction result of DeepLabv1, SegNet, Mobile-Unet and U-Net, respectively, and (g) The prediction result of ours.

three modules significantly outperforms alternative configurations. Compared to the baseline, the joint utilization of the three modules results in an increase of 9.2% in mAP and 5.33% in mIOU.

Table 2. Ablation experimental results. Baseline refers to the network that removes multi-scale convolution modules, attention modules and node modules. MSC, AM and NM stand for Multi-Scale Convolution Module, Attention Module and Node Module, respectively.

Baseline	MSCM	AM	NM	mAP(%) ↑	mIOU(%) ↑
√				75.81	67.82
√	√	√		83.17	71.79
√	√		√	79.08	68.93
√		√	√	79.72	72.40
√	√	√	√	**85.01**	**73.15**

5 Conclusion

In this paper, we propose a Multi-Scale Dense Network (MSDNet) based on the encoder-decoder structure, integrating multi-scale convolution modules, attention modules, and a dense connection network composed of node modules. We create a chip defect dataset and conduct an extensive series of experiments on this dataset. The experimental results demonstrate the efficacy of the proposed method in enhancing the segmentation performance of multi-scale and micro-scale defects on large-scale, high-resolution images. Furthermore, our method excels in accurately delineating defect edges and capturing detailed information. Comparative analysis with other methods, as well as ablation analysis shows a significant improvement in the precision of defect segmentation afforded by the proposed approach.

Acknowledgements. This work was supported in part by the National Key R&D Program of China (2018AAA0102801 and 2018AAA0102803), and in part of the National Natural Science Foundation of China (61772424, 61702418, and 61602383), and in part of the Air Science Foundation (2018ZE53052),

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
2. Cheon, S., Lee, H., Kim, C.O., Lee, S.H.: Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. IEEE Trans. Semicond. Manuf. **32**(2), 163–170 (2019)
3. Fang, J., Tan, X., Wang, Y.: Acrm: Attention cascade r-cnn with mix-nms for metallic surface defect detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 423–430. IEEE (2021)
4. Fang, Z., Wang, X., Li, H., Liu, J., Hu, Q., Xiao, J.: Fastrecon: Few-shot industrial anomaly detection via fast feature reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17481–17490 (2023)

5. Hao, K., Chen, G., Zhao, L., Li, Z., Liu, Y., Wang, C.: An insulator defect detection model in aerial images based on multiscale feature pyramid network. IEEE Trans. Instrum. Meas. **71**, 1–12 (2022)
6. Hittawe, M.M., Muddamsetty, S.M., Sidibé, D., Mériaudeau, F.: Multiple features extraction for timber defects detection and classification using svm. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 427–431. IEEE (2015)
7. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
8. Lei, J., Hu, X., Wang, Y., Liu, D.: Pyramidflow: High-resolution defect contrastive localization using pyramid normalizing flow. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14143–14152 (2023)
9. Li, C.L., Sohn, K., Yoon, J., Pfister, T.: Cutpaste: Self-supervised learning for anomaly detection and localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9664–9674 (2021)
10. Lin, B.S., Cheng, J.S., Liao, H.C., Yang, L.W., Yang, T., Chen, K.C.: Improvement of multi-lines bridge defect classification by hierarchical architecture in artificial intelligence automatic defect classification. IEEE Trans. Semicond. Manuf. **34**(3), 346–351 (2021)
11. Lin, D., Li, Y., Prasad, S., Nwe, T.L., Dong, S., Oo, Z.M.: Cam-unet: Class activation map guided unet with feedback refinement for defect segmentation. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 2131–2135. IEEE (2020)
12. 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)
13. Liu, T., He, Z.: Tas 2-net: Triple-attention semantic segmentation network for small surface defect detection. IEEE Trans. Instrum. Meas. **71**, 1–12 (2022)
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016)
15. 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)
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. 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)
18. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)

21. Rudack, A.C., Kong, L.W., Baker, G.G.: Infrared microscopy for overlay and defect metrology on 3d-interconnect bonded wafers. In: 2010 IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC). pp. 347–352. IEEE (2010)
22. Sari, L., Ertüzün, A.: Texture defect detection using independent vector analysis in wavelet domain. In: 2014 22nd International Conference on Pattern Recognition. pp. 1639–1644. IEEE (2014)
23. Wang, T., Zhang, C., Ding, R., Yang, G.: Mobile phone surface defect detection based on improved faster r-cnn. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 9371–9377. IEEE (2021)
24. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
25. Xie, Q., Li, D., Xu, J., Yu, Z., Wang, J.: Automatic detection and classification of sewer defects via hierarchical deep learning. IEEE Trans. Autom. Sci. Eng. **16**(4), 1836–1847 (2019)
26. Yang, L., Zhou, F., Wang, L.: A scratch detection method based on deep learning and image segmentation. IEEE Trans. Instrum. Meas. **71**, 1–12 (2022)
27. Yi-de, M., Qing, L., Zhi-Bai, Q.: Automated image segmentation using improved pcnn model based on cross-entropy. In: Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004. pp. 743–746. IEEE (2004)
28. 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 (CVPR) (July 2017)

Task Oriented Image Quality Assessment for Synthesized Images

Ke Xu[1], Qian Zhang[1,2](✉), Fei Yang[1], Zhanghao Jiang[1], and Boon-Giin Lee[1]

[1] University of Nottingham Ningbo China, Ningbo, China
{ke.xu,qian.zhang,fei.yang,zhanghao.jiang, boon-giin.lee}@nottingham.edu.cn
[2] Nottingham Ningbo China Beacons of Excellence Research and Innovation Institute, Ningbo, China

Abstract. In this study, we propose a new general learning-based framework, named *Task-Oriented Image Quality Assessment*, for evaluating the performance of Reference-guided image synthesis (RIS) tasks. Our framework uniquely employs both source and target images to construct content- and style-encoded feature embeddings, and then evaluates the quality of the synthesized images by comparing their feature distances to those of the source and target images. We designed a two-branch network that embeds both content and style elements simultaneously. Our training process uses a style-level interpolation strategy to generate intermediate styled images for training, eliminating the need for human annotations. The quality score is calculated using a ratio-based distance that considers both the synthesized image *from* the source image and *to* the target image. Our method was evaluated using the HIDER dataset and RESIDE dataset, which provide subject scores for each image. The obtained result shows the efficiency of our method.

Keywords: Image quality assessment · Reference-guided image synthesis · Two-branch networks

1 Introduction

Reference-guided image synthesis (RIS) aims to change the style of a source image to match the style of a target image while preserving the source image's structural information. Many computer vision problems can be formulated as RIS tasks and solved by Generative Adversarial Networks [5] based supervised learning given paired training data, like semantic image synthesis [19], colorization [13], sketch to photos [11], super resolution [14,21]. For example, in Fig.1, indoor design effect rendering task, the plain image of an interior design is treated as the source image, while the fully rendered target image is used as the reference image to guide the image synthesis. Also, in Fig.2, image dehazing task, the hazing image is treated as the source image, the clear image(ground truth) is used

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 140–153, 2025.
https://doi.org/10.1007/978-3-031-78389-0_10

as target image to guide the image synthesis. A good synthesized image needs to retain the underlying spatial structure of the source image while matching the target image in terms of color, texture and lighting conditions. To further facilitate RIS research, it is keen to develop Image Quality Assessment metrics to quantitatively evaluate the synthesized image with human preference.

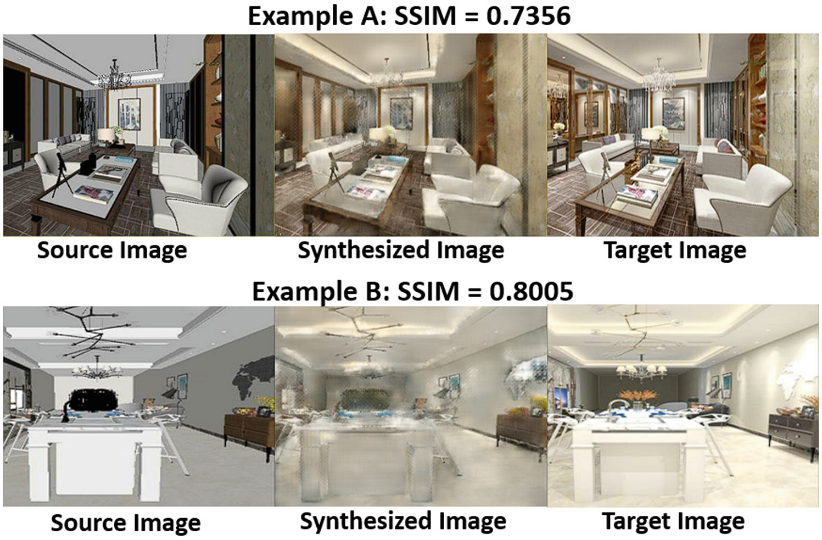


Fig. 1. Two examples of source image, synthesized image, and target image triplets. Example A has an acceptable synthesized result despite having a low SSIM score, while example B has a poor synthesis outcome but has a higher SSIM score.

Although several signal-based evaluation metrics have been proposed, e.g., SSIM [22], PSNR [8], FSIM [27], MS-SSIM [24], and IW-SSIM [23], they mainly focus on *pixel* differences and aim for evaluating distorted images, such as different types of noise and compression levels. However, these methods either focus on semantically irrelevant image components or lack emphasis on specific image characteristics. As seen in Fig.1 and Fig.2, SSIM and PSNR scores do not align with human preference and fail to evaluate synthesized images at a perceptual level.

While several sample-based GAN evaluation metrics, such as Inception Score [1], Wasserstein distance and Frechet Inception Distance [7], have been proposed, these metrics mainly focus on assessing the *overall* generated image distribution rather than evaluating *single* synthesized images. Consequently, these sample-based metrics fail to capture the quality of single synthesized images, missing critical details about the specific quality of individual images, and are thus not robust for evaluating the quality of single synthesized outputs.

Furthermore, there is a growing trend in utilizing network-based approaches to address the challenges of image quality assessment. Such as DeepIQA [2],

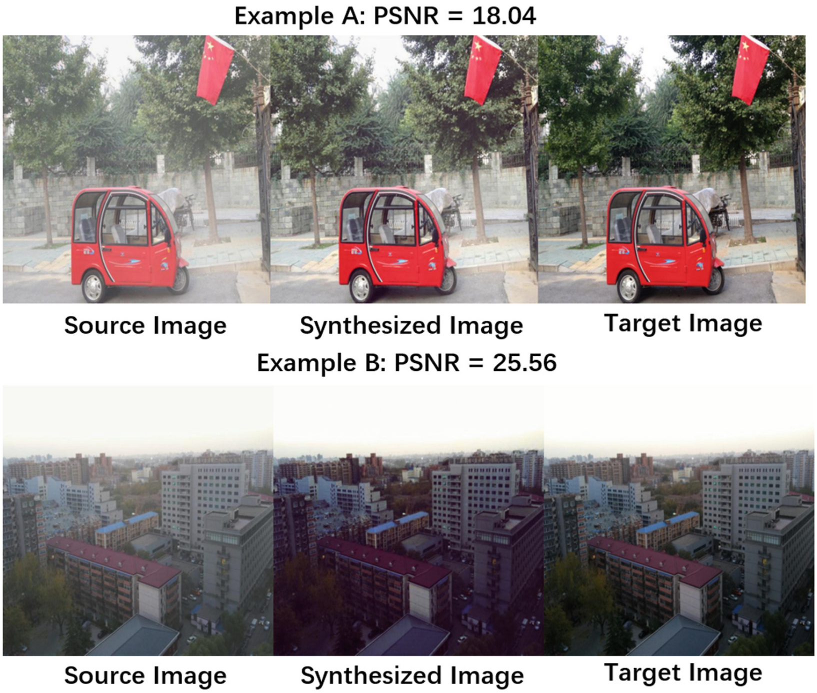


Fig. 2. Two examples of source image, synthesized image, and target image triplets. Example A has an acceptable synthesized result despite having a low PSNR score, while example B has a poor synthesis outcome but has a higher PSNR score.

LPIPS [28], AHIQ [10],CKDN [29],RISA [6] and CLIP-IQA [20]. These methods employ well-trained neural networks to encode semantic features for evaluation purposes. However, it is worth noting that these methods primarily focus on translating a synthesized image *into* a target image, while ignoring the importance of translating *from* a source image. This latter aspect is crucial in RIS assessments, as previously mentioned.

We believe that an effective IQA metric for RIS tasks should include two components: the ability to quantify the *style* similarity between a single synthesized image and its corresponding target image in different domains, as well as the ability to evaluate the preservation of underlying content against the source image. The proposed Task-Oriented Image Quality Assessment framework is a learning-based approach that combines content and style information to effectively evaluate the quality of synthesized images. The two-branch network embeds content and style features, using a content reconstruction branch to ensure content information is accurately captured and a style encoding branch that is trained using a semi-supervised style level classification task. To avoid individual bias and without manually labelled data, a style-level interpolation strategy is also introduced. The final quality score is calculated using a ratio-based distance that takes into account both the content preservation and style similarity between the source, synthesized, and target images. Experimental results show that the proposed method is more consistent with human preferences and on par with state-of-the-art IQA metrics for traditional distortion image evaluation tasks.

2 Related Work

2.1 Reference-guided image synthesis (RIS)

Reference-guided image synthesis has emerged as a significant area of research within computer vision, leveraging reference images to guide the generation of new images. This approach typically combines elements from reference images to create high-quality, specific style and content consistency outputs. In the RIS task, the underlying spatial structure is referred to as content, whereas style refers to the distinctive appearance of an image, which can be evaluated in terms of, but not limited to, hairstyle, gender, texture, reflectivity, and lighting conditions.

U-Net [17], originally proposed by Ronneberger et al. for biomedical image segmentation, has become a cornerstone in the field due to its unique architecture that facilitates precise image synthesis. In the context of reference-guided image synthesis, U-Net has been adapted to incorporate reference images effectively. Image-to-Image Translation with Conditional Adversarial Networks (Pix2Pix) [9] is a framework that learns the mapping from input images to output images using a conditional GAN. The conditional GAN architecture ensures that the generated images adhere closely to the reference images' content and style, showcasing the importance of reference-guided synthesis in practical applications. StarGAN [3] and StarGANv2 [4], enables multi-domain image-to-image translation using a single model, allowing for diverse image transformations guided by domain labels and enhances this capability by supporting continuous and diverse style control, achieving more flexible and high-quality image synthesis across multiple domains. CAPS [25] proposed a novel capsule based conditional generative adversarial network that can automatically synthesize an indoor image as target iamge with realistic and aesthetically pleasing rendering effect from a given reference image rendered without any effects from a interior designed 3D model. Densely Connected Pyramid Dehazing Network (DCPDN) [26] is a cutting-edge approach designed to tackle the problem of image dehazing. The proposed method takes hazing images as reference image, synthesizing the dehazing image. This method leverages a densely connected architecture to enhance feature reuse and gradient flow, which significantly improves the network's performance and efficiency.

2.2 Image Quality Assessment(IQA)

The IQA methods can be calssify into three categories according to the availability of reference:

1)Full-reference IQA: Full-reference Image Quality Assessment (FR-IQA) is a technique used to evaluate the visual quality of an image by comparing it to a reference image known to be of high quality. The most representative methods are the following SSIM [22], PSNR [8], FSIM [27], MS-SSIM [24], and IW-SSIM [23], which mainly focus on pixel differences between reference image and distorted images.

2)Reduce-reference IQA: Reduced-reference Image Quality Assessment (RR-IQA) is a method for evaluating the visual quality of an image using partial information from a reference image. This approach involves extracting and comparing key features from both the test and reference images, allowing for effective quality assessment while requiring significantly less reference data than full-reference IQA. Reduced-Reference Entropic Differencing (RRED) [18] evaluates image quality by measuring the difference in entropy between features of the test and reference images.

3)No-reference IQA: No-reference Image Quality Assessment (NR-IQA) is a method for evaluating the visual quality of an image without the reference image. Early works likes BRISQUE [15] and NIQE [16] have laid the foundation for NR-IQA by utilizing natural scene statistics (NSS) to model the image quality. BRISQUE focuses on measuring the deviations from natural scene statistics in the spatial domain, while NIQE models these deviations without relying on human-rated training data, making it a more generalized approach. Recent advancements in NR-IQA leverage deep learning techniques to enhance the accuracy and robustness of quality predictions. The pioneering approach, DeepIQA [2], employs deep learning models to automatically evaluate the quality of images. Another metric called LPIPS [28] quantifies the perceptual similarity between two images by leveraging a trained deep neural network. AHIQ [10] compares images at the patch level, enhances spatial details, and assigns scores to individual patches while considering their interdependencies. CKDN [29] utilizes degraded images as references for assessing image quality, demonstrating its effectiveness in evaluating GAN-generated images and providing insights into evaluating GAN-based models. RISA [6] captures the style similarity between generated and reference images and also employs unsupervised contrastive loss to enhance assessment. CLIP-IQA [20], on the other hand, leverages Contrastive Language-Image Pre-training models to assess both perceived quality and abstract perception of images in a zero-shot manner without explicit training using labeled data from user studies.

3 Methodology

Given the ground truth image pair I_s (source image) and I_t (target image), the goal is to estimate the quality of synthesized image I_g generated by different RIS approaches. To address this, our proposed method utilizes a two-branch network to encode both content and style features. The content reconstruction branch ensures preservation of structural information, while the style encoding branch captures style variation. As depicted in Fig.3, using both source and

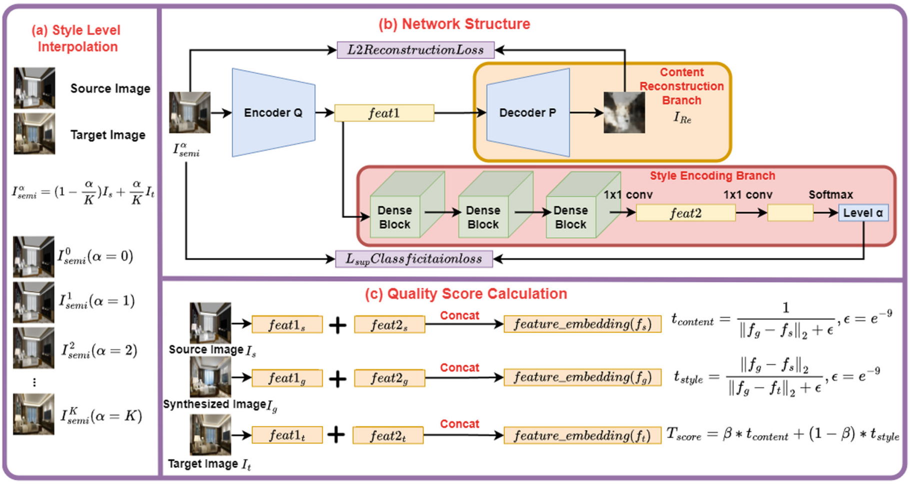

(a) Style Level Interpolation
Source Image
Target Image
$I^{\alpha}_{semi} = (1-\frac{\alpha}{K})I_s + \frac{\alpha}{K}I_t$
$I^0_{semi}(\alpha=0)$
$I^1_{semi}(\alpha=1)$
$I^2_{semi}(\alpha=2)$
⋮
$I^K_{semi}(\alpha=K)$
(b) Network Structure
$L2ReconstructionLoss$
I^{α}_{semi}
Encoder Q
$feat1$
Decoder P
Content Reconstruction Branch
I_{Re}
Style Encoding Branch
Dense Block
Dense Block
Dense Block
1x1 conv
$feat2$
1x1 conv
Softmax
Level α
$L_{sup}Classficitaionloss$
(c) Quality Score Calculation
Source Image I_s
$feat1_s$
$feat2_s$
Concat
$feature\_embedding(f_s)$
$t_{content} = \frac{1}{\|f_g - f_s\|_2 + \epsilon}, \epsilon = e^{-9}$
Synthesized Image I_g
$feat1_g$
$feat2_g$
Concat
$feature\_embedding(f_g)$
$t_{style} = \frac{\|f_g - f_s\|_2}{\|f_g - f_t\|_2 + \epsilon}, \epsilon = e^{-9}$
Target Image I_t
$feat1_t$
$feat2_t$
Concat
$feature\_embedding(f_t)$
$T_{score} = \beta * t_{content} + (1-\beta) * t_{style}$

Fig. 3. Overview of architecture of the proposed method. Panel (a) illustrates the style level interpolation process. Panel (b) depicts the network architecture, which comprises two branches to encode both content and style features. Panel (c) outlines the quality score calculation procedure.

target images as references, the translation score T_{score} is calculated as the ratio of feature embedding distance between the synthesized and reference images.

3.1 Style Level Interpolation for Data Preparation

To ensure successful two-branch training, it is important to prepare the dataset in a way that preserves the image content while adding texture, lighting effects, and other styles to the source image. Therefore, we propose a technique for style-level interpolation that generates semi-translated images I^{α}_{semi} with different levels of style with:

$$I^{\alpha}_{semi} = (1 - \frac{\alpha}{K})I_s + \frac{\alpha}{K}I_t, \tag{1}$$

The level of style-based image translation $\alpha \in [0, K]$ determines the semi-translation ratio and the corresponding semi-translated image is denoted as I^{α}_{semi}.

K refers to the total number of semi-translated image levels. The level is linearly correlated with both the source image I_s and the target image I_t, with $\alpha = 0$ and $\alpha = K$ denoting the source and target image, respectively. This technique allows us to create a dataset that reflects the style variance in a quantifiable manner. In our experiments, we set $K = 10$. In different datasets, it can be set to different values as needed.

3.2 Learning-based Quality Score Estimation

Feature Embeddings The content reconstruction branch of our network is designed to reconstruct the input image I_{semi}^{α} with an auto-encoder structure [9]. Specifically, the encoder takes the input image and extracts the feature embedding, $feat1$, from its final layer. The decoder then reconstructs this feature embedding into the reconstructed image I_{Re}, supervised by the L_2 reconstruction loss.

In addition to the content reconstruction branch, the network includes a style encoding branch, which is designed to classify the style level α of the input image I_{semi}^{α}. To do this, it uses a classification network with three dense blocks [26] and two fully connected layers followed by a softmax layer. The second last layer activation serves as the style feature embedding $feat2$, which is trained in a weakly supervised classification manner using auto-labeled intermediate images I_{semi}^{α}. By emphasizing style-related features, this branch is able to accurately classify the style level of the input image and retain a substantial amount of style information.

Finally, to create a single feature embedding f that balances both the content and style of the input image, we concatenate two normalized feature embeddings, $feat1$ and $feat2$ as shown in Eq.(2):

$$f = Concat(m * feat1, (1 - m) * feat2) \tag{2}$$

where $m \in (0, 1)$ is the hyperparameter used to balance the weights between $feat1$ and $feat2$ ensuring that neither aspect of the image is overemphasized. A higher value of m can result in feature embedding f that is more focused on content-based features, while a lower value of m can result in feature embedding f that is more focused on style-based features. In our experiment, we set the default value of m to 0.5.

Quality Score Formulation Our proposed two-branch network training creates a task-oriented, content-style bounded space by considering both the source and target images. We evaluate the quality of the synthesized image by calculating two scores: the content-related quality score, $t_{content}$, which is determined by comparing the synthesized image to the source image; and the style-related quality score, t_{style}, which is calculated by determining the ratio of the distance between the synthesized image and both the source and target images in the style-encoded space.

To evaluate the performance of the image translation task, which translates I_s into I_t, we measure how closely the synthesized image I_g follows the mapping from f_s to f_t. This mapping is obtained by passing the images I_s, I_g, and I_t through the pre-trained network, and extracting their respective feature embeddings f_s, f_g, and f_t as described in Eq.(2). We then calculate the the content-related quality score as

$$t_{content} = \frac{1}{\|f_g - f_s\|_2 + \epsilon}, \epsilon = e^{-9} \tag{3}$$

where $\|\cdot\|_2$ denotes the Euclidean norm and ϵ is a small constant to avoid the fraction infinite.

And then, we calculate the style-related score by the ratio of mappings between $f_g \to f_s$ and $f_g \to f_t$ as given by:

$$t_{style} = \frac{\|f_g - f_s\|_2}{\|f_g - f_t\|_2 + \epsilon}, \epsilon = e^{-9} \tag{4}$$

Finally, the overall score T_{score} is obtained by combining the two aspect scores with a weighted factor $\beta \in (0, 1)$ by:

$$T_{score} = \beta * t_{content} + (1 - \beta) * t_{style} \tag{5}$$

The weighted factor β regulates the participation ratio of the content measure $t_{content}$ and the style measure t_{style}, allowing the quality score formulation to be applied across a wide range of tasks using different β values. This allows for a more flexible and comprehensive evaluation of the image translation task. In our experiments, the $\beta = 0.5$ produces the best results among different settings.

3.3 Training Objective

Our training objective consists comprises two key components: 1) a reconstruction loss applied in the content reconstruction branch to minimize the difference between the decoded image and the input image and 2) a weakly supervised classification loss applied in the style encoder branch to differentiate the style levels of the input interpolated images.

Reconstruction Loss The reconstruction loss is ensures that the decoded image closely matches the input image. We implement the reconstruction Loss by utilizing the L_2 norm to ensure that the decoded image I_{Re} closely approximates the input image I^{α}_{semi}. This approach constrains the content embedding $feat1$ to retain substantial content-related structural information. This is given by:

$$L_2(I^{\alpha}_{semi}, I_{Re}), \tag{6}$$

Classification Loss The classification loss, implemented using the $softmax$ function combine with cross entropy, ensures that the network's layer activation $feat2$ is sensitive to style variance, thereby enabling it to accurately capture style-related content. The loss is formulated as:

$$L_{sup}(p(I^{\alpha}_{semi}), t(I^{\alpha}_{semi})), \tag{7}$$

where $p(I^{\alpha}_{semi})$ and $t(I^{\alpha}_{semi}))$ denote the predicted and target interpolation levels, respectively.

Full Objective Our full objective is a weighted combination of the reconstruction and classification loss with:

$$L = \lambda L_2(I^{\alpha}_{semi}, I_{Re}) + L_{sup}(p(I^{\alpha}_{semi}), t(I^{\alpha}_{semi})) \tag{8}$$

where $\lambda = 3$ shows best result in our experiments.

4 EXPERIMENT

4.1 Dataset

Our experiments were conducted on the HIDER [25] and RESIDE [12] datasets. Home Interior Design Effect Rendering dataset(HIDER) consists of 238 pairs of plain and rendered images along with corresponding reference guide synthesized images. Plain images are unrendered interior design drawings as source images and rendered images are rendered images as target images with reference guide synthesized images generated by CAPS [25], U-net [17], and Pix2Pix [9]. Each image was evaluated with subjective scores assigned by human reviewers.

Similarly, the RESIDE dataset is contained 500 triplets of hazing images, dehazing image generated using DCPDN [26] and clear images, each also assigned subjective scores by human reviewers. In this task, we take hazing images as source images, clear images as target images, and dehazing images as synthesized images.

We selected the HIDER and RESIDE datasets for our experiments due to their unique and valuable characteristics. These datasets are containing comprehensive sets of source images, synthesized images, and corresponding target images. Additionally, both datasets include subjective scores assigned by human reviewers, providing an objective measure of image quality. This combination of features allows for a thorough evaluation of image quality assessment (IQA) methods, ensuring that our approach is rigorously tested against well-established benchmarks.

4.2 Protocol and Evalution criteria

Pearsons linear correlation coefficient (PLCC) and Spearmans rank order correlation coefficient (SROCC) are measures of the correlation between predicted values and subjective scores in IQA, with higher scores indicating better agreement between predicted values and subjective scores.

4.3 Performance Evalution

In this study, we conducted a comprehensive evaluation of various Image Quality Assessment (IQA) metrics on reference guide synthesized image evaluation tasks using the HIDER and RESIDE datasets. Table 1 shows the PLCC and SROCC results for both the HIDER and RESIDE datasets.

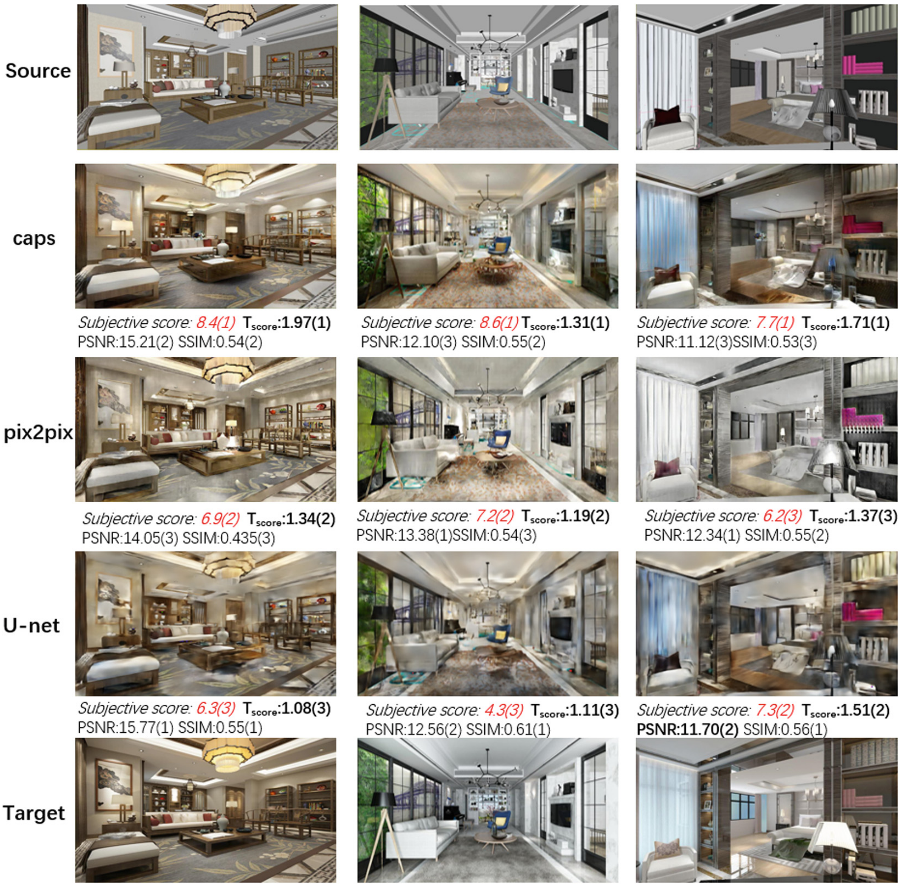

Source
caps
Subjective score: 8.4(1) Tscore:1.97(1)
PSNR:15.21(2) SSIM:0.54(2)
Subjective score: 8.6(1) Tscore:1.31(1)
PSNR:12.10(3) SSIM:0.55(2)
Subjective score: 7.7(1) Tscore:1.71(1)
PSNR:11.12(3)SSIM:0.53(3)
pix2pix
Subjective score: 6.9(2) Tscore:1.34(2)
PSNR:14.05(3) SSIM:0.435(3)
Subjective score: 7.2(2) Tscore:1.19(2)
PSNR:13.38(1)SSIM:0.54(3)
Subjective score: 6.2(3) Tscore:1.37(3)
PSNR:12.34(1) SSIM:0.55(2)
U-net
Subjective score: 6.3(3) Tscore:1.08(3)
PSNR:15.77(1) SSIM:0.55(1)
Subjective score: 4.3(3) Tscore:1.11(3)
PSNR:12.56(2) SSIM:0.61(1)
Subjective score: 7.3(2) Tscore:1.51(2)
PSNR:11.70(2) SSIM:0.56(1)
Target

Fig. 4. Some example results from the HIDER dataset. At the bottom of each synthesized image, the corresponding subjective score, T_{score}, PSNR and SSIM are displayed. The numerical value in parentheses indicates the ranking of the score among the three synthesized images generated by different methods. The IQA method that aligns with the ranking of the subjective score is denoted in bold font.

The proposed method outperforms the other state-of-the-art IQA metrics, as demonstrated by the highest PLCC and SROCC scores across both datasets. However, some network-based methodologies like DeepIQA, LPIPS and CKDN have been unsuccessful in accurately capturing human preferences. This failure can be attributed to their focus on translating an image *to* a target image, ignoring the translation *from* a source image. Additionally, in RESIDE dataset, LPIPS, NIQE, CLIP-IQA perform significantly worse than in HIDER, highlighting inconsistency issues. These metrics struggle to capture minor details affecting dehazing image quality when evaluating similarly styled images. Conversely, DeepIQA struggles to provide reliable assessments across both datasets.

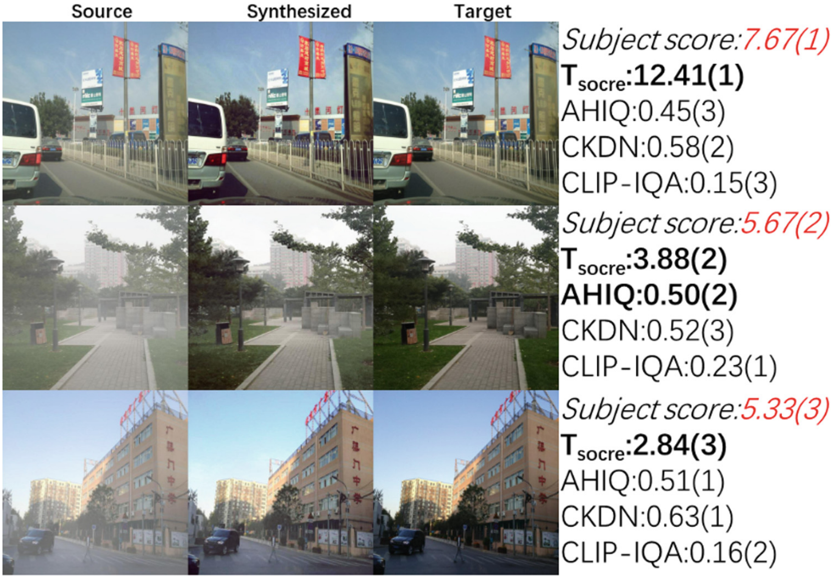


Fig. 5. Some example results from RESIDE dataset. Next to each set of three images, the corresponding subjective score, T_{score}, AHIQ, CKDN and CLIP-IQA are presented. The IQA method that aligns with the ranking of the subjective score is denoted in bold font.

Traditional signal-based methods, PSNR, SSIM, IW_SSIM and FSIM also show limited effectiveness in accurately assessing synthesized image quality, as reflected by their relatively lower correlation scores. These signal-based approaches have also encountered limitations in both experiments as they are primarily designed for naturally degraded images rather than synthesized images and are not sensitive to style variance.

The superior performance of the our proposed method T_{score} can be attributed to its innovative approach. This method combines a content reconstruction branch and a style encoding branch to capture both content and style variance, better aligning with human visual perception. This hybrid methodology ensures a more robust and accurate assessment of various reference guide synthesized image quality tasks.

Fig.4 and Fig.5 show a section of the experimental results obtained by evaluating images synthesized using the HIDER and RESIDE datasets. Both experiments demonstrate that our proposed T_{score} metric provides a more accurate evaluation of the perceptual quality of synthesized images compared to existing methods applied on HIDER and RESIDE datasets.

These two experiments show that our proposed method more fits the image synthesized task. Our method designed the content reconstruction branch that is pre-trained on the style level interpolation dataset in the encoder-decoder manner, which supervises the model to learn the feature of the image content

Table 1. Comparison with other IQA metrics on the synthesized image evaluation tasks: HIDER [25] and RESIDE [12]

IQA metrics	HIDER		RESIDE	
	PLCC	SROCC	PLCC	SROCC
PSNR [8]	0.3640	0.2672	0.3948	0.3419
SSIM [22]	0.3520	0.2999	0.3648	0.2886
IW_SSIM [23]	0.4579	0.4567	0.3365	0.2837
FSIM [23]	0.4086	0.3857	0.3606	0.2876
MS_SSIM [24]	0.4304	0.4151	0.4136	0.3217
DeepIQA [2]	0.1374	0.0942	0.2242	0.2519
LPIPS [28]	0.4984	0.4830	0.1655	0.1541
NIQE [16]	0.2012	0.1853	0.0011	0.0155
CKDN [29]	0.5069	0.4632	0.3950	0.3763
CLIP-IQA [20]	0.3707	0.3490	0.2322	0.2570
T_{score}	**0.5656**	**0.5627**	**0.5457**	**0.6249**

well. The style encoding branch for the style level estimation also contributes to the feature learning to classify the style, such as the light effect and texture.

4.4 CONCLUSION

In this study, we proposed a novel learning-based framework called Task-Oriented Image Quality Assessment for evaluating RIS tasks. Unlike existing methods, our method does not rely on manual data labeling during training but instead utilizes a style-level interpolation strategy to generate intermediate styled images as training data. Our results demonstrate that our method outperforms current image assessment metrics displaying higher consistency with human preferences across various synthesized image datasets. These findings suggest that our method holds considerable potential as a promising approach for evaluating RIS tasks. Future research can explore further improvements and applications of this framework in real-world scenarios.

References

1. Barratt, S., Sharma, R.: A note on the inception score. arXiv preprint arXiv:1801.01973 (2018)
2. Bosse, S., Maniry, D., Müller, K.R., Wiegand, T., Samek, W.: Deep neural networks for no-reference and full-reference image quality assessment. IEEE TIP **27**(1), 206–219 (2017)
3. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8789–8797 (2018)

4. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8188–8197 (2020)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS **27** (2014)
6. Guo, J., Du, C., Wang, J., Huang, H., Wan, P., Huang, G.: Assessing a single image in reference-guided image synthesis. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 753–761 (2022)
7. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS **30** (2017)
8. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of psnr in image/video quality assessment. Electron. Lett. **44**(13), 800–801 (2008)
9. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. pp. 1125–1134 (2017)
10. Lao, S., Gong, Y., Shi, S., Yang, S., Wu, T., Wang, J., Xia, W., Yang, Y.: Attentions help cnns see better: Attention-based hybrid image quality assessment network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1140–1149 (2022)
11. Lei, Y., Du, W., Hu, Q.: Face sketch-to-photo transformation with multi-scale self-attention gan. Neurocomputing **396**, 13–23 (2020)
12. Li, B., Ren, W., Fu, D., Tao, D., Feng, D., Zeng, W., Wang, Z.: Benchmarking single-image dehazing and beyond. IEEE TIP **28**(1), 492–505 (2018)
13. Li, H., Sheng, B., Li, P., Ali, R., Chen, C.P.: Globally and locally semantic colorization via exemplar-based broad-gan. IEEE Trans. Image Process. **30**, 8526–8539 (2021)
14. Liang, J., Zeng, H., Zhang, L.: Details or artifacts: A locally discriminative learning approach to realistic image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5657–5666 (2022)
15. Mittal, A., Moorthy, A.K., Bovik, A.C.: No-reference image quality assessment in the spatial domain. IEEE Trans. Image Process. **21**(12), 4695–4708 (2012)
16. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a "completely blind" image quality analyzer. IEEE Signal Process. Lett. **20**(3), 209–212 (2012)
17. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. Springer (2015)
18. Soundararajan, R., Bovik, A.C.: Rred indices: Reduced reference entropic differencing for image quality assessment. IEEE Trans. Image Process. **21**(2), 517–526 (2011)
19. Sushko, V., Schönfeld, E., Zhang, D., Gall, J., Schiele, B., Khoreva, A.: Oasis: only adversarial supervision for semantic image synthesis. Int. J. Comput. Vision **130**(12), 2903–2923 (2022)
20. Wang, J., Chan, K.C., Loy, C.C.: Exploring clip for assessing the look and feel of images. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2555–2563 (2023)
21. Wang, Y., Hu, Y., Yu, J., Zhang, J.: Gan prior based null-space learning for consistent super-resolution. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2724–2732 (2023)
22. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004)
23. Wang, Z., Li, Q.: Information content weighting for perceptual image quality assessment. IEEE TIP **20**(5), 1185–1198 (2010)

24. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: ACSSC, 2003. vol. 2, pp. 1398–1402. Ieee (2003)
25. Yang, F., Lu, Z., Qiu, G., Lin, J., Zhang, Q.: Capsule based image synthesis for interior design effect rendering. In: ACCV. pp. 183–198. Springer (2018)
26. Zhang, H., Patel, V.M.: Densely connected pyramid dehazing network. In: CVPR. pp. 3194–3203 (2018)
27. Zhang, L., Zhang, L., Mou, X., Zhang, D.: Fsim: A feature similarity index for image quality assessment. IEEE TIP **20**(8), 2378–2386 (2011)
28. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
29. Zheng, H., Yang, H., Fu, J., Zha, Z.J., Luo, J.: Learning conditional knowledge distillation for degraded-reference image quality assessment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10242–10251 (2021)

SANGAM: Synergizing Local and Global Analysis for Simultaneous WBC Classification and Segmentation

Adit Srivastava[1(✉)], Aravind Ramagiri[1], Puneet Gupta[1], and Vivek Gupta[2]

[1] Indian Institute of Technology, Indore, Indore, Madhya Pradesh, India
{mt2302101002,phd2301101006,puneet}@iiti.ac.in
[2] Thapar University, Patiala, Punjab, India
vivek.gupta1@thapar.edu

Abstract. Analyzing different white blood cell (WBC) classes is essential for human health monitoring, making accurate segmentation and classification crucial for diagnosing blood-related conditions. Existing WBC segmentation systems mainly rely on convolution neural networks (CNNs) and Transformers. Unfortunately, they are unable to simultaneously capture global context and local information. Similarly, existing WBC classification systems fail to appropriately focus on the relevant regions of WBC images. Additionally, the processes of WBC classification and segmentation are intertwined, but they are not properly synergized in the literature. These three issues have limited the efficacy of existing WBC segmentation and classification systems. Our proposed system, *SANGAM*, improves the efficacy of WBC segmentation and classification by addressing these issues. Specifically, it integrates the local learning capabilities of CNNs with the global context learning capabilities of Transformers to enhance WBC segmentation. It also improves WBC classification by providing more attention to the relevant areas. Furthermore, it synergizes WBC segmentation and classification. Our experimental results, conducted on publicly available datasets, reveal that *SANGAM* outperforms existing well-known WBC segmentation and classification systems. Additionally, it advocates for the appropriate integration of CNNs and Transformers in WBC segmentation, providing attention to relevant regions in WBC classification, and synergizing WBC classification and segmentation.

Keywords: White Blood Cell · Segmentation · Classification · Deep Learning · CNN · Transformer

1 Introduction

Human blood consists of the following blood cells: Leukocytes or white blood cells (WBCs), Erythrocytes or red blood cells (RBCs), and Thrombocytes or platelets [15]. WBCs are colorless cells with varying numbers of nuclei surrounded by a thin layer of cytoplasm [19]. They can be classified as neutrophils,

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 154–169, 2025.
https://doi.org/10.1007/978-3-031-78389-0_11

eosinophils, basophils, lymphocytes, and monocytes [14]. Each WBC class plays a distinct role in human health conditions [32]. For instance, neutrophils increase during bacterial infections [35], while eosinophils are associated with parasitic infections and allergies [2]. Similarly, lymphocytes are important for the immune response, especially during cancer treatments like radiation and chemotherapy [34]. Additionally, monocytes and basophils provide relevant information about antigens and allergic reactions [6]. In essence, accurate analysis of every WBC class is mandatory to treat, diagnose, and prognosticate a wide range of blood-related diseases. This motivated us to propose an accurate WBC segmentation and classification system in this paper.

Canonically, WBC analysis entails the manual procedure of blood slide introspection involving a medical practitioner. This procedure involving pathologists is affected by human bias [33]. Additionally, analyzing the slide manually is time-consuming [37]. To address these issues, we propose an automated WBC segmentation and classification system in this paper.

Convolution Neural Networks (CNNs) [22] are considered de facto for image segmentation tasks, and WBC segmentation is no different. For instance, variants of CNNs, such as ResNet, MobileNet [21], and DenseNet [3], are found to be useful for WBC segmentation because these variants capture short-range dependencies. However, they often miss the global context, which is crucial for accurate segmentation [1]. This paves the way for Transformers and their variants, including Vision Transformer (ViT) [24], Swin Transformer [30], and Detection Transformer (DETR) [11]. Transformers learn the global context through long-range dependencies [29], thereby surpassing CNNs. Unfortunately, Transformers offer limited performance because they neglect fine details or short-range dependencies, which are also essential for WBC segmentation besides the global context [24]. Cell structures of cytoplasm and nuclei can be better understood by analyzing local features like edges, blobs, and textures. Hence, the efficacy of WBC segmentation can be improved by carefully combining CNNs and Transformers, such that the combination allows the analysis of both local and global features, as in [5]. The efficacy can also be improved by incorporating WBC classification, an aspect yet to be explored in the literature. This leverages the intuition that basophils do not contain cytoplasm, but the WBC segmentation of basophils can contain it. If the correct classification is known, then the erroneous cytoplasm region of the basophil can be rectified.

Another important aspect of WBC analysis is WBC classification, where CNNs and Transformers have been extensively explored. Existing WBC classification systems require the entire input image, including cytoplasm, nuclei, and background, for training. However, better classification can be performed by providing more attention to relevant WBC regions, such as the cytoplasm and nuclei. This observation is supported by studies like [10], demonstrating the superior performance of CNNs with attention mechanisms focused on relevant regions. The relevant WBC regions can be provided by WBC segmentation, suggesting that WBC segmentation and classification are intertwined and can be synergized.

In this work, we introduce a novel system named ***SANGAM***, which ***synergizes*** WBC classification and segmentation. This system enhances WBC segmentation by proposing a new encoder-decoder network that integrates CNNs and Transformers to simultaneously analyze ***local and global*** features. It improves classification by focusing on the relevant regions identified through segmentation and further enhances segmentation by using classification labels. Thus, it jointly performs WBC ***classification*** and ***segmentation***. To summarize, the main contributions of this paper are:

1. To the best of our knowledge, we are the first to propose a novel WBC analysis system that synergizes WBC segmentation and classification.
2. A novel decoder architecture is proposed that combines the strengths of CNNs and Transformers, enabling both precise local and comprehensive global learning.
3. We improve the efficacy of WBC classification by providing additional information to our classifier regarding the relevant WBC regions. Our WBC segmentation identifies these relevant regions.

Our experimental results on WBC segmentation and classification using publicly available datasets reveal that our proposed system, ***SANGAM***, outperforms state-of-the-art (SOTA) systems. The remaining paper is structured as follows: Section 2 refers to the related work, while Section 3 presents our proposed system. Experimental results are provided in Section 4, followed by the conclusion in Section 5.

2 Related Work

2.1 WBC Segmentation

The WBC segmentation can be categorized into 2-class segmentation and 3-class segmentation. In 2-class segmentation, the nucleus is segmented from the input image, and the remaining image is marked as background. In contrast, 3-class segmentation refers to the segmentation of both the nucleus and cytoplasm from the input image, while marking the remaining area as background. Extensive work has been proposed in the literature for 2-class segmentation. Traditional WBC segmentation systems like [1,16,22,26] mainly rely on image processing techniques such as non-local filtering, morphological operations, gray-level thresholding, and clustering. However, these traditional image processing systems lack generalization across diverse microscopic conditions and image types [28]. This limitation paves the way for CNNs and Transformers for segmentation, as in [4,17], which employ CNN variants. Specifically, the system in [17] integrates CNN frameworks with attention mechanisms, while the system in [4] combines image processing and U-Net networks for segmentation. In contrast, work on 3-class segmentation is limited. For instance, the system in [18] performs 3-class segmentation using three different neural networks: CNN, UNet, and SegNet.

It is important to note that existing works provide limited performance because current WBC segmentation systems neglect the importance of WBC classification. Another performance-limiting factor of existing systems is their use of CNN or Transformer-based network architectures, which may neglect long-range dependencies or local/finer details [5].

2.2 WBC Classification

Traditional WBC classification systems mainly rely on image characteristics followed by machine learning, as seen in [28], which extracts shape and color features and feeds them to SVM for classification. Such traditional systems lack generalization across diverse microscopic conditions and image types. This limitation paves the way for deep learning systems, as in [3,9,10,25], where the entire image is provided to CNN variants for classification. Unfortunately, CNNs offer limited performance because they neglect the global information crucial for accurate classification. To address this, existing WBC classification systems have started utilizing Transformer variants for classification, as in [24], where the Deep Vision Transformer (ViT) is employed.

The efficacy of the aforementioned WBC classification systems can be improved by appropriately incorporating WBC segmentation information [4,22,23]. System [23] uses DeepLabv3+ for segmentation followed by AlexNet for classification. System [4] employs image processing and U-Net networks for segmentation followed by ResNet for classification, and System [22] uses thresholding with a non-local average filter for segmentation and then SqueezeNet for classification. These systems demonstrate improved efficacy by focusing on the cell rather than the background. However, these systems crop the segmented regions and then interpolate the segmented regions to a fixed size, which results in blurred boundaries and loss of low-level details such as cell size, thereby deteriorating the system's efficacy.

3 Proposed System

This section presents our proposed system, *SANGAM*. It consists of three stages. In the first stage, our system employs a novel Transformer encoder and Unified Feature Fusion (UFF) decoder architecture to perform WBC segmentation. The UFF is responsible for consolidating the CNN features in our Transformer architecture. In the next stage, the segmentation information and input image are provided to our classification network, which is the SWIN Transformer, for WBC classification. The segmentation information is needed to provide more attention to the relevant WBC regions, such as the cytoplasm and nuclei. Note that the segmentation performed in the first stage may contain errors. Therefore, the last stage performs rectification of the segmentation using the WBC classification. The flowchart of *SANGAM* is shown in Figure 1.

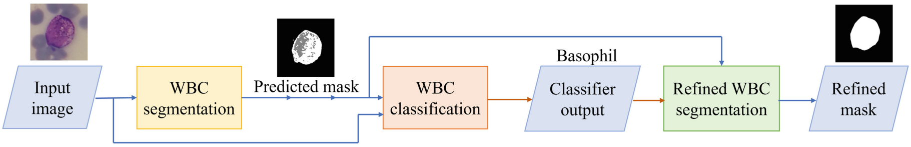


Fig. 1. Flow graph of our proposed system, *SANGAM*

3.1 WBC Segmentation

In this section, we explain our segmentation system, which merges the strengths of CNNs and Transformers to achieve accurate segmentation. It should be noted that we are performing 3-class segmentation at this stage because 2-class segmentation can be easily derived from the 3-class segmentation. The system begins with a Transformer encoder to generate features. This is followed by a Unified Feature Fusion (UFF) decoder that incorporates Spatial Detail Enhancement (SDE) and Hierarchical Feature Integration (HFI) modules. The SDE leverages CNN principles to maintain local details in the extracted features, while the HFI module integrates these features. Eventually, our system parameters are learned during training by optimizing the loss function, which is the Dice loss [27] in our case. The Dice loss assesses how well the predicted segmentation overlaps with the ground truth masks. Each component is detailed further below.

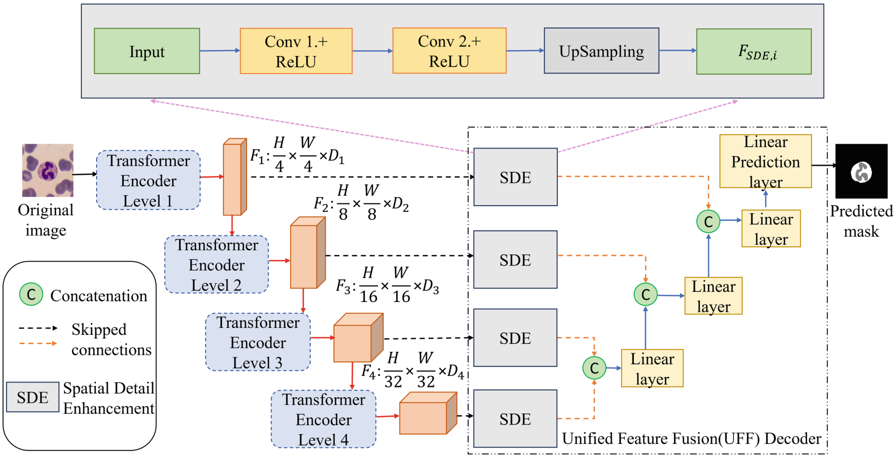

Input
Conv 1.+ ReLU
Conv 2.+ ReLU
UpSampling
$F_{SDE,i}$
Original image
Transformer Encoder Level 1
Transformer Encoder Level 2
Transformer Encoder Level 3
Transformer Encoder Level 4
$F_1: \frac{H}{4} \times \frac{W}{4} \times D_1$
$F_2: \frac{H}{8} \times \frac{W}{8} \times D_2$
$F_3: \frac{H}{16} \times \frac{W}{16} \times D_3$
$F_4: \frac{H}{32} \times \frac{W}{32} \times D_4$
SDE
SDE
SDE
SDE
C
C
C
Linear layer
Linear layer
Linear layer
Linear Prediction layer
Predicted mask
Unified Feature Fusion(UFF) Decoder
C Concatenation
Skipped connections
SDE Spatial Detail Enhancement

Fig. 2. Our WBC segmentation model depicting encoder, UFF decoder, and SDE

We employ a Transformer-based encoder with a pyramid structure, as outlined in PVTv2 [31]. Instead of traditional positional encoding, it uses convolution operations. Given an input image I with dimensions $H \times W \times 3$, the encoder generates four levels of features $\{F_i \mid i = 1, \ldots, 4\}$, each with resolutions $\left[\frac{H}{2^{k-1}}, \frac{W}{2^{k-1}}, D_i\right]$, where $k = \{3, 4, 5, 6\}$ corresponds to the respective i values. The workings of our encoder are depicted in Figure 2.

Unified Feature Fusion (UFF) decoder Experiments [20,36] have shown that the performance of segmentation can be improved by considering both local features and global context. Although Transformers excel at understanding the global context, they often miss local details such as edges, contours, and textures, which are crucial for segmentation. To address this issue, we propose the UFF decoder for feature pyramids. The UFF consists of the following:

1. **SDE:** This module uses convolution operations to focus on nearby patches and enhance local features. Unlike Transformers, which globally analyzes the relationships between all patches, SDE emphasizes features within each patch using a fixed receptive field. Integrating these features with the Transformer helps preserve local details, enhance feature extraction, and effectively synergize the strengths of CNNs and Transformers. Note that we refrain from sharing convolution weights across different levels of our feature pyramid to better adapt to varying feature depths.
   For better understanding, assume that the input feature at level i has an initial dimension D_i. It is first processed by a convolution layer, which transforms its dimension to D without altering the spatial resolution. This intermediate feature is then activated by a Rectified Linear Unit (ReLU) to introduce nonlinearity. The resulting output undergoes another convolution layer, retaining the dimension D. Finally, another ReLU activation is applied, producing the enhanced feature output. The working of SDE can be visualized in Figure 2.
2. **HFI:** Transformers and CNNs differ in their approaches to information exchange within layers. Transformers rely more on residual connections, while CNNs typically use traditional convolution connections [7,20]. Integrating local features from CNNs with global features from Transformers is therefore non-trivial. Our HFI block addresses this challenge by progressively integrating local and global features from various depths. The block achieves this through concatenation, a linear layer, and a prediction layer. It is important to note that our concatenation is effective because the shape of the feature at a depth remains unchanged after passing through the SDE block. Specifically, the concatenation operation merges refined features obtained from the SDE block at levels $i-1$ and i, resulting in a feature vector with a channel size of $2D$. This concatenated feature vector then undergoes a linear (convolution) layer to reshape its dimension to D, as depicted in Figure 2.

3.2 WBC Classification

It is shown in the literature that better classification can be achieved by providing more attention to relevant WBC regions, such as the cytoplasm and nuclei [22]. Therefore, we utilize the results of WBC segmentation from the previous stage to extract the relevant WBC regions. Specifically, we first modify the image to emphasize the relevant WBC regions. This process is referred to as region of interest (ROI) extraction. Subsequently, the modified image is provided to our WBC classification network, the SWIN Transformer [12]. The reason for employing the SWIN Transformer is its ability to effectively capture hierarchical features while maintaining global context. The flowgraph of our proposed WBC classification is shown in Figure 3. The details of these steps are provided below:

ROI extraction : In our case, the relevant WBC regions are the cell, which comprises the nucleus and cytoplasm. Our ROI extraction aims to modify the input image so that the relevant regions remain intact while the remaining regions are marked in black. This modification helps our classification network (mentioned in the subsequent step) focus on the relevant regions and ignore the background for better classification. Therefore, we create a binary image that assigns 1 to the pixels belonging to the cell region (i.e., the nucleus and cytoplasm) and 0 otherwise. Please note that our WBC segmentation (proposed in Section 3.1) provides the segmentation outputs in white, gray, and black regions corresponding to the cytoplasm, nucleus, and background, respectively. Hence, we create our binary mask, B, using:

$$B(x, y) = \begin{cases} 1 & \text{if } M(x, y) = 128 \text{ or } M(x, y) = 255 \\ 0 & \text{if } M(x, y) = 0 \end{cases} \tag{1}$$

where M denotes the segmentation mask; (x, y) represents the pixel location; and $M(x, y) = 128$ or $M(x, y) = 255$ denote the gray or white regions, respectively. Subsequently, we modify the input image by removing the background pixels. To this end, we perform the following mathematical operation to obtain the modified image, R:

$$R = I \cdot B \tag{2}$$

where $\cdot$ denotes the pixel-wise multiplication operation and I is the input image.

Swin Transformer-based classification: The ROI image generated is then fed into the Swin Transformer to classify the image into one of the five WBC classes. The Swin Transformer [12] is notable for its hierarchical feature representation and shifted window approach, which enables it to effectively analyze intricate and comprehensive features of the image. By utilizing the Swin Transformer, we leverage its capability to analyze features more effectively, resulting in accurate WBC classification. The loss function for training our Swin Transformer is cross-entropy [27].

3.3 Refined WBC Segmentation

The segmentation performed in the first stage can be erroneous; hence, this subsection rectifies the WBC segmentation using the WBC classification. It leverages the intuition that a basophil does not contain cytoplasm, but our WBC

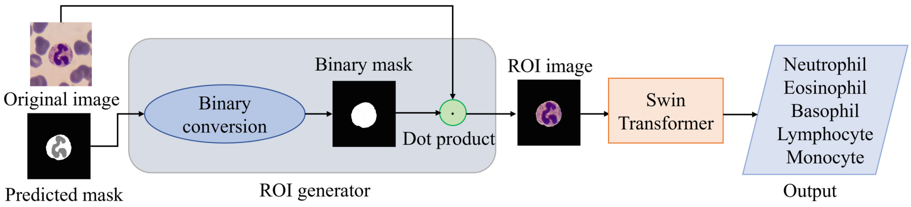


Fig. 3. Flowgraph of our proposed WBC classification

segmentation (mentioned in Section 3.1) for basophils can include it. Fortunately, our WBC classifier provides correct basophil classification in most cases, as evidenced by Table 2. Since the correct classification is known, we rectify the erroneous cytoplasm region of the basophil by marking it as a cell region. One such example is shown in Figure 4, where the erroneous cytoplasm region has been properly rectified.

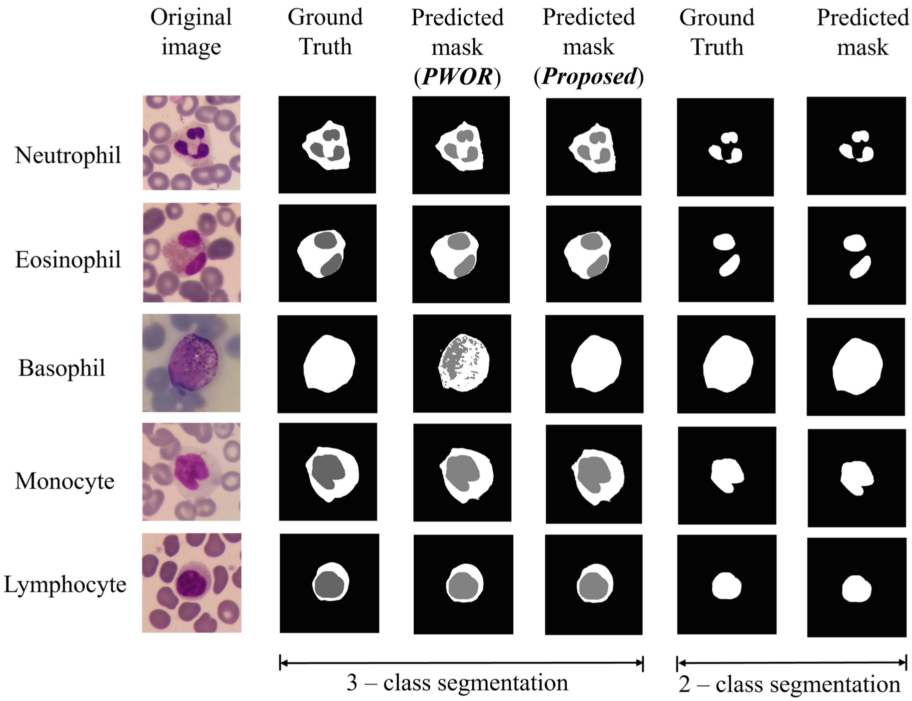


Fig. 4. Qualitative results of our system on Raabin dataset for WBC segmentation

4 Experimental Results

4.1 Experimental settings

Our proposed system, *SANGAM*, performs both WBC classification and segmentation; thus, we conduct our experiments using datasets containing both ground-truth labels. We evaluate our system using two publicly available WBC benchmark datasets: Raabin and LISC. The Raabin dataset includes 1,145 images (each of 575 × 575 pixels) for segmentation, featuring 242 neutrophils, 201 eosinophils, 218 basophils, 242 lymphocytes, and 242 monocytes. The corresponding segmentation masks consist of three areas: nucleus (grey), cytoplasm (white), and background (black). The Raabin dataset is split into Train and Test-A sets for classification. The Train and Test-A sets include 14,514 images, with counts for each cell type: Basophils (212, 89), Eosinophils (744, 322), Lymphocytes (2427, 1034), Monocytes (561, 234), and Neutrophils (6231, 2660). The LISC dataset has 242 images (720 × 576 resolution) with 53 basophils, 39 eosinophils, 52 lymphocytes, 48 monocytes, and 50 neutrophils. The corresponding ground truth masks show the nucleus in light grey, the cytoplasm in grey, and the background in black. We performed two types of segmentation tasks. For 3-class segmentation, we adjusted the ground truth to distinguish background, cytoplasm, and nuclei using black, white, and grey regions, respectively. For 2-class segmentation, we modified the ground truth by labeling cytoplasm as black and nuclei as white.

We employ the following metrics for the quantitative evaluation of WBC segmentation: mean Dice, mean IoU, and accuracy. The evaluation of WBC classification is performed using accuracy, precision, recall, and specificity. The following acronyms are used consistently in the subsections of 4.4 and 4.5: 'DSC: Dice Similarity Coefficient', 'IOU: Intersection Over Union', 'Acc: Accuracy', 'Pre: Precision', 'Rec: Recall', 'F1S: F1 Score', 'Spe: Specificity'.

Please be aware that we evaluate our system against SOTA systems under identical training and testing conditions.

4.2 Implementation details

We conducted our experiments using PyTorch on a PC equipped with an NVIDIA GeForce RTX 3050 GPU, an AMD Ryzen 5 5600X Six-Core Processor, and 16 GB of RAM. We used the AdamW optimizer [13] with an initial learning rate of 0.0001. The batch size was set to 4, and we trained for 200 epochs for both types of segmentation and for classification.

4.3 Training and Testing settings

This subsection presents our training and testing strategy used for the proposed WBC classification and segmentation. Initially, all images were resized to 572 × 572 pixels. The Raabin dataset is split into 912 images for training and 233 images for testing the segmentation network. We first train our segmentation network using this training set. Similarly, the dataset contains 10,175 images for training and 4,339 images for testing the WBC classification network. Therefore, we subsequently train our classification network on the 10,175 images. Since our classification network requires the output of the segmentation network, we freeze the segmentation network and update only the classification network parameters. After training, the efficacy of segmentation and classification is assessed on the respective test datasets. We adopted the same strategy for the LISC dataset, with the difference that we randomly split the dataset into 80% for training and 20% for testing.

4.4 Comparative WBC Segmentation Performance

Table 1. Comparative WBC segmentation of our system with SOTA systems

2-class segmentation							
System	**Raabin Dataset**			**System**	**LISC Dataset**		
	DSC	**IOU**	**Acc**		**DSC**	**IOU**	**Acc**
[38]	0.9719	0.9450	-	-	-	-	-
[17]	0.9633	0.9290	-	-	-	-	-
[8]	0.9198	0.8520	-		-	-	-
[16]	0.9542	0.9120	-	[1]	0.8991	0.8768	0.9598
[28]	0.9675	0.9360	-	[26]	0.9020	0.8997	0.9789
[4]	0.9483	0.9230	0.9899	[22]	0.8910	0.8960	0.9677
SANGAM	**0.9826**	**0.9663**	**0.9967**	***SANGAM***	**0.9334**	**0.9078**	**0.9892**
3-class segmentation							
System	**Raabin Dataset**			**System**	**LISC Dataset**		
	DSC	**IOU**	**Acc**		**DSC**	**IOU**	**Acc**
UNet [18]	0.8430	0.8391	0.9671	UNet [18]	0.7865	0.7679	0.9582
SNet [18]	0.8281	0.8263	0.9420	SNet [18]	0.7677	0.7498	0.9400
CNN [18]	0.7821	0.7739	0.9084	CNN [18]	0.7176	0.7099	0.9184
PWOR	**0.8920**	**0.8552**	**0.9819**	*PWOR*	**0.8115**	**0.7465**	**0.9976**
SANGAM	**0.9371**	**0.9204**	**0.9919**	***SANGAM***	**0.8579**	**0.8011**	**0.9983**

'SNet: SegNet', 'PWOR: Proposed without Refined Segmentation'

In this subsection, we compare the efficacy of our proposed WBC segmentation system with existing SOTA systems, and the results are shown in Table 1. Additionally, our qualitative results are presented in Figure 4 for visualization. The table demonstrates that our system outperforms other SOTA systems for

2-class segmentation. The primary reason for this is the integration of CNNs, which excel at capturing local details, and Transformers, which effectively analyze global patterns. In contrast, the SOTA systems [1,16,26], which rely on traditional image processing techniques, offer limited performance because they struggle with generalizing across diverse microscopic conditions and image types [28]. Similarly, systems based on CNN variants like SqueezeNet, U-Net++, and Mask R-CNN [4,8,17,22,38] fail to capture long-range dependencies, resulting in lower performance compared to *SANGAM*. Furthermore, Table 1 indicates that our system also outperforms other SOTA systems for 3-class segmentation. Compared to 2-class segmentation, existing work on 3-class segmentation is limited. The system [18] utilizes U-Net, SegNet, and CNN for 3-class segmentation but neglects long-range dependencies, making *SANGAM* more effective than [18].

Another important aspect that enhances the efficacy of our proposed system, *SANGAM*, is the incorporation of WBC classification. For a more thorough analysis, we introduce another system named *PWOR*, which performs WBC segmentation without rectification. The corresponding results are also shown in the table, indicating that *SANGAM* outperforms *PWOR*. This demonstrates that better performance is achieved by incorporating rectification through WBC classification. This effect can also be visualized in Figure 1, where the erroneous segmentation of the basophil class has been corrected for the 3-class segmentation task.

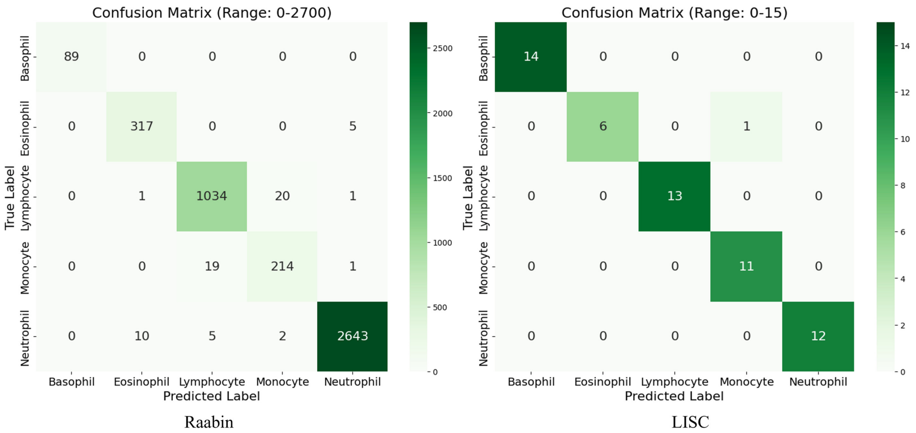
4.5 Comparative WBC Classification Performance

Table 2. Experimental results of individual classes on classification using ROI obtained after 3-class segmentation

Class	**Raabin**					**LISC**				
	Acc	**Pre**	**Rec**	**F1S**	**Spe**	**Acc**	**Pre**	**Rec**	**F1S**	**Spe**
N	0.994	0.997	0.993	0.995	0.995	1.000	1.000	1.000	1.000	1.000
E	0.996	0.966	0.984	0.975	0.997	0.983	1.000	0.857	0.923	1.000
B	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
L	0.989	0.977	0.979	0.978	0.992	1.000	1.000	1.000	1.000	1.000
M	0.990	0.906	0.914	0.910	0.994	0.983	0.917	1.000	0.957	1.000
Overall	0.985	0.970	0.983	0.972	0.996	0.983	0.983	0.980	0.979	0.996

N: Neutrophil', E: Eosinophil', B: Basophil', L: Lymphocyte', M: Monocyte'

In this subsection, we compare the efficacy of our proposed WBC classification system with existing SOTA systems. The results, shown in Table 3, indicate that our system significantly outperforms SOTA classification systems. This is because SOTA systems like [28], which rely on image characteristics and Support



Raabin

LISC

Fig. 5. Confusion matrices of our WBC classification on Raabin and LISC datasets

Table 3. Comparative WBC classification of our system with SOTA systems

System	Raabin Dataset				System	LISC Dataset			
	Acc	**Pre**	**Rec**	**F1S**		**Acc**	**Pre**	**Rec**	**F1S**
[28]	0.947	-	-	-	[28]	0.922	-	-	-
[9]	0.952	0.904	0.934	0.919	[9]	0.932	0.929	0.927	0.928
[24]	0.970	0.970	0.970	0.970	[3]	0.974	0.971	0.964	0.971
[10]	0.928	-	-	-	[10]	0.875	-	-	-
[10]	0.950	-	-	-	[10]	0.906	-	-	-
[22]	0.966	0.960	0.955	0.958	[22]	0.968	0.942	0.949	0.941
Swin-T	0.975	0.970	0.973	0.971	Swin-T	0.970	0.968	0.970	0.964
SWI	**0.979**	**0.970**	**0.976**	**0.972**	*SWI*	**0.980**	**0.983**	**0.975**	**0.978**
Prop	**0.985**	**0.970**	**0.983**	**0.972**	***Prop***	**0.983**	**0.983**	**0.980**	**0.979**

‘Swin-T: Swin-Transformer’, ‘SWI: Swin with Interpolation’, ‘Prop: Proposed system, *SANGAM*’

Vector Machines (SVM), struggle to generalize across diverse conditions. Similarly, other systems [3,9,10] use CNN variants on entire images, thereby missing crucial global information. Nevertheless, Transformer variants like ViT [24] and Swin Transformer (*Swin-T*) have captured the global context, resulting in better performance compared to CNN-based SOTA systems. Note that *Swin-T* is our proposed WBC classification method, with the difference that the entire image is fed into this system. It can also be observed that *Swin-T* outperforms ViT [24], a result consistent with the literature [12], as *Swin-T* analyzes hierarchical features.

For a more comprehensive analysis, we developed a Swin-based classification system named *SWI*. Its input image is generated using the methodology presented in [22]. Specifically, we crop the ROI from our WBC segmentation and then interpolate the extracted region. Table 3 indicates that *SWI* and our proposed system, *SANGAM*, outperform the remaining systems. Both *SWI* and our system rely on incorporating segmentation for classification. Thus, it can be inferred that the efficacy of WBC classification systems can be further improved by considering segmentation, as demonstrated by [22]. Furthermore, our system outperforms *SWI* in the table because the interpolation and cropping of segmented regions (cells) before classification led to blurred boundaries and a loss of low-level details, thereby deteriorating performance.

4.6 Ablation study

The first stage of our proposed system, *SANGAM*, performs 3-class segmentation, followed by WBC classification. To investigate the importance of 3-class segmentation, we created another system, *S2R*, using *SANGAM*. The key difference between *S2R* and our system is that *S2R* performs 2-class segmentation in the first stage rather than 3-class segmentation. The WBC classification performances of *S2R* and our system are shown in Table 4. The results demonstrate that *SANGAM* outperforms *S2R* because WBC classification in *S2R* relies only on nuclei. This reliance leads to increased misclassification, as there can be similarities in nucleus shape between certain pairs, such as eosinophils with neutrophils and monocytes with lymphocytes, as highlighted in the confusion matrix (Figure 5). In contrast, the 3-class segmentation used by *SANGAM* includes additional cytoplasmic information, leading to better classification.

Table 4. Comparitive results of *SANGAM* with *S2R* on Raabin dataset

Class	***S2R***					***SANGAM***				
	Acc	**Pre**	**Rec**	**F1S**	**Spe**	**Acc**	**Pre**	**Rec**	**F1S**	**Spe**
N	0.968	0.984	0.964	0.974	0.974	0.994	0.997	0.993	0.995	0.995
E	0.973	0.784	0.879	0.829	0.981	0.996	0.966	0.984	0.975	0.997
B	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
L	0.985	0.965	0.972	0.968	0.989	0.989	0.977	0.979	0.978	0.992
M	0.987	0.894	0.893	0.888	0.994	0.990	0.906	0.914	0.910	0.994
Overall	0.957	0.924	0.937	0.930	0.987	0.985	0.970	0.983	0.972	0.996

N: Neutrophil, E: Eosinophil, B: Basophil, L: Lymphocyte, M: Monocyte

5 Conclusion

This paper proposes a novel system, *SANGAM*, for WBC segmentation and classification. It leverages the insight that the efficacy of WBC classification and segmentation are intertwined and can therefore be synergized. Unlike the system described in [22], where segmentation guides classification, our *SANGAM* system also allows classification to enhance segmentation. Moreover, our WBC segmentation network effectively combines CNNs and Transformers to learn both local and global features, resulting in performance improvements. Our WBC classification benefits from being guided by WBC segmentation, leading to enhanced performance. To this end, our proposed ROI extraction appropriately directs the attention of our WBC classification network to the relevant image regions. Additionally, when errors occur in WBC segmentation, our refined segmentation module mitigates these issues. Experimental results conducted on publicly available datasets reveal that *SANGAM* outperforms existing well-known WBC segmentation and classification systems. Furthermore, our results demonstrate that efficacy can be improved by appropriately integrating CNNs and Transformers in WBC segmentation, providing attention to relevant regions in WBC classification, and synergizing WBC classification and segmentation.

In the future, we aim to explore this system in an unsupervised setting, primarily due to the limited availability of ground truth data in supervised settings. For this purpose, we plan to employ contrastive learning [5,27].

References

1. Jamal Ferdosi Bilkis. “Unified Approach for White Blood Cell Segmentation, Feature Extraction, and Counting using Max-Tree Data Structure”. In: International Journal of Advanced Computer Science and Applications 11.9 (2020)
2. Emine Cengil, Ahmet Çınar, and Muhammed Yıldırım. “A hybrid approach for efficient multi-classification of white blood cells based on transfer learning techniques and traditional machine learning methods”. In: Concurrency and Computation: Practice and Experience 34.6 (2022), e6756
3. Hua Chen et al. “Accurate classification of white blood cells by coupling pretrained ResNet and DenseNet with SCAM mechanism”. In: BMC bioinformatics 23.1 (2022), p. 282
4. Jose Luis Diaz Resendiz et al. “Explainable CAD System for Classification of Acute Lymphoblastic Leukemia Based on a Robust White Blood Cell Segmentation”. In: Cancers 15.13 (2023), p. 3376
5. Dixit, A., et al.: UNFOLD: 3D U-Net, 3D CNN and 3D Transformer based Hyperspectral Image Denoising. IEEE Trans. Geosci. Remote Sens. **61**, 1–10 (2023)
6. Adnan Haider et al. “Deep features aggregation-based joint segmentation of cytoplasm and nuclei in white blood cells”. In: IEEE Journal of Biomedical and Health Informatics 26.8 (2022), pp. 3685–3696
7. Kaiming He et al. “Deep residual learning for image recognition”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 770–778
8. Kaiming He et al. “Mask r-cnn”. In: Proceedings of the IEEE international conference on computer vision. 2017, pp. 2961–2969
9. Lei Jiang, Chang Tang, and Hua Zhou. “White blood cell classification via a discriminative region detection assisted feature aggregation network”. In: Biomedical Optics Express 13.10 (2022), pp. 5246–5260

10. Siraj Khan et al. “Efficient leukocytes detection and classification in microscopic blood images using convolutional neural network coupled with a dual attention network”. In: Computers in Biology and Medicine (2024), p. 108146
11. Bing Leng et al. “Deep learning detection network for peripheral blood leukocytes based on improved detection transformer”. In: Biomedical Signal Processing and Control 82 (2023), p. 104518
12. Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: Proceedings of the IEEE/CVF international conference on computer vision. 2021, pp. 10012–10022
13. Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: arXiv preprint arXiv:1711.05101 (2017)
14. Mimosette Makem et al. “A robust algorithm for white blood cell nuclei segmentation”. In: Multimedia Tools and Applications 81.13 (2022), pp. 17849–17874
15. A Meenakshi et al. “Automatic classification of white blood cells using deep features based convolutional neural network”. In: Multimedia Tools and Applications 81.21 (2022), pp. 30121–30142
16. Zahra Mousavi Kouzehkanan, Sajad Tavakoli, and Arezoo Alipanah. “Easy-GT: Open-Source Software to Facilitate Making the Ground Truth for White Blood Cells Nucleus”. In: arXiv e-prints (2021), arXiv-2101
17. Ozan Oktay et al. “Attention u-net: Learning where to look for the pancreas”. In: arXiv preprint arXiv:1804.03999 (2018)
18. Şeyma Nur Özcan, Tansel Uyar, and Gökay Karayeğen. “Comprehensive data analysis of white blood cells with classification and segmentation by using deep learning approaches”. In: Cytometry Part A (2024)
19. Jimut Bahan Pal et al. “Advancing instance segmentation and WBC classification in peripheral blood smear through domain adaptation: A study on PBC and the novel RV-PBS datasets”. In: Expert Systems with Applications 249 (2024), p. 123660
20. Raghu, M., et al.: Do vision transformers see like convolutional neural networks? Adv. Neural. Inf. Process. Syst. **34**, 12116–12128 (2021)
21. Bairaboina Sai Sambasiva Rao and Battula Srinivasa Rao: An effective WBC segmentation and classification using MobilenetV3–ShufflenetV2 based deep learning framework. IEEE Access **11**, 27739–27748 (2023)
22. S Ratheesh and A Ajisha Breethi. “Deep learning based Non-Local k-best renyi entropy for classification of white blood cell subtypes”. In: Biomedical Signal Processing and Control 90 (2024), p. 105812
23. M Roy Reena and PM Ameer. “Localization and recognition of leukocytes in peripheral blood: A deep learning approach”. In: Computers in Biology and Medicine 126 (2020), p. 104034
24. Rufus Rubin et al. “Transforming Healthcare: Raabin White Blood Cell Classification with Deep Vision Transformer”. In: 2023 6th International Conference on Signal Processing and Information Security (ICSPIS). IEEE. 2023, pp. 212–217
25. Saba Saleem et al. “A deep network designed for segmentation and classification of leukemia using fusion of the transfer learning models”. In: Complex & Intelligent Systems (2021), pp. 1–16
26. S Sapna and A Renuka. “Computer-aided system for Leukocyte nucleus segmentation and Leukocyte classification based on nucleus characteristics”. In: International Journal of Computers and Applications 42.6 (2020), pp. 622–633
27. Sneha Shukla, Anup Kumar Gupta, and Puneet Gupta. “Exploring the feasibility of adversarial attacks on medical image segmentation”. In: Multimedia Tools and Applications 83.4 (2024), pp. 11745–11768

28. Sajad Tavakoli et al. “New segmentation and feature extraction algorithm for classification of white blood cells in peripheral smear images”. In: Scientific Reports 11.1 (2021), p. 19428
29. Yi Tay et al. “Long range arena: A benchmark for efficient transformers”. In: arXiv preprint arXiv:2011.04006 (2020)
30. Hüseyin Üzen and Hüseyin Firat. “A hybrid approach based on multipath Swin transformer and ConvMixer for white blood cells classification”. In: Health Information Science and Systems 12.1 (2024), p. 33
31. Wenhai Wang et al. “Pvt v2: Improved baselines with pyramid vision transformer”. In: Computational Visual Media 8.3 (2022), pp. 415–424
32. Jiangping Wu et al. “WBC image segmentation based on residual networks and attentional mechanisms”. In: Computational Intelligence and Neuroscience 2022 (2022)
33. Dongxu Yang et al. “Leukocyte subtypes identification using bilinear self-attention convolutional neural network”. In: Measurement 173 (2021), p. 108643
34. Qiang Zhai et al. “Automatic white blood cell classification based on whole-slide images with a deeply aggregated neural network”. In: Journal of Medical and Biological Engineering 42.1 (2022), pp. 126–137
35. Zhao, M., et al.: MSS-WISN: Multiscale multistaining WBCs instance segmentation network. IEEE Access **10**, 65598–65610 (2022)
36. Sixiao Zheng et al. “Rethinking semantic segmentation from a sequence-tosequence perspective with transformers”. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, pp. 6881–6890
37. Xin Zheng et al. “White blood cell detection using saliency detection and CenterNet: A two-stage approach”. In: Journal of Biophotonics 16.3 (2023), e202200174
38. Zongwei Zhou et al. “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, Spain, September 20, 2018, Proceedings 4. Springer. 2018, pp. 3–11

MeDiANet: A Lightweight Network for Large-scale Multi-disease Classification of Multi-modal Medical Images Using Dilated Convolution and Attention Network

Dipayan Dewan[1](✉), Asim Manna[1], Apoorva Srivastava[1,2], Anupam Borthakur[1], and Debdoot Sheet[1]

[1] Indian Institute of Technology Kharagpur, Kharagpur 721302, India
{diiipayan93,asimmanna17,anupamborthakur}@kgpian.iitkgp.ac.in, debdoot@ee.iitkgp.ac.in
[2] University of Warwick, Coventry CV47AL, UK

Abstract. Medical image classification is a critical component of modern healthcare, providing numerous advantages, including improved diagnostic accuracy and treatment planning. Integrating deep learning for medical image classification gives the ability to provide high accuracy and extract automated features. The residual connections, dilated convolutions, and attention mechanisms were introduced to enhance the performance of a very deep neural network. However, it remains challenging to achieve high classification accuracy in medical image classification tasks using neural network models with fewer trainable parameters and floating point operations per second (FLOPs). In this paper, we propose a lightweight neural network model for medical image classification, named as **Me**dical **Di**lated Convolution and **A**ttention **Net**work (MeDiANet), which achieves better accuracy even with the fewer parameters and FLOPs as compared to the state-of-the-art (SOTA) models. The Dilated Residual Attention (DiET) is introduced in `MeDiANet` which provides the access to usage of different dilation rate based on the depth of the network. Also, by inflating the kernel size, more features can be extracted while keeping the parameter count low. The performance of MeDiANet has been evaluated on a large-scale multi-modal dataset. It achieves an accuracy of 94.18% with 0.38M trainable parameters, 0.08 FLOPSs, and lower inference time of 4.2 seconds which is on average, 1.66% higher in accuracy with 12.50× and 2.55% lower trainable parameter and FLOPs respectively, with 7.44% higher inference time compared to prior art.

Keywords: Deep learning · Dilated convolution · Lightweight deep neural network · Medical image classification

1 Introduction

Deep learning has been successfully implemented in various application-specific studies, including medical image analysis [26], satellite image recognition [19],

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 170–184, 2025.
https://doi.org/10.1007/978-3-031-78389-0_12

video processing [21], and more [5,6,24,25]. In the field of medical image recognition, the implementation of deep neural networks (DNNs) has shown a significant development over the past few years [26]. Medical image classification is a crucial task in the field of image recognition, that aims to provide valuable assistance to researchers and doctors in disease diagnosis and research endeavors. Although researchers have made substantial contributions to medical image classification, still the the inherent diversity of medical images obtained from various sources pose challenges to medical image classification. The challenges include variations in image contrast and focusing region, coupled with the presence of inner structures with variable pixel densities and textures, all of which contribute to the complexity of medical image classification. Additionally, medical images often contain inner structures with varying pixel densities and textures. Consequently, relying on classical features for medical image classification can increase the risk of mis-classification [2,33] (Fig. 1).

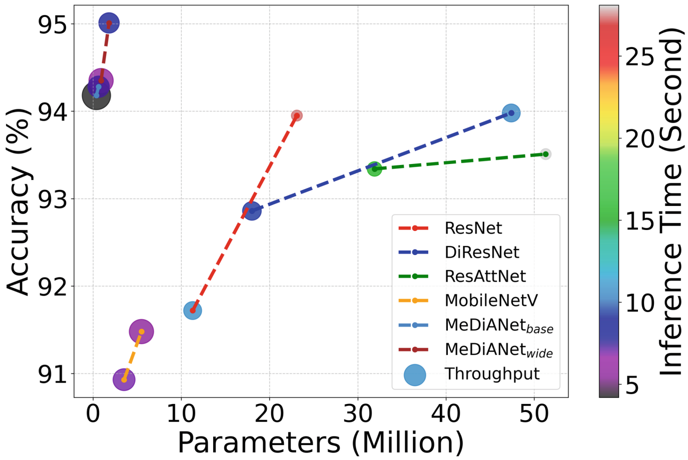


Fig. 1. An overview of proposed MeDiANet performance with state-of-the-art models. Accuracy, Parameter, Inference time, and Throughput of all models have been shown here. Bubble radius denotes the throughput where bigger bubbles are better. Every bubble has a different color corresponding to its inference time.

Convolutional Neural Network (CNN) architectures [10,12,22] have shown promising results not only in the field of medical imaging but also in other domains. As a powerful branch of machine learning, CNN models can effectively compute final class labels when raw pixels of biomedical images are provided as input to the model. However, certain variations in biomedical images, such as irregularities and scale differences in Region of Interests (ROIs), can pose difficulties for CNNs. These challenges require more specialized architectures and techniques to robustly analyze and classify the images. The proposed Medical Dilated Convolution and Attention Network (MeDiANet) architecture builds upon the strengths of CNNs and incorporates additional components, such as the Residual Attention Network (ResAttNet) [3], to specifically address the challenges present in biomedical image classification. MeDiANet aims to provide a more robust and versatile solution that can effectively handle irregularities, scale differences in the ROIs of the biomedical images while effectively capturing variations within the images, and thus improves the accuracy and performance of biomedical image classification compared to standard CNN architectures.

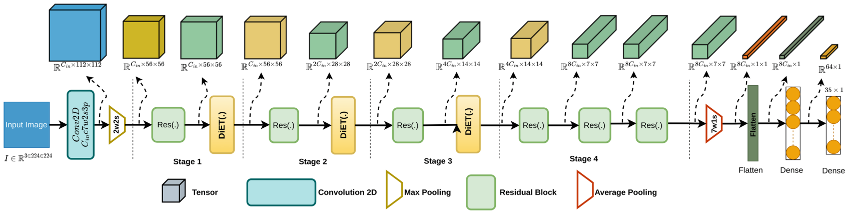

$\mathbb{R}^{C_{in}\times112\times112}$
$\mathbb{R}^{C_{in}\times56\times56}$
$\mathbb{R}^{C_{in}\times56\times56}$
$\mathbb{R}^{C_{in}\times56\times56}$
$\mathbb{R}^{2C_{in}\times28\times28}$
$\mathbb{R}^{2C_{in}\times28\times28}$
$\mathbb{R}^{4C_{in}\times14\times14}$
$\mathbb{R}^{4C_{in}\times14\times14}$
$\mathbb{R}^{8C_{in}\times7\times7}$
$\mathbb{R}^{8C_{in}\times7\times7}$
$\mathbb{R}^{8C_{in}\times7\times7}$
$\mathbb{R}^{8C_{in}\times1\times1}$
$\mathbb{R}^{8C_{in}\times1}$
$\mathbb{R}^{64\times1}$
35 × 1
Input Image
$I \in \mathbb{R}^{3\in224\in224}$
$Conv2D$
$C_{in}c7w2s3p$
2w2s
Res(.)
DiET(.)
Res(.)
DiET(.)
Res(.)
DiET(.)
Res(.)
Res(.)
Res(.)
7w1s
Flatten
Stage 1
Stage 2
Stage 3
Stage 4
Flatten
Dense
Dense
Tensor
Convolution 2D
Max Pooling
Residual Block
Average Pooling

Fig. 2. The overall architecture of MeDiANet. The input image $I \in \mathbb{R}^{3\in224\in224}$ is passes through a $Conv2D$ layer with C_{in} output channels (c) and kernel size (w) of 7 × 7 with a stride (s) of 2 and zero-padding (p) of 3, followed by a max pool layer with kernel size (w) 2, stride (s) 2. This architecture contains total of four stages. The first three stages contains one residual block ($Res(\cdot)$) and one $DiET(\cdot)$ block each. The fourth stage contains only $Res(\cdot)$. Output shape is 1 × 35.

In the present study, we incorporated two key concepts to improve the performance of the proposed MeDiANet architecture. Firstly, we utilized dilated convolution, as highlighted in the study by Yu *et al.* in [31], to address the challenge of extracting features from images that exhibit variations in scale. This is achieved by utilizing a multi-dilated residual block within the Dilated Residual Attention (DiET) block, that allows the network to extract broader range of features from images that carry more spatial information at different scales. Thus the model effectively captures the information across various levels of detail and improves its ability to analyze and classify images with varying scales. Secondly, we have employed the pre-activated residual block, as introduced by He *et al.* [8]. This block can not only addresse the vanishing gradient problem commonly encountered in DNNs but also enhances the overall performance of the network. The primary objective here is to strike a balance between model complexity and performance, seeking a more efficient and lightweight architecture for biomedical image classification. By reducing the number of trainable parameters, MeDiANet aims to improve computational efficiency and mitigate issues related to overfitting. Despite having fewer parameters, the model strives to maintain or even surpass the performance of existing approaches, ensuring robust and accurate classification results for biomedical images.

Our contributions are summarized as follows:

1. A lightweight and computationally efficient DNN namely, MeDiANet is proposed. It integrates several novel components pre-activated residual block,

multi-dilated residual block, and novel dilated residual attention block. These blocks are designed to reduce the overall parameter of the network and extract features in different scales while maintaining low compute complexity.
2. MeDiANet is robust enough to handle multi-resolution, multi-disease, and multi-modal medical image datasets.
3. The proposed MeDiANet's accuracy outperforms the state-of-the-art with regards to classification scores and demonstrates better convergence with $\approx$ $135\times$ lesser parameters.

The paper is structured as follows: In Section 2, we delve into the related prior work. The working principle of MeDiANet is explained in Section 3. We then proceed to Section 4, where we outline the experimental setup. The results of our experiments are presented in Section 5. Finally, we conclude our study in Section 6.

2 Related Work

CNNs [10,11] have been extensively utilized in various computer vision tasks for a decade. According to He et al. [7], incorporating residual connections in the ResNet network enhances performance without increasing the number of parameters. In addition to developing high-performing models, it is crucial that these models remain lightweight, thereby making them irreplaceable for various applications where resource constraints and real-time performance are critical. The models such as MobileNetv2 [20] and MobileNetv3 [9] use the concept of inverted residual, linear bottleneck to have better classification accuracy with fewer floating point operations per second (FLOPs) on Imagenet data for edge devices. ShuffleNet [17] further reduce the FLOPs count by utilizing channel shuffle and group convolution. The performance of the classical ResNet network is further improved in Resnext [30] by using the concept of group convolutions while keeping the same parameter as ResNet. Futher, EfficientNet [27] introduces a neural architecture search leveraging a highly effective compound scaling method to scale up the performance of the model while reducing the FLOPs of the network significantly. Furthermore models such as ConvNext [13] and ConvNextv2 [29] were introduced to further elevate the classification performance on ImageNet with fewer parameters by utilizing the ConvNext block with GeLU activation and global response normalization with masked autoencoder respectively.

In recent times, several CNN-based architectures have emerged as robust tools for medical image classification, providing accurate and efficient analysis of various imaging modalities. Several studies have explored the application of deep learning techniques in medical image analysis, paving the way for significant advancements in this field. The lightweight models such as ShuffleNet, MobileNetv2, MobileNetv3, ConvNeXt-A are also employed on medical images for the diagnosis of skin disease [23], cervical cancer[28], etc. In addition to utilizing lightweight models, researchers also employed concepts such as dilated convolutions and attention networks to enhance performance. In [35], Zhou *et al.* explored the effectiveness of integrating dilated convolutions into medical image classification networks. Their research highlights the benefits of using dilated convolutions in enlarging the receptive field without significantly increasing the number of parameters. A dual attention network is proposed in [4] for the classification of diabetic retinopathy severity. The network combines dilated convolutions with spatial attention mechanisms to capture multi-scale multi-level features to highlight salient features in retinal images.

Despite these advancements, the computational complexity of these networks has not been adequately addressed. Additionally, these proposed networks are trained and tested for a single modality of medical image data leading to different models for different modalities. In this work, we propose a network architecture that addresses and resolves this issue, aiming to develop a computationally efficient solution for medical image analysis tasks. Also, the model is trained and tested for different modalities of medical images leading to a single model for different medical image modalities.

3 Methodology

In this work, a lightweight and computationally efficient classification network architecture, named as medical dilated convolution and attention network (MeDiANet) is proposed (Figure2). The proposed network incorporates two key components: the multi-dilated residual block ($\texttt{MDiRes}(\cdot)$) and the dilated residual attention block ($\texttt{DiET}(\cdot)$).

3.1 Residual Block

The proposed network module includes a pre-activated residual blocks [8] as an essential architectural element, as illustrated in Figure 3. $\texttt{Mish}(\cdot)$ activation function which is introduced by Mishra et al., in [18] is used in the network, instead of $\texttt{ReLU}(\cdot)$. This pre-activated residual block, denoted as $\texttt{Res}(\cdot)$, replaces the standard residual module and plays a significant role in the proposed network's architecture. The modification has been done in the channel dimension to reduce the no of trainable parameters of the $\texttt{Res}(\cdot)$, which eventually results in the overall reduction of trainable parameters of the proposed MeDiANet. Let $\mathbf{X} \in \mathbb{R}^{K\times M\times N}$ and $\mathbf{R} \in \mathbb{R}^{K\times M\times N}$ respectively be the input and output features of a residual block, such that

$$\begin{aligned} \mathbf{R} &= \texttt{Res}(\mathbf{X}) \\ \texttt{Res}(\mathbf{X}) &= \texttt{F}(\mathbf{X}) + \mathbf{X} \end{aligned} \tag{1}$$

$$\begin{aligned} \texttt{F}(\cdot) \mapsto \texttt{BatchNormalization} \to \texttt{Mish} \to \texttt{Conv2D}\left(\frac{\texttt{K}}{4}\texttt{c1w1s0p}\right) \to \\ \texttt{BatchNormalization} \to \texttt{Mish} \to \texttt{Conv2D}\left(\frac{\texttt{K}}{4}\texttt{c3w1s2p}\right) \to \\ \texttt{BatchNormalization} \to \texttt{Mish} \to \texttt{Conv2D}\left(\texttt{Kc1w1s0p}\right) \end{aligned} \tag{2}$$

Here Conv2D $\left(\frac{\texttt{K}}{4}\texttt{c1w1s0p}\right)$ represents a Conv2D operation of $\frac{\texttt{K}}{4}$ kernels (c) of size (w) 1×1, with a stride (s) 1, 0 padding (p). In traditional residual block, channel expansion is done in the last convolution layer by a factor of 4 to make the number of channels 4K from an input feature with K number of channels (Figure 3). However, in our proposed MDiRes$(\cdot)$, the number of channels remains the same as the input (K). In order to maintain the expansion factor of 4 like a traditional Residual Block, the number of channels of the 1_{st} two convolution layer inside the Res$(\cdot)$ has been reduced by a factor of $\frac{1}{4}$. The above formulation allows the layers of the residual block to learn not only the entire transformation but also the modifications to the identity mapping [7], which, in turn, enhances the performance of the proposed DNN.

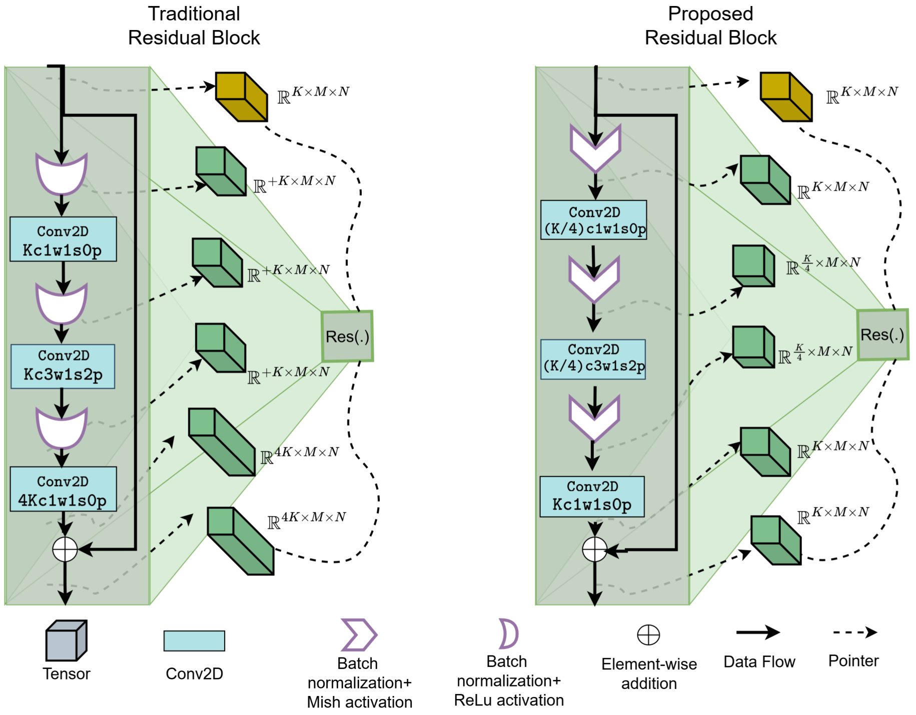

Traditional Residual Block
Proposed Residual Block
Conv2D Kc1w1s0p
Conv2D Kc3w1s2p
Conv2D 4Kc1w1s0p
Conv2D (K/4)c1w1s0p
Conv2D (K/4)c3w1s2p
Conv2D Kc1w1s0p
Res(.)
Res(.)
$\mathbb{R}^{K\times M\times N}$
$\mathbb{R}^{+K\times M\times N}$
$\mathbb{R}^{+K\times M\times N}$
$\mathbb{R}^{+K\times M\times N}$
$\mathbb{R}^{4K\times M\times N}$
$\mathbb{R}^{4K\times M\times N}$
$\mathbb{R}^{K\times M\times N}$
$\mathbb{R}^{K\times M\times N}$
$\mathbb{R}^{\frac{K}{4}\times M\times N}$
$\mathbb{R}^{\frac{K}{4}\times M\times N}$
$\mathbb{R}^{K\times M\times N}$
$\mathbb{R}^{K\times M\times N}$
Tensor
Conv2D
Batch normalization+ Mish activation
Batch normalization+ ReLu activation
Element-wise addition
Data Flow
Pointer

Fig. 3. Architecture of proposed residual block (Res$(\cdot)$), where the number of kernels for the input feature maps are same which is denoted by K, unlike traditional residual block

3.2 Multi Dilated residual block

Each multi dilated residual block defined as MDiRes$(\cdot)$, includes three parallel Conv2D of different dilation rates with the residual connection (shown in Figure

4a), which helps extracting relevant features of the input images even if they differ in sizes and scales. Similar to the $\texttt{Res}(\cdot)$, $\texttt{MDiRes}(\cdot)$ can be represented as following:

$$
\begin{aligned}
\mathtt{F_d}(\cdot) \mapsto \mathtt{BatchNormalization} &\rightarrow \mathtt{Mish} \rightarrow \\
&\mathtt{Conv2D}\left(\frac{\mathtt{K}}{4}\mathtt{c1w1s0p}\right) \rightarrow \\
&\mathtt{BatchNormalization} \rightarrow \mathtt{Mish} \rightarrow \\
&\left\{\begin{array}{l}\mathtt{Conv2D}\left(\frac{\mathtt{K}}{4}\mathtt{c3w1s2pR_1d}\right) \\ \| \ \mathtt{Conv2D}\left(\frac{\mathtt{K}}{4}\mathtt{c3w1s2pR_2d}\right) \\ \| \ \mathtt{Conv2D}\left(\frac{\mathtt{K}}{4}\mathtt{c3w1s2pR_3d}\right)\end{array}\right\} \rightarrow \mathtt{Add}(\cdot) \rightarrow \\
&\mathtt{BatchNormalization} \rightarrow \mathtt{Mish} \rightarrow \\
&\mathtt{Conv2D(Kc1w1s0p)}
\end{aligned} \tag{3}
$$

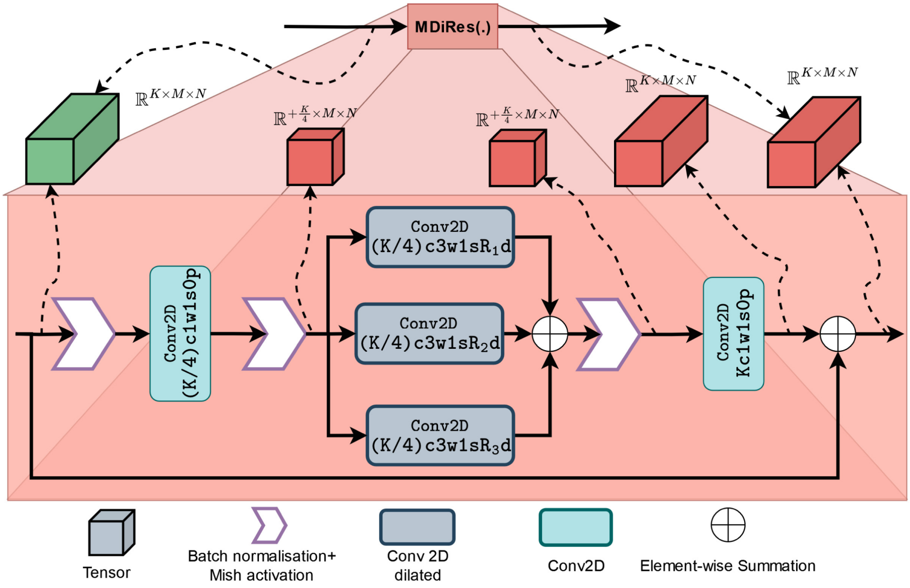

MDiRes(.)
$\mathbb{R}^{K\times M\times N}$
$\mathbb{R}^{+\frac{K}{4}\times M\times N}$
$\mathbb{R}^{+\frac{K}{4}\times M\times N}$
$\mathbb{R}^{K\times M\times N}$
$\mathbb{R}^{K\times M\times N}$
Conv2D (K/4)c1w1s0p
Conv2D (K/4)c3w1sR$_1$d
Conv2D (K/4)c3w1sR$_2$d
Conv2D (K/4)c3w1sR$_3$d
Conv2D Kc1w1s0p
Tensor
Batch normalisation+ Mish activation
Conv 2D dilated
Conv2D
Element-wise Summation

Fig. 4. Architecture of $\texttt{MDiRes}(\cdot)$. In this block, after first convolutional layer, the output passes through into three parallel convolution layer with dilation rate ($\texttt{d}$) of $\mathtt{R_1}$, $\mathtt{R_2}$, $\mathtt{R_3}$ respectively to construct a $\texttt{MDiRes}(\cdot)$

$\mathtt{R_1}$ in $\frac{\mathtt{K}}{4}\mathtt{c3w1s2pR_1d}$ represent the dilation rate ($\texttt{d}$) of the convolution operation. The exact values of $\mathtt{R_1}, \mathtt{R_2}, \mathtt{R_3}$ used in (3) network is mentioned in Table 1.

3.3 Dilated Residual Attention Block

Previously attention mechanism has been deployed in various computer vision tasks including medical image classification, which enable it to enhance the performance of the model. It has been seen that inclusion of residual attention block provides better result than channel attention or spatial attention [3]. Thus in this proposed network, $\mathtt{DiET}(\cdot)$ block is introduced where the residual attention mechanism has been deployed using $\mathtt{MDiRes}(\cdot)$. The overall block diagram of $\mathtt{DiET}(\cdot)$ has been given in Figure 5.

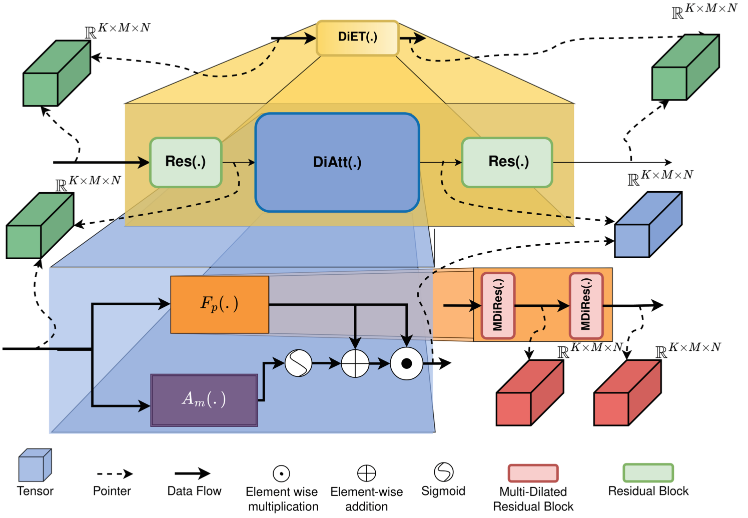

$\mathbb{R}^{K\times M\times N}$
DiET(.)
$\mathbb{R}^{K\times M\times N}$
Res(.)
DiAtt(.)
Res(.)
$\mathbb{R}^{K\times M\times N}$
$\mathbb{R}^{K\times M\times N}$
$F_p(.)$
$A_m(.)$
MDiRes(.)
MDiRes(.)
$\mathbb{R}^{K\times M\times N}$
$\mathbb{R}^{K\times M\times N}$
Tensor
Pointer
Data Flow
Element wise multiplication
Element-wise addition
Sigmoid
Multi-Dilated Residual Block
Residual Block

Fig. 5. The overview of $\mathtt{DiET}(\cdot)$ block which consists of $\mathtt{Res}(\cdot)$ block and $\mathtt{DiAtt}(\cdot)$ block. Here input feature passes through a $\mathtt{Res}(\cdot)$ before going into the $\mathtt{DiAtt}(\cdot)$. The $\mathtt{DiAtt}(\cdot)$ output features then again will go through a $\mathtt{Res}(\cdot)$ and finally produce the output feature map of $\mathtt{DiET}(\cdot)$.

The output of the $\mathtt{DiET}(\cdot)$ block for a given input feature $\mathbf{X} \in \mathbb{R}^{K\times M\times N}$ is $\mathbf{D} \in \mathbb{R}^{K\times M\times N}$. In each $\mathtt{DiET}(\cdot)$ block the output feature of previous block is passed through a residual block. Then the output of this residual block goes into the dilated attention block denoted as $\mathtt{DiAtt}(\cdot)$, followed by a residual block to produce the output of the $\mathtt{DiET}(\cdot)$ block, such that

$$\mathbf{D} = \mathtt{DiET}(\mathbf{X}) = \mathtt{Res}(\mathtt{DiAtt}(\mathtt{Res}(\mathbf{X}))) \tag{4}$$

$\mathtt{DiAtt}(\cdot)$ consists of two parallel branches. The branch which performs feature processing is called as the trunk branch ($\mathtt{F_p}(\cdot)$), and the other branch acts as an attention mask ($\mathtt{A_m}(\cdot)$) that provides the feature selection mechanism during forwarding inference generation. This can be represented as,

$$\mathtt{DiAtt}(\cdot) = \mathtt{F_p}(\cdot) \oplus (\mathtt{F_p}(\cdot) \odot (\mathtt{Sigmoid}(\mathtt{A_m}(\cdot)))) \tag{5}$$

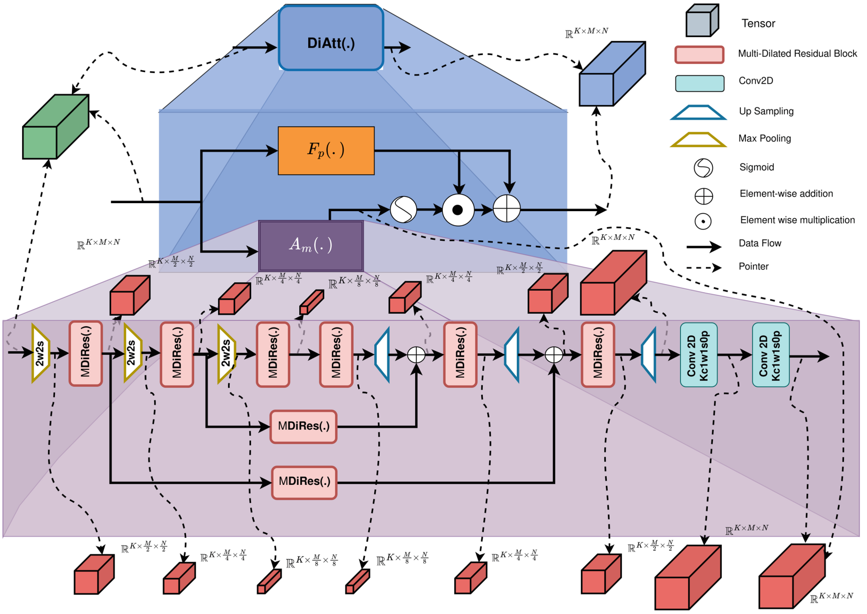

DiAtt(.)
$F_p(.)$
$A_m(.)$
Tensor
Multi-Dilated Residual Block
Conv2D
Up Sampling
Max Pooling
Sigmoid
Element-wise addition
Element wise multiplication
Data Flow
Pointer
$\mathbb{R}^{K\times M\times N}$
$\mathbb{R}^{K\times \frac{M}{2}\times \frac{N}{2}}$
$\mathbb{R}^{K\times \frac{M}{4}\times \frac{N}{4}}$
$\mathbb{R}^{K\times \frac{M}{8}\times \frac{N}{8}}$
2w2s
MDiRes(.)
Conv 2D Kc1w1s0p

Fig. 6. `DiAtt`(·) has been elaborated with two parallel path of trunk branch (`Fp`(·)) and attention mask (`Am`(·)) branch. `Am`(·) is illustrated further in here, which consists of multiple `MDiRes`(·), `MaxPool`, `UpSample`, `Conv2D` layer.

where, ⊕ and ⊙ denotes element-wise addition and multiplication, respectively. Here, output of `Res`(·) is fed as input to the both $\mathtt{A_m}(\cdot)$ and $\mathtt{F_p}(\cdot)$ in `DiAtt`(·) .

In this architecture, the key entity is the attention mask $\mathtt{A_m}(\cdot)$ as it fetches the good features as well as suppresses the noise from the feature processed by $\mathtt{F_p}(\cdot)$ in (5). $\mathtt{F_p}(\cdot)$ comprises of two successive `MDiRes`(·), such that

$$\mathtt{F_p}(\cdot) = \mathtt{MDiRes}(\mathtt{MDiRes}(\cdot)) \tag{6}$$

The $\mathtt{A_m}(\cdot)$ shown in figure 6, utilizes the bottom-up top-down approach [14] similar to an encoder-decoder based architecture. The $\mathtt{A_m}(\cdot)$, consists of multiple `Up`(·) and `Down`(·) blocks, which are defined as,

$$\begin{aligned}\mathtt{A_m}(\cdot) = \mathtt{Conv2D}(\mathtt{Conv2D}(\mathtt{Up_3}(\mathtt{Up_2}(\mathtt{Up_1}(\mathtt{Down_3}(\mathtt{Down_2}(\mathtt{Down_1}(\cdot))))\\ + \mathtt{MDiRes}(\mathtt{Down_2}(\mathtt{Down_1}(\cdot))))\\ + \mathtt{MDiRes}(\mathtt{Down_1}(\cdot)))))\end{aligned} \tag{7}$$

$$\mathtt{Down}_i(\cdot) = \mathtt{MDiRes}_i(\mathtt{MaxPool}_i(\cdot)) \tag{8}$$

$$\mathtt{Up}_i(\cdot) = \mathtt{Upsample}_i(\mathtt{MDiRes}_i(\cdot)) \tag{9}$$

Here in eq (7), `Conv2D`(·) has hyperparameter defined as `Conv2D(Kc1w1s1p0d)` where `K` is defined as number of kernels. Also, here $i \in \{1, 2, 3\}$ denotes the index of the corresponding block in (8) and (9). The $\mathtt{Sigmoid}(\cdot)$ activation is applied to normalize the output range of the $\mathtt{A_m}(\cdot)$ to [0, 1].

3.4 MeDiANet

The overall network architecture is illustrated in Figure 2. Here, two different versions of MeDiANet are proposed based on the number of channels used in the first convolution layer of the network, denoted as C_{in}. The proposed network named as MeDiANet$_{base}$ for $C_{in} = 16$ and MeDiANet$_{wide}$ for $C_{in} = 32$. Each of these 2 variants of the network has two different versions depending on the number of trainable layers present in the network. The total number of trainable layers in the network is denoted as B, where $\mathrm{B} = 16m + 21$, and m is the number of $\mathtt{DiET}(\cdot)$ blocks belonging to {3,6}. The details of these MeDiANet$_{base}$-B have been given in Table 1.

Table 1. Detailed architecture of proposed MeDiANet$_{base}$-B

ïż£

Output Size	**Dilation** $[\mathtt{R}_1, \mathtt{R}_2, \mathtt{R}_3]$	**Stride**	**Kernel**	**Channel out**	**B**	
					69	**117**
112 ÃČ- 112	1	2	7 ÃČ- 7	C_{in}	Conv2D	Conv2D
56 ÃČ- 56	NA	2	2 ÃČ- 2	C_{in}	MaxPool2D	MaxPool2D
56 ÃČ- 56	1	1	3 ÃČ- 3	C_{in}	$\mathtt{Res}(\cdot)$-1	$\mathtt{Res}(\cdot)$-1
56 ÃČ- 56	[4,8,12]	1	3 ÃČ- 3	C_{in}	$\mathtt{DiET}(\cdot)$-1	$\mathtt{DiET}(\cdot)$-1
28 ÃČ- 28	1	2	3 ÃČ- 3	$2C_{in}$	$\mathtt{Res}(\cdot)$-2	$\mathtt{Res}(\cdot)$-2
28 ÃČ- 28	[2,4,6]	1	3 ÃČ- 3	$2C_{in}$	$\mathtt{DiET}(\cdot)$-2	2 ÃČ- $\mathtt{DiET}(\cdot)$-2
14 ÃČ- 14	1	2	3 ÃČ- 3	$4C_{in}$	$\mathtt{Res}(\cdot)$-3	$\mathtt{Res}(\cdot)$-3
14 ÃČ- 14	[1,2,3]	1	3 ÃČ- 3	$4C_{in}$	$\mathtt{DiET}(\cdot)$-3	3 ÃČ- $\mathtt{DiET}(\cdot)$-3
7 ÃČ- 7	2	2	3 ÃČ- 3	$8C_{in}$	$\mathtt{Res}(\cdot)$-4	$\mathtt{Res}(\cdot)$-4
7 ÃČ- 7	1	1	3 ÃČ- 3	$8C_{in}$	$\mathtt{Res}(\cdot)$-5	$\mathtt{Res}(\cdot)$-5
7 ÃČ- 7	1	1	3 ÃČ- 3	$8C_{in}$	$\mathtt{Res}(\cdot)$-6	$\mathtt{Res}(\cdot)$-6
1 ÃČ- 1	NA	NA	7 ÃČ- 7	$8C_{in}$	AvgPool2D + Dropout	AvgPool2D + Dropout
1	NA	NA	NA	64	Linear	Linear
1	NA	NA	NA	#classes	Linear	Linear
Total Parameter (Million)					**0.38**	**0.63**

4 Experimental Setup

4.1 Dataset

Multiple dataset from different sources has been aggregated to create a large-scale medical image benchmark dataset in order to measure its performance. As each of the dataset's images are of different sizes, the images are resized to 3 × 224 × 224 before the training process. This dataset contains total of 35 diseases and is divided into train, validation, and test with a ratio of 7 : 1 : 2. The details of this dataset are presented in Table 2.

Table 2. An overview of the dataset used for this experiment. A total of 35 classes are taken from 4 different modalities.

Dataset	Modality	#Classes	Number of images(Train, Validation, Test)
Blood[a]	Microscope	8	(11534, 1276, 2135)
RetinalFundus [b]	Fundus	11	(14699, 1623, 2713)
Path [c]	Colon Pathology	9	(58397, 7481, 12456)
Skin [d]	Dermatoscope	7	(6761, 756, 1256)

[a] https://data.mendeley.com/datasets/snkd93bnjr/1
[b] https://www5.cs.fau.de/research/data/fundus-images/
[c] https://zenodo.org/records/1214456
[d] https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000/home/

4.2 Implementation Details

Experiments were performed on 2x Intel(R) Xeon(R) 4110CPU, 4x32 GB DDR4 ECC Regd. RAM, 2xNvidia Tesla V100 SXM2 with 16GB HBM2 & NVLink, 1x2TB HDD. The model is implemented in TensorFlow 2.15.1, using Python 3.9 version. The model has been trained with a batch size of 192 using AdamW [16] optimizer. The Cosine decay learning rate scheduler [15] is used with a warmup target of 2×10^{-4} after the first 40 epochs. The initial learning rate for warmup was found by hyperparameter tuned and set to 7×10^{-4}. The network has been trained for 400 epochs. $\texttt{Softmax}(\cdot)$ activation is used at the output layer as this is a multi-class classification task with 35 classes and employs the categorical cross-entropy loss[34] function with label smoothing of 0.1 to calculate the prediction loss.

4.3 Evaluation Metrics

Precision, recall, and F1 score are used as performance metrics for measurement of the experiment result of the proposed network. Though accuracy is widely used for the evaluation of multi-class classification networks, the above three metrics are also used for better understanding of the model performance for individual classes. Table 3 refers to the comparison of the proposed model performances.

5 Results & Discussions

A customised 2D medical images dataset (Table 2 has been used to evaluate the model performance of the proposed and state-of-the-art (SOTA) models. The classical ResNet [7] as well as a number of recently proposed extensions of the classical ResNet architecture, including the DRN [32], ResAttNet [3], ConvNext [13], and ConvNextV2 [29] has been used for comparison purposes. In order to observe the efficiency, the proposed network has been also compared with the SOTA lightweight networks such as MobileNetv2 [20], MobileNetV3 [9], MobileNetV 4[9], ShuffleNetV2 [17], and EfficientNet lite [27] etc.

Table 3. Comparison of the proposed model with the SOTA. Throughput (image/second) during inference and Inference Time (s) has been measured on a single core of the Intel(R) Xeon(R) 4110CPU with 256 batch size in float16. **Bold** values represent the best performance and underline represent the reference for comparison

Model	Accuracy (%)	Parameter (Million)	FLOPs (G)	Throughput	Inference Time (s)
ResNet-18 [7]	91.72 (-2.46)	11.29 (↑ 6.23×)	1.82(↑ 3.1×)	26(-3)	9.3 (↑ 14.8%)
ResNet-50 [7]	92.95 (-1.23)	23.10 (↑ 12.16×)	3.82(↑ 6.6×)	24(-5)	10.5 (↑ 29.6%)
DRN-A-18 [32]	92.86 (-1.32)	11.29 (↑ 10.11×)	2.89(↑ 4.9×)	24(-5)	10.8 (↑ 33.3%)
DRN-B-26 [32]	93.51 (-0.67)	21.27 (↑ 26.18×)	8.25(↑ 14.2×)	9(-20)	27.4 (↑ 238%)
ResAttNet-56 [3]	93.34 (-0.84)	31.90 (↑ 17.64×)	6.28(↑ 10.8×)	16(-13)	15.6 (↑ 92.6%)
ResAttNet-92 [3]	93.98 (-0.10)	51.30 (↑ 28.34×)	10.4(↑ 17.9×)	9(-20)	28.1 (↑ 247%)
MobileNetV2 [20]	90.93 (-3.25)	3.51 (↑ 1.93×)	0.31(↓ 1.8×)	36(+5)	7.1 (↓ 12.3%)
MobileNetV3Large [9]	91.48 (-2.70)	5.48 (↑ 3.02×)	0.25(↓ 3.2×)	44(+13)	5.6 (↓ 30.8%)
MobileNetV4 [a]	92.56 (-1.62)	2.56 (↑ 1.42×)	0.2(↓ 2.9×)	48(+17)	5.2 (↓ 35.8%)
EfficientNetlite-B0 [1]	93.86 (-0.32)	3.45 (↑ 1.92×)	0.41(↓ 1.41×)	32(+1)	7.5 (↓ 7.4%)
CSP-DarkNet-Tiny [b]	92.45 (-1.73)	2.39 (↑ 1.32×)	0.58(↓ 1×)	30(-1)	8.4 (↑ 3.7%)
ShuffleNetV2 2.0× [17]	92.26 (-1.92)	5.78 (↑ 3.19×)	0.53(↓ 1.09×)	32(+1)	6.5 (↓ 19.7%)
ConvNext-A [13]	92.18 (-2.00)	3.49 (↑ 1.93×)	0.56(↓ 1.03×)	28(-3)	9.0 (↑ 11.1%)
ConvNextV2-A [29]	92.75 (-1.43)	3.49 (↑ 1.93×)	0.56(↓ 1.03×)	32(+1)	10.2 (↑ 25.9%)
MeDiANet$_{base}$-69	94.18	**0.38** (↓ 4.76×)	**0.08**(↓ 7.2×)	**60(+29)**	**4.2** (↓ 48.1%)
MeDiANet$_{wide}$-69	94.35 (+0.17)	0.91 (↓ 1.98×)	0.25(↓ 2.3×)	34(+3)	5.8 (↓ 28.4%)
MeDiANet$_{base}$-117	94.28 (+0.10)	0.63 (↓ 2.87×)	0.18(↓ 3.2×)	44(+13)	7.4 (↓ 8.6%)
MeDiANet$_{wide}$-117	**95.01 (+0.83)**	1.81	0.58	31	8.1

[a] https://github.com/tensorflow/models/blob/master/official/vision/modeling/backbones/mobilenet.py

[b] https://keras.io/api/keras_cv/models/backbones/csp_darknet/

In terms of accuracy, MeDiANet$_{base}$-69 has achieved 94.18% compared to 93.98% of ResAttNet-92 with 135× fewer parameter and 208× fewer FLOPs. The MeDiANet$_{wide}$-117 which is deeper and wider variant of MeDiaNet has demonstrated superior performance, achieving the highest accuracy of 95.01%. This represents an average improvement of 2.41% over other SOTA networks, while also utilizing 9.23× fewer parameters on average. The lightweight networks like MobileNetV2, MobileNetV3, MobileNetV4, EfficientNet, ESP-DarkNet, ShuffleNetV2, and ConvNext used lesser number of FLOPs as compared to best performing network (MeDiANet$_{wide}$-117). However, on average, these networks exhibited a 2.76% lower accuracy. On the other side, MeDiANet$_{base}$-69 on an average utilizes 12.49× lesser parameters as compared to SOTA network, with a trade-off of a 0.83% reduction in accuracy compared to the MeDiANet$wide$-117. Additionally, in terms of FLOPs and inference time, the MeDiANet$_{base}$-69 variant, on average, utilized 2.55× fewer FLOPs and achieved a 177% improvement in inference time relative to SOTA network. MeDiANet$_{base}$-69 has greater efficiency with approximately 7× faster in terms of inference time latency and has 566% of improved throughput than the SOTA network (ResAttNet-92).

While MeDiANet$_{wide}$-117 achieved the best accuracy, it is slightly slower than MobileNetv3-Large (1.4×) and MobileNetV2 (1.1×) during inference. By incorporating the pre-activated residual block into the proposed architecture, we are able to alleviate the gradient vanishing issue and improve the network's training process, resulting in enhanced performance. Furthermore, the proposed MediaNet offers a more efficient and lightweight deeper network architecture that achieves high performance while significantly reducing the number of trainable parameters.

6 Conclusion

This work presents a novel DNN architecture to enhance significant 2D biomedical image classification accuracy. This architecture is inspired by the Residual Attention Network [3], which combines the soft-attention mask with the feature map extracted from the trunk branch consisting of two residual blocks. We propose a modified architecture by implementing an `MDiRes`(·) block with the trunk and attention mask branches which provides significant advantage for handling diverse biomedical images. The proposed network achieves comparable performance with significantly fewer parameters concerning the other SOTA method available on the dataset, which is demonstrated in Table 2. The proposed network is validated on a large-scale and multi-modal medical image dataset. Since the number of parameters of MeDiANets is significantly less than the SOTA network makes its fast and robust to train a dataset in real-time. Apart from that biomedical image classification task, the proposed network can be used for other natural image classification.

Acknowledgments. The authors acknowledge computing supported by the KOSAMBE Cluster of Department of Electrical Engineering, IIT Kharagpur.

Competing Interests. The authors have no conflicts of interest to declare that are relevant to the content of this chapter.

References

1. Higher accuracy on vision models with EfficientNet-Lite (Jun 2024), https://blog.tensorflow.org/2020/03/higher-accuracy-on-vision-models-with-efficientnet-lite.html, [Online; accessed 7. Jul. 2024]
2. Acevedo, A., Merino, A., Alférez, S., Molina, Á., Boldú, L., Rodellar, J.: A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. Data in Brief **30** (2020)
3. Wang et al, F.: Residual attention network for image classification. In: Proc. Conf Computer Visi. patt recog. pp. 3156–3164 (2017)
4. Al-Antary, M.T., Arafa, Y.: Multi-Scale Attention Network for Diabetic Retinopathy Classification. IEEE Access **9**, 54190–54200 (2021). https://doi.org/10.1109/ACCESS.2021.3070685

5. Dewan, D., Ghosh, L., Chakraborty, B., Chowdhury, A., Konar, A., Nagar, A.K.: Cognitive analysis of mental states of people according to ethical decisions using deep learning approach. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2020)
6. Ghosh, L., Dewan, D., Chowdhury, A., Konar, A.: Exploration of face-perceptual ability by eeg induced deep learning algorithm. Biomed. Signal Process. Control **66**, 102368 (2021)
7. 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)
8. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European conference on computer vision. pp. 630–645. Springer (2016)
9. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1314–1324 (2019)
10. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks **3361**(10), 1995 (1995)
11. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
12. Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D.D., Chen, M.: Medical image classification with convolutional neural network. In: 2014 13th international conference on control automation robotics & vision (ICARCV). pp. 844–848. IEEE (2014)
13. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)
14. 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)
15. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: Int. Conf. on Learn. Representations (2017)
16. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Int. Conf. on Learn. Representations (2019)
17. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131 (2018)
18. Misra, D.: Mish: A self regularized non-monotonic activation function. British Machine Vision Conference (2020)
19. Neupane, B., Horanont, T., Aryal, J.: Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis. Remote Sensing **13**(4), 808 (2021)
20. 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)
21. Sharma, V., Gupta, M., Kumar, A., Mishra, D.: Video processing using deep learning techniques: A systematic literature review. IEEE Access **9**, 139489–139507 (2021)
22. Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M.: Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging **35**(5), 1285–1298 (2016)

23. Srinivasu, P.N., SivaSai, J.G., Ijaz, M.F., Bhoi, A.K., Kim, W., Kang, J.J.: Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm. Sensors **21**(8), 2852 (2021)
24. Srivastava, A., Hari, A., Pratiher, S., Alam, S., Ghosh, N., Banerjee, N., Patra, A.: Channel self-attention deep learning framework for multi-cardiac abnormality diagnosis from varied-lead ecg signals. In: 2021 Computing in Cardiology (CinC). vol. 48, pp. 1–4. IEEE (2021)
25. Srivastava, A., Pratiher, S., Alam, S., Hari, A., Banerjee, N., Ghosh, N., Patra, A.: A deep residual inception network with channel attention modules for multi-label cardiac abnormality detection from reduced-lead ecg. Physiol. Meas. **43**(6), 064005 (2022)
26. Suganyadevi, S., Seethalakshmi, V., Balasamy, K.: A review on deep learning in medical image analysis. International Journal of Multimedia Information Retrieval **11**(1), 19–38 (2022)
27. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)
28. Tang, J., Zhang, T., Gong, Z., Huang, X.: High precision cervical precancerous lesion classification method based on convnext. Bioengineering **10**(12), 1424 (2023)
29. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: Convnext v2: Co-designing and scaling convnets with masked autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16133–16142 (2023)
30. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)
31. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (ICLR) (May 2016)
32. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 472–480 (2017)
33. Zare, M.R., Mueen, A., Awedh, M., Seng, W.C.: Automatic classification of medical x-ray images: hybrid generative-discriminative approach. IET Image Proc. **7**(5), 523–532 (2013)
34. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems **31** (2018)
35. Zhou, Y., Chang, H., Lu, Y., Lu, X.: CDTNet: Improved Image Classification Method Using Standard. Dilated and Transposed Convolutions. Appl. Sci. **12**(12), 5984 (2022). https://doi.org/10.3390/app12125984

A Data Augmentation Approach for Well Log Interpretation

Ziyi Zhu[1], Yaobin Wang[1], Xiaoyang Yu[1], Guojun Li[2], Guanwen Zhang[1](✉), and Wei Zhou[1]

[1] Northwestern Polytechnical University, Xi'an, China
guanwen.zh@nwpu.edu.cn
[2] China National Logging Corporation, Xi'an, China

Abstract. Well log interpretation involves analyzing geological structures and reservoir contents using well log data. The advancement of artificial intelligence has led to the wide-ranging application of deep learning in well log interpretation. However, due to cost limitations, only a small number of wells are typically surveyed and tested in each logging block, posing a significant challenge in achieving accurate interpretation under few-shot conditions. To tackle this challenge, we propose a data augmentation method that integrates time-domain and frequency-domain features of well log curves. This method aims to harness the characteristics of well log curves in both domains to enrich the limited data. Furthermore, we have developed a neural network architecture search space tailored to the few-shot well log data problem and demonstrated the effectiveness of the proposed time-domain and frequency-domain data augmentation for well log interpretation. Our method has shown the ability to achieve accuracies of up to 93.00% and 95.05% on well logging interpretation tasks when applying different augmentation methods to the data from different blocks. These performance indicators are comparable to the results of training with all training wells under the entire block.

Keywords: Well log curves · Few-shot learning · Time Domain · Frequency Domain · Data Augmentation

1 Introduction

Well log curves [9] are the data of various geological parameters at different depths recorded by sensors during the well log [13] process. The variation characteristics of their amplitude and shape can reflect important information such as geological structure, oil, gas, and water content. Combining deep learning with well logs interpretation can enable the automatic extraction of rich geological insights by learning the complex patterns in well log curves, significantly improving the accuracy and efficiency of well logs interpretation [3,23,30]. However, in actual practice, the high costs of oil and gas exploration often result in

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 185–200, 2025.
https://doi.org/10.1007/978-3-031-78389-0_13

only a small number of test wells being developed and tested in each geological block, and the differences in formation conditions across blocks can lead to large variations in parameters like formations and reservoirs. To promote the establishment of intelligent well logs interpretation processes, it is crucial to improve the accuracy of well logs interpretation models using limited well log data.

Few-shot learning [19] aims to fully utilize the limited labeled samples to learn the mapping relationship between input and output, in order to improve the model's predictive performance and generalization ability when facing new samples. In this context, data augmentation is a direct approach to solving the problem of few-shot well log. In previous research, data augmentation operations commonly used in the field of image classification[7,8,31,35], can effectively expand the training data and improve the model's predictive ability and generalization capability. In comparison, well log curve data, as discrete numerical sequences, differ significantly from image and text data, and cannot be directly applied to the aforementioned augmentation techniques. Existing research on well log data augmentation has primarily focused on two approaches: directly leveraging the time-domain characteristics of well log data at different depths, and utilizing wavelet transform [25] and other techniques to analyze the curves in the frequency domain. However, these studies have not simultaneously exploited the time-domain and frequency-domain features of well log data.

In this paper, we propose a well log data augmentation method based on the fusion of time-domain and frequency-domain features, aiming to address the challenge of few-shot well log learning. The method leverages the information of well log curves in both the time and frequency domains. In the time domain, it applies smoothing and fluctuation enhancement operations, while in the frequency domain, it applies Fourier transform, wavelet transform, and wavelet domain denoising. This enhances the diversity and representativeness of the data samples, thereby improving the performance of deep learning models in well log interpretation. The main contributions of this work are as follows:

1) In order to tackle the challenge of few-shot learning in well log interpolation, we design two time-domain and three frequency-domain data augmentation methods, as well as three strategies to integrate the time-domain and frequency-domain features. These methods effectively increase the diversity and representativeness of the data samples, thereby enhancing the performance of deep learning models in well log interpretation under limited sample conditions.

2) We integrate the proposed time-frequency feature fusion augmentation method with mainstream deep learning networks such as ResNet18 and SENet18. The experimental results demonstrate that the three proposed augmentation methods can significantly improve the model's prediction performance in few-shot well log learning. Specifically, our method can achieve accuracies of up to 91.66% and 94.93% on well log interpretation with the number of training wells being 2 and 4, respectively, which is comparable to the results of training with all training wells under the entire block.

3) We design a dedicated neural network architecture search space for the few-shot learning in well log, and combine it with the time-frequency feature

fusion data augmentation method. This further validates the effectiveness of time-frequency features in enhancing well log data modeling performance. The experimental results show that the architecture found through the search process can improve the highest accuracy of 2 and 4 training well data augmentation to 93.00% and 95.05% respectively.

2 Related Works

2.1 Data Augmentation

In the context of few-shot learning for classification tasks, there are three main approaches: data augmentation-based [12], metric-based [11], and meta-learning-based [21]. data augmentation [28] is a method that increases sample diversity by transforming or expanding training samples so that the model can better adapt to new sample distributions. Compared with the latter two, data augmentation is the most direct way to solve the limited sample problem.

When solving few-shot learning tasks based on data augmentation, researchers have employed various approaches, including manually designing augmentation techniques based on the characteristics of the data, exemplified by methods [16,33]. Additionally Kim et al. [18] proposed a technique that utilizes slot-based noise addition to convert data into short sentences with the same context but different slot labels, aiming to solve oral few-shot learning tasks. Furthermore, data augmentation can also be achieved through deep learning models [1,20,34] Chu et al. [4] presented a method to solve the long-tail data distribution problem [38], wherein they employed Class Activation Maps [37] to divide the samples into class-generic and class-specific features, enabling them to combine the common features of majority classes with the unique features of minority (long-tail) classes.

Building upon the previous approaches, researchers have further combined the power of Automated Machine Learning [5,14,15] with data augmentation techniques to automate the process of data augmentation for specific datasets and tasks. The Adversarial AutoAugment algorithm designed by Zhang et al. [36] in 2020 introduced the "adversarial" idea of GAN into AutoAugment, so that the model can simultaneously search for data augmentation strategies and train classification models. In addition, Cubuk et al. [6] improved the AutoAugment algorithm in 2020 and proposed the RandAugment algorithm, which can significantly reduce the search space and does not require an independent search phase.

2.2 Time-frequency Augmentation

Time-frequency analysis is a valuable tool in speech signal processing and geophysical data analysis, as it can provide richer information compared to time domain analysis alone. By combining time and frequency domain analysis, one can focus on the characteristics of data in both dimensions simultaneously, which is crucial for gaining a deeper understanding of the intrinsic properties of the data and improving model performance under limited sample conditions.

For audio data, in addition to directly applying time shifting and background sound mixing, common data augmentation methods can also generate more samples by performing spectral transformations. The SpecAugment algorithm proposed by Park et al. [27] enhances the spectrogram information of input speech samples by distorting the spectrogram in the time direction and applying signal masking in the time and frequency domains. Similarly, the SpecMix algorithm introduced by Kim et al. [17] uses a time-frequency masking strategy to maintain the spectral correlation between different audio samples while mixing them for data augmentation.Beyond these two approaches, wavelet transform [10,25,26] is also widely used for data enhancement.

In the field of geophysical logging, time-frequency analysis, such as short-time Fourier transform, continuous wavelet transform and other technologies [24,29] also play an important role in processing geophysical data such as seismic and magnetic data. Yu et al. [32] also optimized the wavelet decomposition parameters by integrating multi-scale wavelet transform technology when conducting Glutenite reservoirs.

3 Method

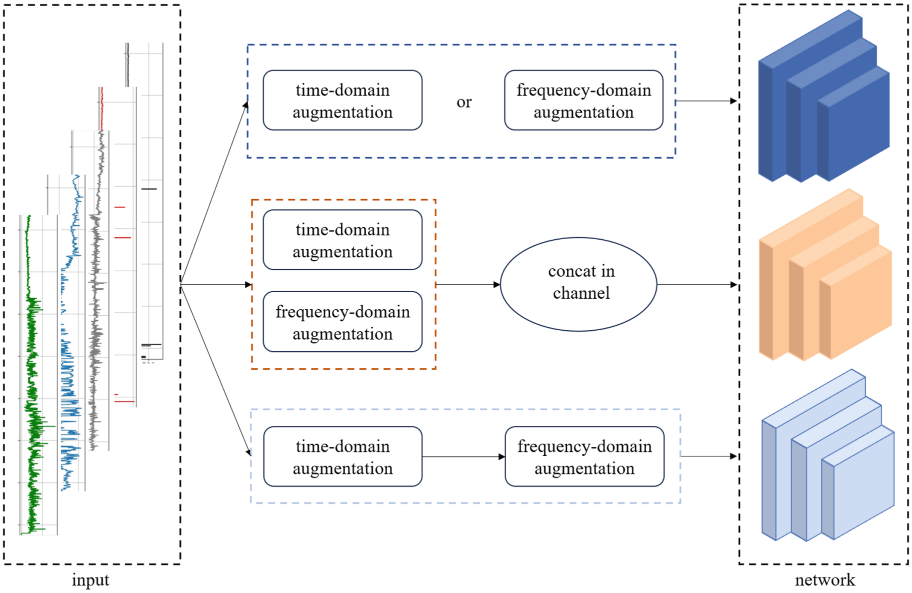


Fig. 1. The overall of our proposed method. To address the challenge of few shot learning for well log, we mainly proposed three methods to fuse the time domain and frequency domain information.

The overall structure of our proposed method for few-shot learning augmentation of well log data shows in Figure 1. We design three approaches to fuse the log data in these two domains: separate augmentation in the time and frequency domains, sequential augmentation in the time and frequency domains, and fusion of time and frequency domain features in the channel dimension.

The separate augmentation in the time and frequency domains involves enhancing the original curve in the time domain and performing multi-scale decomposition in the frequency domain, with the features enhanced in the time and frequency domains being used as model training data. When fusing time and frequency domain features in the channel dimension, after performing time and frequency domain augmentation operations on the original data, we first convert the frequency domain augmented data back to the time domain, and then concatenate the time domain augmented data and the frequency domain augmented data in the channel dimension to serve as the training data for the few-shot logging model. Besides, in the sequential augmentation in the time and frequency domains, we first perform denoising or fluctuation enhancement operations on the logging data in the time domain, and then convert the data augmented in the time domain to the frequency domain to complete the augmentation in the frequency domain.

3.1 Time-domain Method

Well logging data is data collected by sensors along the depth direction, which has obvious time domain characteristics. In the few-shot learning problem for well log, we design two time domain augmentation methods.

Smoothing Convolution The harsh downhole conditions and sensor limitations introduce noise and artifacts, leading to suboptimal feature extraction by the model, ultimately degrading the final interpretation results. To mitigate this challenge, we develop a time domain augmentation technique leveraging one-dimensional convolution operations to smooth the logging data. Figure 2 illustrates the process of applying a stack of convolutions to smooth the logging curve.

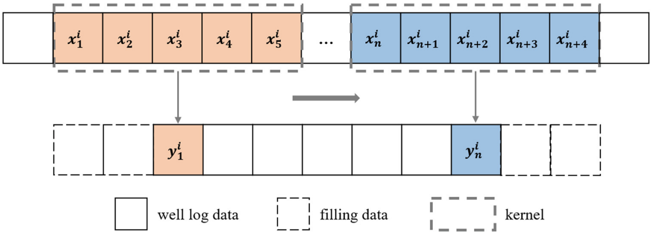


Fig. 2. Smoothing and denoising the well logging data using one-dimensional convolution.

We first select appropriate kernel size and stride parameters for the 1D convolution operation, and then perform this convolution processing along the depth dimension of each logging curve. Crucially, to maintain the fidelity of the original data distribution, we normalize the convolved log data by dividing it by the sum of the convolution kernel elements. Additionally, we employ a copy-and-fill strategy to ensure consistency in the data length before and after the augmentation process. The mathematical expression for the log data value after the convolution smoothing is given by Eq. 1, where m denotes the convolution kernel length, w_k represents the kernel element values, i indexes the individual logging curves, and x_{n+k} and y_k correspond to the original and augmented logging data, respectively.

$$y_n^i = \frac{\sum_{k=0}^{m} x_{n+k}^i w_k}{\sum_{k=0}^{m} w_k} \tag{1}$$

Fluctuation Enhancement The primary driver of fluctuations in well logging curves is the changes in formation properties. To encourage the deep learning model to focus more on the variations at the formation interfaces, we design a targeted data augmentation technique that amplifies the fluctuations in the logging curves.

This method first computes the maximum($x_{\max}^l$), minimum($x_{\min}^l$), and average (x_{mean}^l) values for each logging slice, where l denotes the different slices. We then selectively apply a fluctuation enhancement operation on the slices where the difference between the maximum and minimum exceeds the average value For these targeted slices, we scale up the logging data greater than the average by a factor of $(m+1)/m$, while scaling down the remaining data by a factor of $(m-1)/m$, where m represents the number of logging curves. The mathematical formulation of this process is presented in the following equation:

$$y_n^l = \begin{cases} \frac{(m+1)x_n^l}{m}, & x_n^l > x_{\text{mean}}^l \\ \frac{(m-1)x_n^l}{m}, & x_n^l \leq x_{\text{mean}}^l \end{cases} \tag{2}$$

3.2 Frequency-domain Method

In the few-shot learning scenario involving small log datasets, decomposing the log data into different frequency bands can help isolate the valuable low-frequency geological signals from the overall data. This separation of frequency components aids the deep learning model in better capturing the implicit geological features contained within the log curves. To address this few-shot learning problem, we develop three frequency domain data augmentation.

Fourier Transform In the problem of few-shot learning with limited log samples, Fourier transform [2] can be leveraged to convert the time-domain characterization signals into the frequency domain. This transformation allows analyzing the distribution of different frequency components within the signals, which is crucial for understanding the periodic information in the formation transformation and gaining insights into the formation structure at various scales. Since log experts primarily rely on peak changes during logging analysis, and the amplitude in Fourier transform corresponds to the fluctuation amplitude of different depth points, we perform data augmentation based on Fourier transform. Specifically, we obtain the phase and amplitude information in the frequency spectrum, and selectively retain only the amplitude data. The process of Fourier transform-based data enhancement for the value at depth point n in the i-th logging curve is shown in Eq.3 and Eq.4, where N represents the number of depth points contained in the logging slice that undergoes Fourier transformation.

$$X(k) = \sum_{n=0}^{N-1} x_n^i e^{-i2\pi kn/N} \tag{3}$$

$$Y(k) = |X(k)| \tag{4}$$

Wavelet Transform Wavelet transform is a method that can detect signal mutation points and describe the degree of signal mutation, which helps to capture the changing characteristics of the interface in logging data. Figure 3(a) is a schematic diagram of a two-level wavelet decomposition, where cD_1, cD_2 and cA_2 are the high-frequency detail coefficients and low-frequency components of the first and second layers after wavelet decomposition, and the lengths of the three are 1/2, 1/4, and 1/4 of the original logging data, respectively.

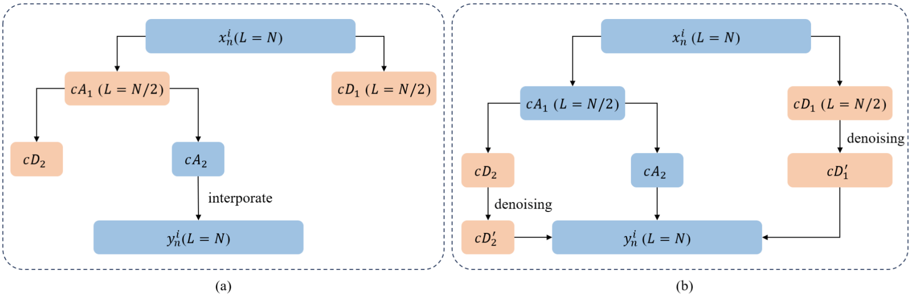


Fig. 3. The process diagram of wavelet transform(a) and wavelet domain denoising(b), where N is the curve length.

In the analysis of logging data, the high-frequency band corresponds to the rock formation details, which contain significant high-frequency noise. To reduce the impact of this noise when converting the logging curve to the frequency domain using wavelet transform, the low-frequency components of the wavelet variables are used for data augmentation. To ensure the consistency of the depth points before and after data augmentation, interpolation is applied to restore the wavelet components to the original data length.

Wavelet Domain Denoising To address the noise problem introduced in the process of log curve acquisition, in addition to convolutional smoothing denoising in the time domain, a wavelet domain denoising method is designed in the frequency domain. Fig.3(b) describes the process of wavelet domain denoising.

When performing wavelet domain denoising on the logging curve, the original curve is first decomposed by wavelet to obtain wavelet coefficients at different scales. The high-frequency components in the wavelet coefficients are then subjected to soft threshold filtering to remove the high-frequency noise components. Finally, the denoised wavelet coefficients are reconstructed to obtain the logging curve with the high-frequency noise removed. The soft threshold λ and filtering formula $y_n^i = \text{sign}(x_n^i)(|x_n^i| - \lambda)$ are used in the wavelet domain denoising process, where cD_1 is the first-layer detail coefficient of wavelet decomposition, N is the length of the original curve, j is the number of wavelet decomposition layers, and x_n^i and y_n^i are the values before and after high-frequency component denoising, respectively. The formula below represents the soft threshold λ and the filtering equation used in the wavelet-based denoising process for logging curves:

$$\lambda = \frac{\text{median}(|\text{cD}_1|)\sqrt{2\ln N}}{0.6745\log_2(j+1)} \tag{5}$$

$$y_n^i = \begin{cases} \text{sgn}(x_n^i)(|x_n^i| - \lambda), & |x_n^i| \geq \lambda \\ 0, & |x_n^i| < \lambda \end{cases} \tag{6}$$

4 Experiment

4.1 Dataset

Table 1. The number of appraisal and exploratory wells contained in each block. We use the appraisal wells for model training and the exploration wells for testing during the training process.

Block	Block-0	Block-1	Block-2	Block-3	Block-4	Block-5	Noise
Appraisal well num	23	92	7	6	12	57	3
Exploratory well num	0	13	3	0	2	9	1

In the experiments, we evaluate the performance of the above methods on oil wells from a specific region of the Changqing Oilfield. The dataset comprises 200 appraisal wells and 28 exploratory wells, which are used as training sets and test sets respectively during training in order to simulate the real exploration process. The relevant information for each well includes 6 well log curves (DEPTH, GR, AC, SP, RT1, RT2). As shown in Fig. 4, the location DBSCAN clustering of the dataset shows in an oil well distribution map, where the training and test wells can be partitioned into 5 and 4 blocks, respectively. The well counts within each blocks are provided in Table 1.

When conducting few-shot learning on this dataset, the training and test wells should be sampled from the same geological blocks. Besides, To ensure the well log data conforms to the input requirements of the convolutional neural network, we first apply a sliding window approach to segment each well log into fixed-size slices. Specifically, we use a slice length and sliding step of 96 depth points, corresponding to 12 meters in vertical depth without overlap between slices.Furthermore, we select the middle point of each slice, i.e., the 47th depth point, as the label for the entire slice.

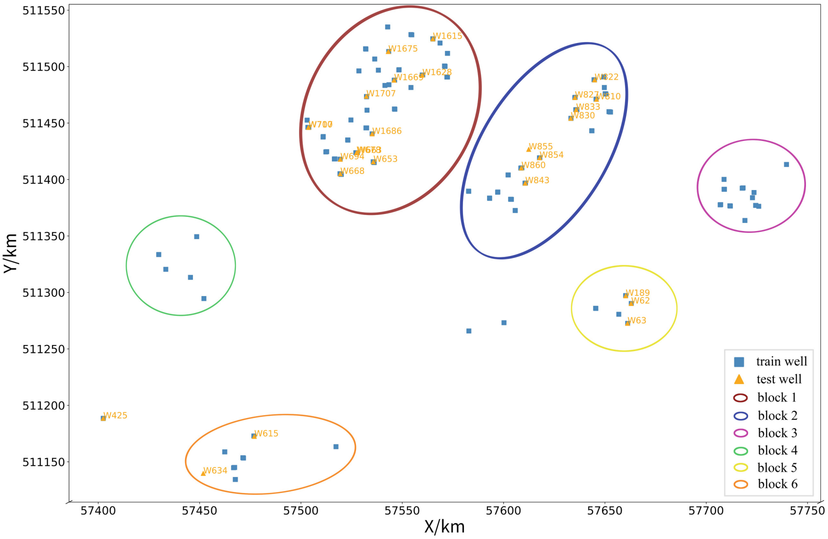


Fig. 4. The information of dataset block. In addition to the training and test sets divided into six blocks, the dataset also includes three training wells and two test wells that are not assigned to any specific block.

4.2 Experimental Setting

In evaluating the proposed method on the well log dataset, in addition to the classic ResNet18 and SENet networks, we also design a lightweight neural network architecture for the small-sample well log problem, referencing the DARTS [22]. The architecture search space includes one-dimensional convolutions with kernel sizes of 1, 3, 5, 7, and 11, one-dimensional dilated convolutions with a kernel size of 3 but dilation rates of 2 and 3, as well as a two-layer one-dimensional residual structure. For the convolution operations, we employ the ReLU-Conv-BN sequence. During the training process, we first identify the optimal Cell structure within the search space based on different data augmentation strategies, and then proceed to train the model using this Cell structure. The number of stacked Cell layers in both search and training phases is set to 4, while the other settings referenced the DARTS configuration. Additionally, all algorithms are implemented using the PyTorch framework. To ensure the fairness of the experimental results, we employ specific experimental settings as follows: a batch size of 1024, 100 training epochs, an initial learning rate of 5×10^{-4}, Adam optimizer and an exponential learning rate decay strategy with a decay factor of 0.98.

4.3 Experimental Results

To evaluate the effectiveness of the method we proposed for time-domain and frequency-domain data augmentation, we conduct stratigraphic segmentation tasks on well logging blocks.

We first test the model's performance directly on Block-1 and Block-4 without any data augmentation(Table 2, Table 3) as baseline, and the results show that the model achieves maximum performances of 90.77% and 94.46% on these two blocks without data augmentation.

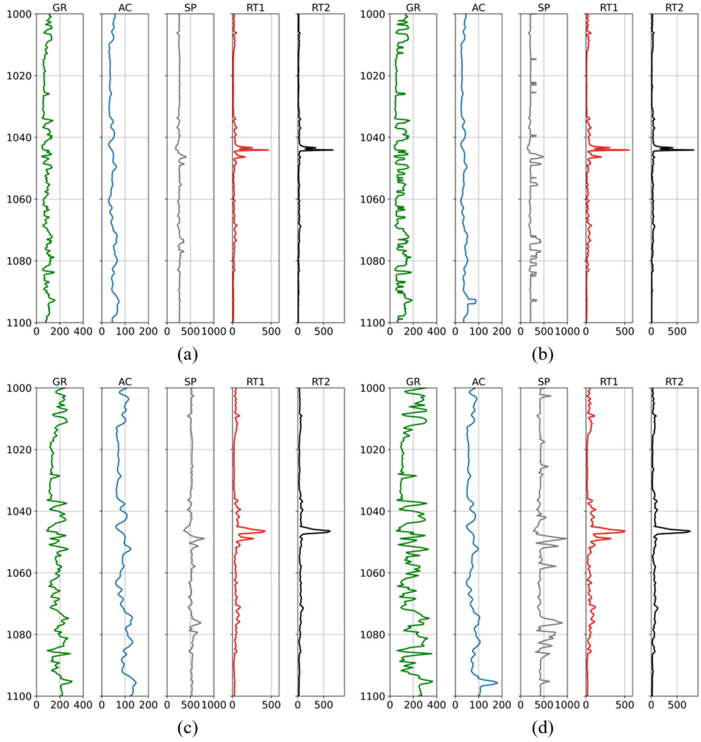
Table 2. Top-1 accuracy(% ↑) obtained by different models on Block-1 without data augmentation.

Model	2	4	6	10	50	92
ResNet18	82.02	83.17	85.48	87.56	89.64	90.24
SENet18	83.24	83.96	84.58	87.38	89.82	**90.77**
NAS	85.51	87.58	86.30	87.93	89.99	90.12

Table 3. Top-1 accuracy(% ↑) obtained by different models on Block-4 without data augmentation

Model	2	4	6	8	10	12
ResNet18	82.49	88.41	89.13	89.86	93.84	**94.46**
SENet18	86.47	89.41	89.49	90.58	92.39	93.84
NAS	88.59	90.94	90.70	92.87	92.87	93.40

As a comparison, Table 4 shows the data augmentation results under different models when 2 and 4 wells are selected as training sets, and all exploratory wells in the block are selected as test sets in Block-1. And Table 5 shows the experimental results of Block-4. TD, FD, TFS, T+F, TFC represent time domain, frequency fomain, time domain and frequency domain data augmentation performed separately, time domain and frequency domain data augmentation performed successively, and time domain and frequency domain data augmentation fused on the channel, respectively. A, B, C, D, E represent smoothing convolution, fluctuation enhancement, fourier transform, wavelet trainsform, wavelet domain denoising respectively. 2, 4 represent the number of training wells. The results indicate that the frequency-domain enhancement methods(FFT and wavelet denoising) outperformed the time-domain enhancement method. Furthermore, the three time-domain and frequency-domain enhancement and combination methods we proposed achieved better performance than using a single enhancement method. Specifically, the data augmentation approach combining time-domain and frequency-domain features improved the model accuracy for the stratigraphic segmentation task on Block-1 by up to 6.69% compared to using the original data directly. Compared with the best results when using only a single augmentation method, the combined time-domain and frequency-domain approach improved the accuracy by up to 2.95%. Additionally, when using the network architecture discovered through neural architecture search (NAS), the performance was further improved, with the accuracy of the geological stratification reaching up to 90.70%.



(a)

(b)

(c)

(d)

Fig. 5. Visualization of the original curve and the curve after data augmentation. The four subfigures depict: (a) the original curve, (b) the curve after time-domain fluctuation enhancement, (c) the curve after wavelet transform, (d) the curve obtained by first applying fluctuation enhancement and then wavelet transform.

Table 4. Top-1 accuracy(% ↑) of different data augmentation methods using ResNet18, SENet18 and NAS models in **Block-1**.

Method		ResNet18		SENet18		NAS	
		2	4	2	4	2	4
Baseline		82.02	83.17	83.24	83.96	85.51	87.58
TD	A	82.09	82.66	83.60	83.86	86.09	88.15
	B	82.73	83.28	83.23	85.40	85.64	88.71
FD	C	85.07	85.54	85.35	85.95	89.07	88.69
	D	85.74	86.64	84.61	84.94	87.35	88.14
	E	82.23	83.30	84.11	83.83	86.38	87.30
TFS	A+C	**88.71**	89.08	**87.20**	**88.71**	89.04	89.90
	A+D	86.62	87.87	85.61	85.62	86.55	89.84
	A+E	83.08	83.43	83.86	84.71	87.04	88.31
	B+C	88.28	**89.59**	85.79	86.75	90.26	89.39
	B+D	85.58	86.35	86.48	85.62	87.13	89.32
T+F	A+C	87.19	87.94	86.93	85.02	**90.70**	89.14
	A+D	86.38	86.77	84.49	84.69	87.43	88.25
	A+E	82.54	84.17	83.73	84.22	86.49	88.32
	B+C	87.26	88.24	86.28	86.23	90.16	**90.60**
	B+D	85.91	86.60	85.84	85.87	87.77	89.53
TFC	A+E	83.99	85.10	84.36	86.27	86.98	88.56

In Figure 5, an oil well from Block-4 with a depth segment of 1000-1100 was randomly selected, and its original curve, the curve after time-domain fluctuation enhancement, the curve after wavelet transformation, and the result of applying time-domain fluctuation enhancement followed by retaining only the low-frequency component through wavelet transformation were visualized. The results show that while time-domain fluctuation enhancement increases the amplitude of the curve fluctuations, it also proportionally increases the noise. Conversely, retaining only the low-frequency component through wavelet transformation reduces the high-frequency noise but also alters the waveform shape. Compared to these two approaches, the wavelet transform of the curve after fluctuation enhancement can retain the fluctuation amplitude of the interface while removing the high-frequency noise in the curve, further improving the prediction accuracy of the model in the few-shot learning settings.

Table 5. Top-1 accuracy(% ↑) of different data augmentation methods using ResNet18, SENet18 and NAS models in **Block-4**.

Method		ResNet18		SENet18		NAS	
		2	4	2	4	2	4
Baseline		82.49	88.41	86.47	89.41	88.59	90.94
TD	A	82.61	89.37	86.39	90.10	88.41	92.58
	B	82.71	88.68	86.83	90.58	88.95	90.79
FD	C	85.51	87.09	89.01	89.42	90.55	91.07
	D	86.83	89.13	88.77	89.86	90.19	91.41
	E	82.97	88.47	86.78	89.61	88.29	89.91
TFS	A+C	**91.66**	**94.32**	88.53	**94.93**	91.55	93.97
	A+D	85.75	92.75	90.10	92.87	91.37	93.00
	A+E	82.97	90.75	86.35	91.18	89.38	90.34
	B+C	88.54	89.26	88.16	88.83	90.46	91.79
	B+D	87.20	89.62	88.65	90.10	91.07	92.88
T+F	A+C	91.06	90.40	**90.46**	89.48	94.33	**95.05**
	A+D	91.30	89.49	**90.46**	90.58	91.91	92.33
	A+E	83.85	88.94	88.29	90.58	88.65	91.52
	B+C	91.06	91.40	89.25	89.09	**93.00**	93.96
	B+D	89.49	91.30	88.98	90.70	92.45	93.36
TFC	A+E	85.38	91.31	86.63	90.70	89.86	91.16

5 Conclusion

In this paper, we propose a data augmentation method based on the fusion of time domain and frequency domain features to address the problem of few-shot log data. By fusing the logging curves after denoising and increasing the fluctuation amplitude in the time domain with the well log curves after multi-scale decomposition and enhancement in the frequency domain, the richness and representativeness of the limited data are improved. Additionally, as there is no deep learning model specifically designed for the logging problem, we also design an architecture search space for the problem of limited log samples, which can search for the most suitable model parameters for the logging data sets augmented in different ways. The experimental results prove the effectiveness of our proposed method.

Acknowledgements. This work was supported in part by the National Key R&D Program of China (2018AAA0102801 and 2018AAA0102803), the National Natural Science Foundation of China (61772424, 61702418, and 61602383), and the project of China National Logging Corporation(2023ZZ05-04).

References

1. Alfassy, A., Karlinsky, L., Aides, A., Shtok, J., Harary, S., Feris, R., Giryes, R., Bronstein, A.M.: Laso: Label-set operations networks for multi-label few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6548–6557 (2019)
2. Brigham, E.O., Morrow, R.: The fast fourier transform. IEEE spectrum **4**(12), 63–70 (1967)
3. Chang, J., Kang, Y., Zheng, W.X., Cao, Y., Li, Z., Lv, W., Wang, X.M.: Active domain adaptation with application to intelligent logging lithology identification. IEEE Transactions on Cybernetics **52**(8), 8073–8087 (2021)
4. Chu, P., Bian, X., Liu, S., Ling, H.: Feature space augmentation for long-tailed data. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16. pp. 694–710. Springer (2020)
5. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 113–123 (2019)
6. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 702–703 (2020)
7. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
8. Fadaee, M., Bisazza, A., Monz, C.: Data augmentation for low-resource neural machine translation. arXiv preprint arXiv:1705.00440 (2017)
9. Fan, P., Deng, R., Qiu, J., Zhao, Z., Wu, S.: Well logging curve reconstruction based on kernel ridge regression. Arab. J. Geosci. **14**, 1–10 (2021)
10. Fujieda, S., Takayama, K., Hachisuka, T.: Wavelet convolutional neural networks for texture classification. arXiv preprint arXiv:1707.07394 (2017)
11. Guan, H., Michael, S.: Cobnet: Cross attention on object and background for few-shot segmentation. In: 2022 26th International Conference on Pattern Recognition (ICPR). pp. 39–45. IEEE (2022)
12. Guo, D., Kim, Y., Rush, A.M.: Sequence-level mixed sample data augmentation. arXiv preprint arXiv:2011.09039 (2020)
13. He, M., Gu, H., Wan, H.: Log interpretation for lithology and fluid identification using deep neural network combined with mahakil in a tight sandstone reservoir. J. Petrol. Sci. Eng. **194**, 107498 (2020)
14. He, X., Zhao, K., Chu, X.: Automl: A survey of the state-of-the-art. Knowl.-Based Syst. **212**, 106622 (2021)
15. Ho, D., Liang, E., Chen, X., Stoica, I., Abbeel, P.: Population based augmentation: Efficient learning of augmentation policy schedules. In: International conference on machine learning. pp. 2731–2741. PMLR (2019)
16. Inoue, H.: Data augmentation by pairing samples for images classification. arXiv preprint arXiv:1801.02929 (2018)
17. Kim, G., Han, D.K., Ko, H.: Specmix: A mixed sample data augmentation method for training withtime-frequency domain features. arXiv preprint arXiv:2108.03020 (2021)
18. Kim, H.Y., Roh, Y.H., Kim, Y.G.: Data augmentation by data noising for open-vocabulary slots in spoken language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop. pp. 97–102 (2019)

19. Lake, B.M., Salakhutdinov, R.R., Tenenbaum, J.: One-shot learning by inverting a compositional causal process. Advances in neural information processing systems **26** (2013)
20. Li, K., Zhang, Y., Li, K., Fu, Y.: Adversarial feature hallucination networks for few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13470–13479 (2020)
21. Lin, J., Wu, Z., Lin, W., Huang, J., Luo, R.: M2sd: Multiple mixing self-distillation for few-shot class-incremental learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 3422–3431 (2024)
22. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
23. Liu, J.J., Liu, J.C.: Integrating deep learning and logging data analytics for lithofacies classification and 3d modeling of tight sandstone reservoirs. Geosci. Front. **13**(1), 101311 (2022)
24. Mohammadi, A.K., Mohebian, R., Moradzadeh, A.: High-resolution seismic impedance inversion using improved ceemd with adaptive noise. J. Seism. Explor. **30**(5), 481–504 (2021)
25. Morlet, J., Arens, G., Fourgeau, E., Glard, D.: Wave propagation and sampling theory-part i: Complex signal and scattering in multilayered media. Geophysics **47**(2), 203–221 (1982)
26. Oyelade, O.N., Ezugwu, A.E.: A novel wavelet decomposition and transformation convolutional neural network with data augmentation for breast cancer detection using digital mammogram. Sci. Rep. **12**(1), 5913 (2022)
27. Park, D.S., Chan, W., Zhang, Y., Chiu, C.C., Zoph, B., Cubuk, E.D., Le, Q.V.: Specaugment: A simple data augmentation method for automatic speech recognition. arXiv preprint arXiv:1904.08779 (2019)
28. Royle, J.A., Dorazio, R.M., Link, W.A.: Analysis of multinomial models with unknown index using data augmentation. J. Comput. Graph. Stat. **16**(1), 67–85 (2007)
29. Tary, J.B., Herrera, R.H., Han, J., van der Baan, M.: Spectral estimation-what is new? what is next? Rev. Geophys. **52**(4), 723–749 (2014)
30. Wu, Q., Li, Z., Wang, Y., Cao, C., Qiao, B., Huang, Y., Yu, X.: Combination of seismic attributes using clustering and neural networks to identify environments with sandstone-type uranium mineralization. Acta Geophys. **71**(6), 2715–2731 (2023)
31. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10687–10698 (2020)
32. Yu, Z., Wang, Z., Wang, J.: Continuous wavelet transform and dynamic time warping-based fine division and correlation of glutenite sedimentary cycles. Math. Geosci. **55**(4), 521–539 (2023)
33. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
34. Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., Song, Y.: Metagan: An adversarial approach to few-shot learning. Advances in neural information processing systems **31** (2018)
35. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. Advances in neural information processing systems **28** (2015)
36. Zhang, X., Wang, Q., Zhang, J., Zhong, Z.: Adversarial autoaugment. arXiv preprint arXiv:1912.11188 (2019)

37. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2921–2929 (2016)
38. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9719–9728 (2020)

TSAK: Two-Stage Semantic-Aware Knowledge Distillation for Efficient Wearable Modality and Model Optimization in Manufacturing Lines

Hymalai Bello[1,2](✉), Daniel Geißler[1], Sungho Suh[1,2], Bo Zhou[1,2], and Paul Lukowicz[1,2]

[1] German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
hymalai.bello@dfki.de

[2] Department of Computer Science, RPTU Kaiserslautern-Landau, Kaiserslautern, Germany

Abstract. Smaller machine learning models, with less complex architectures and sensor inputs, can benefit wearable sensor-based human activity recognition (HAR) systems in many ways, from complexity and cost to battery life. In the specific case of smart factories, optimizing human-robot collaboration hinges on the implementation of cutting-edge, human-centric AI systems. To this end, workers' activity recognition enables accurate quantification of performance metrics, improving efficiency holistically. We present a two-stage semantic-aware knowledge distillation (KD) approach, TSAK, for efficient, privacy-aware, and wearable HAR in manufacturing lines, which reduces the input sensor modalities as well as the machine learning model size, while reaching similar recognition performance as a larger multi-modal and multi-positional teacher model. The first stage incorporates a teacher classifier model encoding attention, causal, and combined representations. The second stage encompasses a semantic classifier merging the three representations from the first stage. To evaluate TSAK, we recorded a multi-modal dataset at a smart factory testbed with wearable and privacy-aware sensors (IMU and capacitive) located on both workers' hands. In addition, we evaluated our approach on OpenPack, the only available open dataset mimicking the wearable sensor placements on both hands in the manufacturing HAR scenario. We compared several KD strategies with different representations to regulate the training process of a smaller student model. Compared to the larger teacher model, the student model takes fewer sensor channels from a single hand, has 79% fewer parameters, runs 8.88 times faster, and requires 96.6% less computing power (FLOPS). Our results show that with TSAK distillation, the efficient model has significantly improved in recognition performance compared to the model trained without TSAK, with up to 10% higher F1 score.

Keywords: Knowledge Distillation · Multimodal Fusion · Work Activity Recognition · Inertial Sensing · Capacitive Sensing

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 201–216, 2025.
https://doi.org/10.1007/978-3-031-78389-0_14

1 Introduction

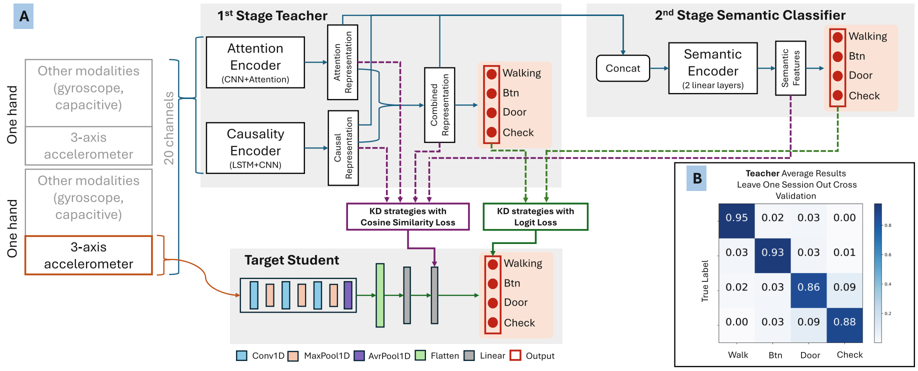


Fig. 1. (A) In the TSAK knowledge distillation (KD) approach, five distillation methods were compared. The cosine similarity loss distills knowledge from one of the hidden vectors at a time; Attention Representation (Attn-Rep), Causal Representation (Causal-Rep), and Combined Representation (Combi-Rep). A shallow classifier is employed to merge and distill knowledge from all the hidden vectors simultaneously, preserving the semantics of the ground truth by logit-based KD (Semantic Classifier). Logit KD is also performed with the teacher's outputs. The output categories are walking, touching screen/buttons (Btn), opening/closing the door (Door), and working inside the factory module (Check). (B) Teacher average results with an F1 score of 85.91% across twelve users.

The utilization of multi-modal sensing approaches, capturing the diversity of human behavior, has been widely leveraged to improve the accuracy and robustness of human activity recognition (HAR) systems. The fusion of multimodal and multipositional information increases performance in the case of complementary sources, and it is robust to perturbations that may affect a particular sensing modality [2]. However, most existing works prioritize improving the recognition performance while compromising efficiency, as smaller models with fewer sensor modality inputs usually suffer accuracy degradation. Compared to other domains such as language and vision, wearable systems such as smartwatches or fitness trackers, are designed to be worn on the body for extended periods, making energy efficiency a critical factor. Prolonged battery life is essential to ensure user acceptance and adherence, as frequent recharging can be cumbersome and may lead to device abandonment. Furthermore, wearables often rely on small, low-power processors incapable of handling computationally intensive tasks. However, most existing works leveraging multi-modal sensor fusion with machine learning techniques in pursuit of better accuracy, such as early or late fusion, transfer learning (TF), knowledge distillation (KD) [7,14,21,28], and contrastive learning (CL) [12,33], have ignored efficiency aspects such as model compression, reduced complexity, and latency-awareness.

Although KD in particular has the potential to compress knowledge from a teacher model to a smaller learner model, studies, including ours, have shown that naïve KD with a small learner model has limited improvement compared to training the small model without KD, especially when the teacher model has more modalities and input channels. An underlying contributing factor could be the semantic structural differences between the teacher and student making the representations from the teacher model less relevant for the student architecture. We propose, TSAK, a Two-stage Semantic-Aware Knowledge distillation focusing on training efficient and fast student models with enhanced recognition performance, as shown in Fig. 1. In the first stage, we train a teacher model with both self-attention and LSTM branches extracting the attention and causal information. In the second stage, the attention, causal, and combined representations are merged to feed into a semantic classifier. The student model is a much smaller and more efficient model with simple convolution and linear operations, taking only 3 channels of sensor input (compared to up to 20 channels for the teacher). To test the TSAK approach, we used the public dataset OpenPack [34] and our dataset at a smart factory testbed with smart gloves from 12 participants. The **main contributions** of this work are described as follows:

- TSAK enhances a lightweight single-position accelerometer-only model (3-axis) for **wearable HAR in manufacturing lines** using two-stage KD from a larger multimodal and multipositional teacher model.
- Through ablation studies of different KD strategies including latent representations and logits [1,8,15,16], we have found out that using the second stage semantic classifier's logits output as the KD regularization improves the student performance significantly.
- Experimental results with the OpenPack dataset and the dataset collected in this work show that TSAK increases the student F1 score by 5.4%, and 10.5%, respectively, with the student model being 79.0% smaller, running 8.88 times faster, and 96.6% less computation demanding (FLOPS).

2 Related Work

Representation learning has been extensively studied for HAR applications. CL across inertial measurement unit (IMU) locations for wearable HAR was proposed in [12] to learn a hidden representation from IMU sensors at different locations, but at the time of inference only use data from a single IMU. Each IMU has its encoder and the objective is to guide the target encoder to learn from the best encoder, i.e., from the most informative IMU position. An improvement of F1 between 5% to 13% using PAMAP2 and Opportunity Dataset was obtained. This is an unimodal transfer learning strategy with no efficiency improvement in the target model. In [21], a KD method with multimodal fusion from audio to inertial sensors was proposed to transfer audio context information to the motion data for HAR, achieving an increase of 4.5% F1 score for 23 activities (72.4%). Their student employs acceleration and gyroscope data with separate branches of DeepConvLSTM [7] for each input type, leading to a student with 3.6 million parameters and high complexity due to the stack of LSTM layers. VAX in [28] is a cross-modality transfer learning where the video/audio (VA) is the teacher modality and X is a privacy-aware sensor from a selection of pervasive devices. Video/audio models are widely spread and trained with labeled data, alleviating the lack of training data with X sensors under the guidance of the VAX teacher. An improvement of 5% in accuracy compared to the baseline was obtained.

HAR with KD opens the possibility of transferring information from a cumbersome teacher model to a lightweight target-student model. A capable, smaller learner offers advantages, such as less memory, less power, and lower latency. KD has shown incredible potential in computer vision [32]. The student can be guided by multiple methods, including logit-based, feature-based, and cross-modality among others [14]. However, most of these methods have not yet been evaluated in wearable datasets for HAR in manufacturing lines. Table 1 reviews the state-of-the-art (SOTA) of engineering applications using cross-modality KD for wearable HAR. Overall, the SOTA shows that most solutions use complex structures for both the teacher and student network. For example, using deep convolutional LSTMs [21] and transformers [23]. These high complexity and high-performance networks are not yet supported by low memory, low power microcontrollers with limited FLOPS compared to a GPU [27]. In addition, the size difference between the teacher network and student network has to be moderate to increase the performance of the target model [24]. To the best of our knowledge, among the SOTA, there are no solutions with model enhancement through KD that produce target student models efficient enough for microcontroller deployment for wearables, which is one of the focuses of TSAK.

Table 1. Engineering applications using cross-modality knowledge distillation for wearable HAR

Study	Accessory	Modality	Application	Improvement	KD Type
Liang [21]	Watch	Audio to (Acc+Gyro)	Daily Living	4.5% F1	Logit
J Ni [25]	Wristband	Skeleton to Acc	Body Motion	4.99% Acc	Logit
Liu [22]	NA	Inertial to Videos	Body Motion	4.01% Acc	Feat
Ni [26]	Wristband	Videos to Acc	Body Motion	2.1% Acc	Feat
Liu. Y[23]	NA	EEG to GSR*	Emotions	3.41% F1	Logit+Feat
Ours	Gloves	(IMU+Cap) to Acc**	**Factory**	**5.4-10.5% F1**	Fig. 1

*Electroencephalogram (EEG), Galvanic Skin Response (GSR). **Capacitive (Cap)

3 Apparatus and Dataset

Apparatus: We use a glove-based system to monitor HAR in a smart factory environment [4]. The system backbone is the Adafruit Feather Sense development board with an Arm Cortex M4. Two gloves are worn by the participants as shown in Fig. 2. Each glove has an IMU and four textile capacitive channels. Inside the Feather Sense, the BLE communication is handled by the Nordic nRF52832. The capacitive channels consist of four conductive thin patches distributed on the index, thumb, little finger, and around the wrist. Moreover, the IMU-selected placement is on the wrist. This approach reduces the number of connections, and flexibility and comfort are considered. Noticeably, the gloves do not cover the entire area of the fingers, minimally affecting the user's mobility. Only the inertial (accelerometer and gyroscope) and capacitive data are used. For a total of 10 channels per glove. Capacitive sensing has been used in textile designs such as neckbands [9], jackets [5,6], pants [13] and particularly gloves [3]. The textile capacitive sensor is based on the state-of-the-art capacitance-to-digital converter (CDC) FDC2214 following the design in [3]. The excitation frequency of the CDC is set at 13.7 MHz with an 18uH external inductor and 33pf capacitor for each channel, operating with single-end sensing mode at a 50 Hz sampling rate. Four channels of long electrodes (e-textile Shieldex Technik-tex P130+B) with the dimensions 0.55 mm wide and between 11-15 cm long were thermally bound from the sensing module to 3 fingers (the first, second, and fifth digits) and around the wrist.

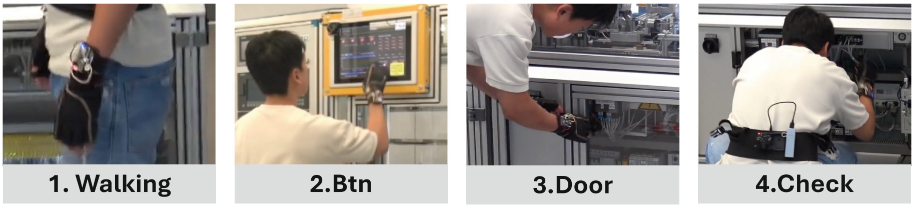


Fig. 2. Activities dictionary in the smart factory testbed

Dataset: There are multiple datasets with wearable sensors, e.g. PAMAP2, Opportunity, WISDM, and RealWorld, among others [17]. None of these human activity datasets meet the requirements to be representative and add completeness to the manufacturing line scenario. Moreover, publicly available sensor datasets in industrial settings are limited by difficulties in collecting realistic data, thus requiring close collaboration with industrial sites. The Fraunhofer Institute has made publicly available[1] a list of more than 120 open datasets from production environments, none of them related to human activity recognition and wearable devices. In HA4M [11], the first dataset about an assembly task with multiple vision sensors is introduced. The authors of HA-ViD [35] and IndustReal [29] also collected a human assembly video dataset, but still lack the wearable sensors for HAR. To address these challenges and contribute to research on HAR in industrial settings, OpenPack [34] recently introduced a dataset for packaging work recognition. In addition, in this work we have collected data

[1] https://www.bigdata-ai.fraunhofer.de/s/datasets/index.html.

from wearable sensors in a smart factory testbed, aiming to add completeness and relevance to our evaluation method. Our dataset selection criteria are based on three aspects: 1. HAR in manufacturing lines; 2. wearable, privacy-aware multimodal sensors; 3. sensor position on both hands. This led us to collecting our own dataset, called Smart Factory Dataset, which is then complemented by the OpenPack dataset for secondary evaluation of our approach.

Smart Factory Dataset: Twelve volunteers were recruited. They identify themselves as ten male and two female. Their age ranges from 23 to 59 years old (mean of 30.75). The height ranges from 160-184 centimeters (mean 178 cm). Only one of the participants was left-handed. The participants wore two gloves equipped with inertial and textile capacitive sensing. At the beginning of the experiment, the sensors' data is synchronized in front of a video camera. The camera time is then used as a global clock to synchronize the data from the two gloves. The volunteers were asked to walk around the factory modules and simulate working activities on the factory floor. Fig. 2 depicts the activities performed by each volunteer were categorized as walking (1. Walk), touching the screen/buttons (2. Btn), opening/closing the door (3. Door), and working inside the factory module (4. Check). The activities were performed on each module (in total 6) in each session. A total of five sessions per participant were recorded. In between every session, the hardware was removed from the wearer and a rest of ten to twenty minutes was enforced. This makes the results accountable for the re-wearing of the system, which is typically expected in wearable devices. Each session lasted around 20 minutes on average. One participant performed two sessions one day and three sessions another. One volunteer only performs three sessions in total. The participant with less than five sessions was only included in the training set. For eleven participants, the data is split into 4 sessions for training and 1 session for testing. A 5-fold cross-validation with a leave-one-session-out evaluation scheme is performed. [2]

OpenPack (Public): The dataset contains packaging work activities in an industrial testbed, including work operations, actions, and outliers. The main reasons for selecting OpenPack for analysis are the semi-realistic industrial setting and the configuration of sensors on volunteers' wrists, similar to our smart factory dataset. It contains 53.8 hours of multimodal data, including key points, depth images, IMU data, and scarce readings from IoT devices (e.g., barcode scanners) [34]. We focus on the IMUs data (accelerometer and gyroscope) from the users's wrists (right/left) and for the case of work operations activities. The creators define the activities as 1. picking, 2. relocate item label, 3. assemble box, 4. insert items, 5. close box, 6. attach box label, 7. scan label, 8. attach shipping label, 9. put on back table, and 10. fill out. Due to the low granularity of the labeling procedure, the categories are mixed within different labels. This led us to merge them into four classes; Pick (1,9,10), label (2,6,8), Assemble (3,4,5), and Scan. The sampling rate for the sensors is around 30 Hz. The idea is to transfer knowledge from a multimodal teacher to a student with one-handed acceleration

[2] All participants signed an agreement following the policies of the university's committee for protecting human subjects and following the Declaration of Helsinki [30].

data as input. Furthermore, we use five of eleven users' data to avoid faulty data. A 5-fold cross-user validation with a leave-one-session-out evaluation scheme is performed with five users' data.

4 Knowledge Distillation Approach

Pre-Processing Factory: The three accelerations (m/s^2) and the four capacitive channels are normalized between zero and one. The angular velocity channels are kept in their original range (± 250 dps). This is followed by a 2-second resample window (100 samples at 50Hz). Then, a second-degree Butterworth low pass filter of 30 Hz was used to remove the jitter on the resampled signals. The resampling to 50Hz is for synchronization purposes with the video-ground truth with 50 frames per second. A sliding window of 2 seconds with a step size of 0.5 seconds is used. After pre-processing, the dataset structure is ten channels for each glove with a window size of 2 seconds and 25% overlapping. The ground truth of the worker's activity is extracted manually from the recorded videos.

Pre-Processing OpenPack: Acceleration and angular velocity data are resampled from 30 Hz to a 2-second resampling window (100 samples at 50 Hz) to match the factory dataset. The same 2-second sliding window and 25% overlap are used. The dataset structure is six channels for each IMU on the user's wrists.

Next, we train the teacher for the knowledge distillation to the student. The Pytorch 2.1.0 version is used to train the neural networks (NN). The evaluation scheme is defined as a 5-fold cross-validation with leave-one-session out. The training of the NNs ran for 100 epochs with early stopping (patience 10) to avoid overfitting. And, a 64-batch size was selected. The optimizer was Adadelta with a learning rate of 0.9. The loss function is categorical cross-entropy and the metric to monitor is accuracy.

Factory Teacher: We have trained a multimodal and multipositional teacher model. Inertial and capacitive channels from both gloves (left and right hand) were fused, for an input size of 20 channels. The teacher NN architecture details are in Table 2. The structure contains two branches. One branch of the neural networks is focused on extracting features of the cross-channel interaction between the modalities and sensors' positions (**TASmart**), resulting in the hidden vector **Attn-Rep**. And, the second branch extracts the causality of the multimodal time series input (**TCSmart**), leading to the hidden vector **Causal-Rep**. The causality extractor network is based on one-layer Long-Short-Term Memory (LSTM). Both networks are then concatenated (**Combi-Rep**) and fed into a classifier layer (Linear Layer), followed by an output layer with the softmax activation function. Combining the two concatenate NNs can capture spatial and temporal information, thus effectively solving complex time series problems. Table 3, shows the profiling information of the teacher structure. These three hidden vectors are defined to compare the feature-based knowledge distillation (KD) method using different embedded representations within the teacher's NN structure. The feature-based KD is compared with logit-based KD. Furthermore, a shallow classifier merges and distills knowledge from the three hidden zones simultaneously, preserving the semantics of the ground truth by logit-based KD (see Fig. 1). Fig. 1**B**, depicts the average results of the teacher for twelve volunteers and leave-one-session out cross-user validation scheme (F1 = 85.91%).

OpenPack Teacher: The factory teacher is modified to match the dataset as follows: 1. the input structure only includes the acceleration and angular velocity channels and 2. the teacher structure is modified accordingly. The implementation details of the NNs are shown in Table 2.

The Target Model: The idea is to explore the performance enhancement of small and simple networks that can be deployed in wearable and embedded devices with a reduced impact on power consumption and memory. The structure of the student/target NN is depicted in Fig. 1 (Bottom Left). The input layer consists of one-handed acceleration data. The NN combines a feature extraction and a classifier with two linear layers. The NN is a CNN-based model for compatibility with supported operation on wearable embedded devices. To avoid degradation of the student performance the gap between the teacher and the student has to be moderate [24]. The student model is 79% smaller, 8.88 times faster, and 96.6% less computation demanding (FLOPS) than the teacher's, providing an embedded and sustainable solution. The implementation details are in Table 2. The student's profile is compared with the teacher in Table 3.

TSAK Distillation Approach: The distillation approach is depicted in Fig. 1. TSAK approach compares five distillation methods independently. The cosine similarity loss distills knowledge from one of the hidden vectors at a time; **Attn-Rep**, **Causal-Rep**, and **Combi-Rep**. Moreover, a shallow classifier merged and distilled knowledge from the three hidden vectors of the teacher simultaneously, preserving the semantics of the ground truth by logit-based KD (Semantic Classifier). The semantic classifier is trained in an incremental stage. The frozen teacher is used to train the semantic classifier, which is subsequently also frozen and used to train the student. Logit-based KD is also performed with the teacher's soft outputs. The loss function $\mathcal{L}_{SC}$ for the logit-based (Logit) and semantic-based KD (SC-Logit) follows the Eq. (1). Where x is the input, W are the student model parameters, y is the ground truth label, $\mathcal{L}_{CE}$ is the cross-entropy loss function, $\mathcal{L}_{KL}$ is the KL-divergence loss, σ is the softmax function parameterized by the temperature T and α is a coefficient (weight). z_s and z_t are the logits of the student and teacher respectively. In our experiments, α varies between 0.1, 0.5, 0.99, and 0.99. τ values are 1, 4, and 20. This follows the experimental setting in [10]. Fig. 4 depicts a comparison for different α and temperatures. For the case of feature-based KD, the loss function $\mathcal{L}_F$ follows the Eq. (2), where $\mathcal{L}_{CSKD}$ is the cosine similarity loss, h_t and h_s are the hidden vectors of the teacher and student model, respectively.

$$\begin{aligned}\mathcal{L}_{SC}(x;W) = \alpha \times \mathcal{L}_{CE}(y, \sigma(z_s;T=1)) \\ + (1-\alpha) \times \mathcal{L}_{KL}(\sigma(z_t;T=\tau), \sigma(z_s, T=\tau))\end{aligned} \tag{1}$$

$$\mathcal{L}_F(x;W) = \alpha \times \mathcal{L}_{CE}(y, z_s) + (1-\alpha) \times \mathcal{L}_{CSKD}(h_t, h_s) \tag{2}$$

Table 2. Implementation details of the neural networks

Network	Layer	Details Kernel(K), Stride(S), Output(O)
TASmart*	Conv1; MaxPool; Dropout	K=3, S=1, O=100; K=2, S=2; 0.2
	Conv1; MaxPool; Dropout	K=3, S=1, O=20; K=2, S=2; 0.2
	Conv1; MaxPool; Dropout	K=3, S=1, O=10; K=2, S=2; 0.2
	Self Attention1	(10,10)
	Self Attention2	(6,6)
TCSmart**	LSTM; Dropout	(10,10); 0.2
TAPack***	Conv1; MaxPool; Dropout	K=3, S=1, O=100; K=2, S=2; 0.1
	Conv1; MaxPool; Dropout	K=3, S=1, O=20; K=2, S=2; 0.1
	Conv1; MaxPool; Dropout	K=3, S=1, O=10; K=2, S=2; 0.1
	Self Attention1	(10,10)
	Self Attention2	(4,4)
TCPack****	LSTM; Dropout	(6,6); 0.1
Classifier	Linear; Linear; Linear	(120,10); (30,10); (10,4)
Student	Conv1; MaxPool; Dropout	K=3, S=1, O=100; K=2, S=2; 0.1
	Conv1; MaxPool; Dropout	K=3, S=1, O=5; K=2, S=2; 0.1
	Conv1; MaxPool; Dropout	K=3, S=1, O=5; K=2, S=2; 0.1
	Linear; Linear	(5,10); (10,4)

*TASmart: Teacher Attention (TA) Branch for Smart Factory. TCSmart: Teacher Causality Extraction (TC) for Smart Factory. TAPack***: TA for OpenPack Dataset. TCPack****: TC for OpenPack

Table 3. Teacher and student profile comparison for Smart Factory and OpenPack dataset

Model (Channels)	FLOPS	Latency	Throughput	Parameters
Teacher Factory (20)	651.85 M	27.91 ms	23.35 GFLOPS	12.65K
Teacher OpenPack (12)	418.08 M	18.68 ms	22.38 GFLOPS	9.82K
Student (3)	22.48 M	3.14 ms	7.15 GFLOPS	2.69K

Using Google Colab with a GPU V100 and DeepSpeed4Science [31] with Step = 5

5 Result and Discussion

Fig. 1**B** shows the teacher results for the smart factory scenario. The model uses 20 inputs of inertial and capacitive channels from both gloves (F1 score of 85.91%) with twelve users and four classes. The Door and Check classes have the higher confusion with 0.9%. These two classes involve grabbing and pulling/pushing objects, in the first, the door is the object and in the second, the objects are inside the factory modules. Fig. 3**A** shows the confusion matrix for the target baseline trained without any knowledge distillation (KD) technique and with the right-handed acceleration input data (3 channels) with 75.94% F1 score. For the baseline, the classes Door and Check are 50% confused. Hence, our teacher contains complementary information compared to the target model.

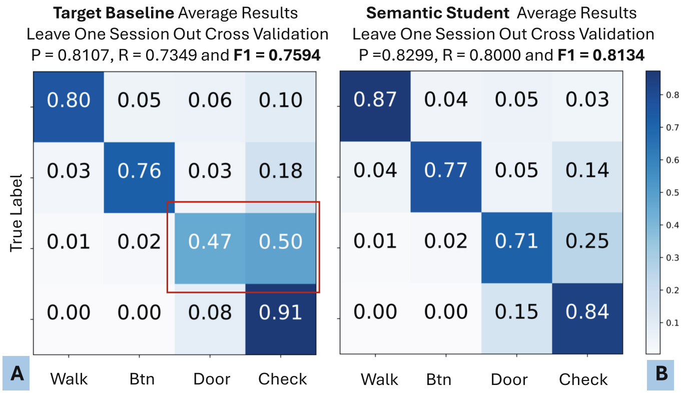


Fig. 3. Twelve users' results in the smart factory scenario with right-handed acceleration channels as input. (A) Baseline; F1 of 75.94% (B) Semantic student; F1 of 81.34%

We have evaluated the performance with different α and temperatures as shown in Fig. 4. In general, the **Combi-Rep** performs best compared to the other hidden vectors. This vector contains causality and multimodal feature extraction information. The **Causal-Rep** is the second best in performance for different alpha values. This vector provides multimodal and multipositional causality knowledge to the student. The **Attn-Rep** (without causality) is the worst case compared to all the KD methods and for the right-hand target model without an increase in the F1 score compared to the baseline. The increase in F1 score compared to the baseline is observed for $\alpha > 0.5$ with the best $\alpha = 0.99$.

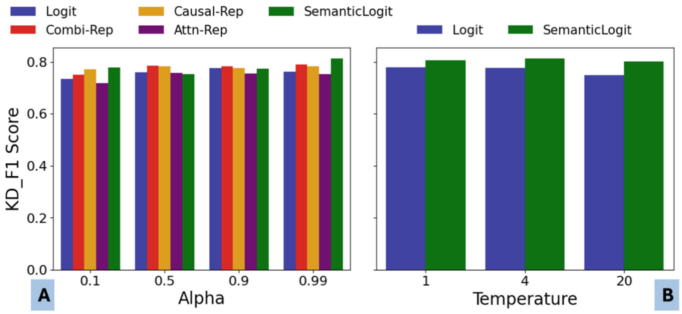


Fig. 4. KD-based student comparison with right-handed acceleration inputs. (A) α variations. (B) Logit and SemanticLogit for different temperatures ($alpha = 0.99$).

Table 4. Results of KD approaches one-handed acceleration data for smart factory scenario

Function	Type	Precision	Recall	F1 Score
Teachers	ATTN	0.7703	0.8013	0.7840
	ATTN+LSTM	0.8574	0.861	0.8591
	Semantic Classifier (SC)	0.9060	0.9057	0.9058
Right Hand Target	Baseline	0.8106	0.7349	**0.7594**
	Logit	0.7834	0.7545	0.7616
	Combi-Rep	0.8209	0.7682	0.7887
	Causal-Rep	0.8061	0.7652	0.7826
	Attn-Rep	0.7835	0.7315	0.7528
	Merged Loss	0.8029	0.7447	0.7676
	SC-Logit	0.8299	0.8000	**0.8134**
	SC-Feature	0.8212	0.7568	0.7817
Left Hand Target	Baseline	0.7512	0.6152	**0.6516**
	Logit	0.7240	0.6510	0.6744
	Combi-Rep	0.7571	0.6770	0.7029
	Causal-Rep	0.7514	0.6633	0.6886
	Attn-Rep	0.7563	0.6674	0.6911
	Merged Loss	0.7574	0.6612	0.6885
	SC-Logit	0.7758	0.7055	**0.7283**
	SC-Feature	0.7673	0.6792	0.7047

SC-Logit (Alpha 0.99 and Temperature 4) with F1-Score of 81.34% and 72.83%

The best result is obtained with the semantic classifier KD method (SC-Logit) with 81.34% F1 score for an $\alpha = 0.99$ and $\tau = 4$ (see Fig. 4**B**). The semantic classifier has an F1 score of 90.58% (+4.67% > teacher). The confusion matrix of the semantic student trained with the SC-Logit KD method is presented in Fig. 3**B**. The Door and Check classes reduce misclassification to 25%, and classes such as Walk and interacting with buttons (Btn) have a 7%, and 1% increase in recall compared to the baseline (see Fig. 3**A**). Table 4 compares the results between the teacher and different KD methods used to train a student with one-handed acceleration data as input. The teacher (CNN-LSTM) with two branches outperformed the teacher with a single-branch CNN, 85.91% compared to 78.40% F1 score. Based on $\alpha = 0.99$, the best case, we have also compared the TSAK KD distillation method with a Merged-Loss (see Table 4). The Merged-Loss is based on the following equation; $\mathcal{L}(x; W) = \alpha \times \mathcal{L}_{CE}(y, z_s) + (0.01) \times \mathcal{L}_{CSKD}(Attn-Rep, h_s) + (0.01) \times \mathcal{L}_{CSKD}(Causal-Rep, h_s) + (0.01) \times \mathcal{L}_{CSKD}(Combi-Rep, h_s)$. Overall, the Merged-Loss outperformed the logit KD and the baseline. Moreover, in Table 4, we included a KD method based on the semantic classifier's last hidden vector (SC-Feature). SC-Feature KD distills knowledge to the one-handed student using the cosine similarity loss ($\alpha = 0.99$). The best result is from the student trained with the semantic classifier (SC-Logit) with an increase of 5.4% and 7.67% F1 score for the right-handed and left-handed targets, respectively.

Fig. 5**A** presents the results for the OpenPack teacher with twelve input channels and 84.43% F1 score. These results are for five participants with a leave-one-session cross-user validation scheme (5 sessions in total). Fig. 5**B** shows the best-distilled student for the OpenPack (SC-Logit KD) with an increase of 10.5% in F1 score compared to the baseline (the target model trained without KD). The semantic classifier has an F1 score of 86.16% (1.73% > teacher).

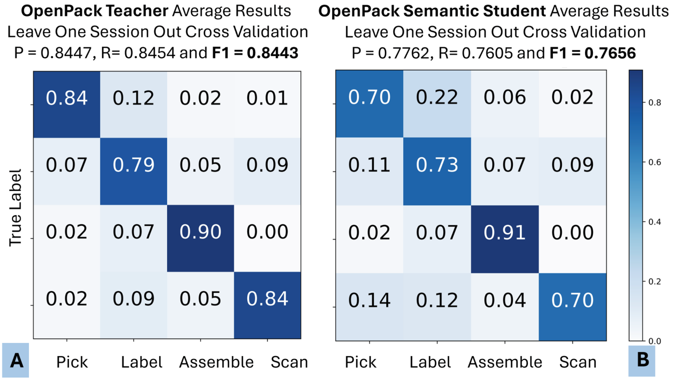


Fig. 5. Five users' results with OpenPack. (A) Teacher results (12 channels); accelerometer and gyroscope (both hands). (B) Student results with right-handed accelerations as input (Semantic KD) and 10.5% F1 score increase.

Table 5 compares the results of the teacher, baseline, and the right-handed student with acceleration channels as input and trained with the five types of KD. The training method for the teacher and the baseline NNs is the same as in the factory dataset case. And, to train the different distilled students the settings are the best cases from the factory dataset. This means these results do not include hyperparameter tuning or any optimization method. We can expect that tuning the NNs for this specific dataset will increase the performance in the future. For both datasets, the second best KD method was the feature-based with the **Combi-Rep**. Overall, our results show the potential of cross-modal knowledge distillation for inertial sensing systems for HAR in a factory testbed.

The majority of distilled student models (excluding **Attn-Rep**) with $\alpha > 0.5$ outperformed the target model baselines. This increase in performance indicates that the multimodal and multipositional teacher effectively guides the student. The semantic classifier (SC-Logit) as the teacher is the most effective KD method for both datasets. Importantly, the semantic classifier is trained in an incremental step from the frozen teacher. In addition, SC-Logit combines feature-based knowledge with logit-based knowledge. This could be the reason for performance improvement. With this method, the lightweight single-position

(3-axis) accelerometer-only model is compressed and improved. This means that our solution retains the practicability of using only three channels IMU available on even the simplest smartwatches, and at the same time, improves performance. For a student model 79% smaller, 8.88 times faster, and 96.6% less computation demanding than the teacher's. On the other hand, our solution has **limitations** and possibilities for improvement:

1. We employ the **Kullback-Leibler Divergence loss** successfully controlling the "soft" targets via the temperature scaling parameter. In [19], the authors proved that the MSE loss outperforms the KLD loss, explained by the difference in the penultimate layer representations between the two losses.
2. We **manually varied the α and T** of the KD. This can be substituted for an automatic parameter search [20].
3. In the future, **quantization-aware KD** will be a strong method to evaluate solutions for wearable HAR. The idea is to coordinate the quantization and KD approach to fine-tune a quantized low-precision student network [18] for an optimized embedded solution.
4. We evaluated our method on **two wearable HAR datasets in manufacturing lines**. This is mainly due to the restricted availability of open datasets with wearable sensors for HAR in industrial settings. However, we believe the method can easily be applied to other scenarios of HAR.

Table 5. Results of KD approaches with right-handed acceleration data for OpenPack

KD Type	Precision	Recall	F1 score
Teacher (CNN-LSTM)	0.8447	0.8454	0.8443
Semantic Classifier	0.8657	0.8596	0.8616
Baseline	0.6799	0.6541	**0.6606**
Logit	0.7119	0.7026	0.7060
Combi-Rep	0.7654	0.7567	0.7571
Causal-Rep	0.7292	0.7329	0.7306
Attn-Rep	0.7493	0.7402	0.7422
Merged-Loss	0.7608	0.7478	0.7531
SC-Logit	0.7762	0.7605	**0.7656**
SC-Feature	0.7627	0.7582	0.7598

6 Conclusion

In this paper, we presented TSAK, prioritizing both model efficiency and precision for activity recognition, a two-stage semantic-aware knowledge distillation approach. In the industrial manufacturing scenario, we tested TSAK with two datasets with smart wearables of wrist and hand-worn sensors: OpenPack and our own recorded dataset at a smart factory testbed. We show that the second stage contributes significantly to the lightweight student model without altering its architecture. The student model takes only 3-axis accelerometer data and has only simple 1D convolution and linear operations with 2.69k parameters, which is qualified to be deployed on most modern microprocessors. The student model enhanced with TSAK has up to 10% better F1 score compared to the same model without KD. With merely 3.4% the computational demand of the first stage teacher model, the student model's F1 score can reach 4.57% short of the teacher model in the best scenario. By leveraging the strengths of multi-modal sensing and machine learning techniques, while prioritizing energy efficiency and model compactness, we have demonstrated significant improvements in recognition performance, computational speed, and energy consumption. As we look to the future, TSAK underpins the development of sustainable, efficient, and accurate wearable HAR systems that can be seamlessly integrated into a variety of applications.

Acknowledgements. The research reported in this paper was partially supported by the German Federal Ministry of Education and Research (BMBF) in the project VidGenSense (01IW21003) and by the European Union in the project SustainML (101070408).

References

1. Aguilar, G., Ling, Y., Zhang, Y., Yao, B., Fan, X., Guo, C.: Knowledge distillation from internal representations. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 7350–7357 (2020)
2. Bello, H., Marin, L.A.S., Suh, S., Zhou, B., Lukowicz, P.: Inmyface: Inertial and mechanomyography-based sensor fusion for wearable facial activity recognition. Information Fusion **99**, 101886 (2023)
3. Bello, H., Suh, S., Geißler, D., Ray, L.S.S., Zhou, B., Lukowicz, P.: Captainglove: Capacitive and inertial fusion-based glove for real-time on edge hand gesture recognition for drone control. In: Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing. pp. 165–169 (2023)
4. Bello, H., Suh, S., Zhou, B., Lukowicz, P.: Besound: Bluetooth-based position estimation enhancing with cross-modality distillation. arXiv preprint arXiv:2404.15999 (2024)
5. Bello, H., Zhou, B., Suh, S., Lukowicz, P.: Mocapaci: Posture and gesture detection in loose garments using textile cables as capacitive antennas. In: Proceedings of the 2021 ACM International Symposium on Wearable Computers. pp. 78–83 (2021)
6. Bello, H., Zhou, B., Suh, S., Sanchez Marin, L.A., Lukowicz, P.: Move with the theremin: Body posture and gesture recognition using the theremin in loose-garment with embedded textile cables as antennas. Frontiers in Computer Science **4**, 915280 (2022)
7. Bock, M., Hölzemann, A., Moeller, M., Van Laerhoven, K.: Improving deep learning for har with shallow lstms. In: Proceedings of the 2021 ACM International Symposium on Wearable Computers. pp. 7–12 (2021)

8. Buciluă, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 535–541 (2006)
9. Cheng, J., Zhou, B., Kunze, K., Rheinländer, C.C., Wille, S., Wehn, N., Weppner, J., Lukowicz, P.: Activity recognition and nutrition monitoring in every day situations with a textile capacitive neckband. In: Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication. pp. 155–158 (2013)
10. Cho, J.H., Hariharan, B.: On the efficacy of knowledge distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4794–4802 (2019)
11. Cicirelli, G., Marani, R., Romeo, L., Domínguez, M.G., Heras, J., Perri, A.G., D'Orazio, T.: The ha4m dataset: Multi-modal monitoring of an assembly task for human action recognition in manufacturing. Scientific Data **9**(1), 745 (2022)
12. Fortes Rey, V., Suh, S., Lukowicz, P.: Learning from the best: contrastive representations learning across sensor locations for wearable activity recognition. In: Proceedings of the 2022 ACM International Symposium on Wearable Computers. pp. 28–32 (2022)
13. Geißler, D., Zahn, E.F., Bello, H., Ray, L.S.S., Woop, E., Zhou, B., Lukowicz, P., Joost, G.: Moca'collection: Normalizing dynamic textile geometry with capacitive sensing in design centric wearables. In: Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing. pp. 276–280 (2023)
14. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. Int. J. Comput. Vision **129**(6), 1789–1819 (2021)
15. Guo, G., Han, L., Wang, L., Zhang, D., Han, J.: Semantic-aware knowledge distillation with parameter-free feature uniformization. Visual Intelligence **1**(1), 6 (2023)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
17. Huang, Y., Zhou, Y., Zhao, H., Riedel, T., Beigl, M.: A survey on wearable human activity recognition: Innovative pipeline development for enhanced research and practice. In: 2024 IEEE International Joint Conference on Neural Networks (IJCNN 2024), Yokohama, 30th June-5th July 2024 (2024)
18. Kim, J., Bhalgat, Y., Lee, J., Patel, C., Kwak, N.: Qkd: Quantization-aware knowledge distillation. arXiv preprint arXiv:1911.12491 (2019)
19. Kim, T., Oh, J., Kim, N., Cho, S., Yun, S.Y.: Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. arXiv preprint arXiv:2105.08919 (2021)
20. Li, L., Dong, P., Wei, Z., Yang, Y.: Automated knowledge distillation via monte carlo tree search. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17413–17424 (2023)
21. Liang, D., Li, G., Adaimi, R., Marculescu, R., Thomaz, E.: Audioimu: Enhancing inertial sensing-based activity recognition with acoustic models. In: Proceedings of the 2022 ACM International Symposium on Wearable Computers. pp. 44–48 (2022)
22. Liu, Y., Wang, K., Li, G., Lin, L.: Semantics-aware adaptive knowledge distillation for sensor-to-vision action recognition. IEEE Trans. Image Process. **30**, 5573–5588 (2021)
23. Liu, Y., Jia, Z., Wang, H.: Emotionkd: A cross-modal knowledge distillation framework for emotion recognition based on physiological signals. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 6122–6131 (2023)

24. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 5191–5198 (2020)
25. Ni, J., Ngu, A.H., Yan, Y.: Progressive cross-modal knowledge distillation for human action recognition. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 5903–5912 (2022)
26. Ni, J., Sarbajna, R., Liu, Y., Ngu, A.H., Yan, Y.: Cross-modal knowledge distillation for vision-to-sensor action recognition. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4448–4452. IEEE (2022)
27. Nikolskiy, V., Stegailov, V.: Floating-point performance of arm cores and their efficiency in classical molecular dynamics. In: Journal of Physics: Conference Series. vol. 681, p. 012049. IOP Publishing (2016)
28. Patidar, P., Goel, M., Agarwal, Y.: Vax: Using existing video and audio-based activity recognition models to bootstrap privacy-sensitive sensors. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **7**(3), 1–24 (2023)
29. Schoonbeek, T.J., Houben, T., Onvlee, H., Van der Sommen, F., et al.: Industreal: A dataset for procedure step recognition handling execution errors in egocentric videos in an industrial-like setting. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4365–4374 (2024)
30. Shephard, D.A.: The 1975 declaration of helsinki and consent. Can. Med. Assoc. J. **115**(12), 1191 (1976)
31. Song, S.L., Kruft, B., Zhang, M., Li, C., Chen, S., Zhang, C., Tanaka, M., Wu, X., Rasley, J., Awan, A.A., et al.: Deepspeed4science initiative: Enabling large-scale scientific discovery through sophisticated ai system technologies. arXiv preprint arXiv:2310.04610 (2023)
32. Wang, L., Yoon, K.J.: Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. IEEE Trans. Pattern Anal. Mach. Intell. **44**(6), 3048–3068 (2021)
33. Yoon, H., Cha, H., Nguyen, C.H., Gong, T., Lee, S.J.: Img2imu: Applying knowledge from large-scale images to imu applications via contrastive learning. arXiv preprint arXiv:2209.00945 (2022)
34. Yoshimura, N., Morales, J., Maekawa, T., Hara, T.: Openpack: A large-scale dataset for recognizing packaging works in iot-enabled logistic environments. In: 2024 IEEE International Conference on Pervasive Computing and Communications (PerCom). pp. 90–97. IEEE (2024)
35. Zheng, H., Lee, R., Lu, Y.: Ha-vid: a human assembly video dataset for comprehensive assembly knowledge understanding. Advances in Neural Information Processing Systems **36** (2024)

ArtNeRF: A Stylized Neural Field for 3D-Aware Artistic Face Synthesis

Zichen Tang[1] and Hongyu Yang[1,2](✉)

[1] School of Artificial Intelligence, Beihang University, Beijing, China
{zctang,hongyuyang}@buaa.edu.cn
[2] Shanghai Artificial Intelligence Laboratory, Shanghai, China

Abstract. Recent advances in generative visual models and neural radiance fields have greatly boosted 3D-aware image synthesis and stylization tasks. However, previous NeRF-based work is limited to single scene stylization, training a model to generate 3D-aware artistic faces with arbitrary styles remains unsolved. We propose ArtNeRF, a novel face stylization framework derived from 3D-aware GAN to tackle this problem. In this framework, we utilize an expressive generator to synthesize stylized faces and a triple-branch discriminator module to improve the visual quality and style consistency of the generated faces. Specifically, a style encoder based on contrastive learning is leveraged to extract robust low-dimensional embeddings of style images, empowering the generator with the knowledge of various styles. To smooth the training process of cross-domain transfer learning, we propose an adaptive style blending module which helps inject style information and allows users to freely tune the level of stylization. We further introduce a neural rendering module to achieve efficient real-time rendering of images with higher resolutions. Extensive experiments demonstrate that ArtNeRF is versatile in generating high-quality 3D-aware artistic faces with arbitrary styles. Code is available at: https://github.com/silence-tang/ArtNeRF.

Keywords: Generative Adversarial Network · Neural Radiance Field · 3D-Aware Image Synthesis · Neural Style Transfer

1 Introduction

With the rise of concepts like Metaverse and Artificial Intelligence Generated Content (AIGC), 3D stylization technology has become increasingly pivotal in various application scenarios such as AR/VR. In this work, we address a novel task of 3D-aware image stylization: given a latent identity code, a style image with arbitrary artistic style, and multiple camera poses, the model should generate 3D-aware stylized faces with high multi-view consistency while preserving the style characteristics of the style image. The challenges of this task are primarily threefold: (1) How to ensure the style consistency between the style image

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 217–232, 2025.
https://doi.org/10.1007/978-3-031-78389-0_15

and the generated image. (2) How to prevent structural information such as the pose of the reference style image from leaking into the generated image. (3) How to guarantee high multi-view consistency and visual quality of the results while achieving efficient real-time rendering.

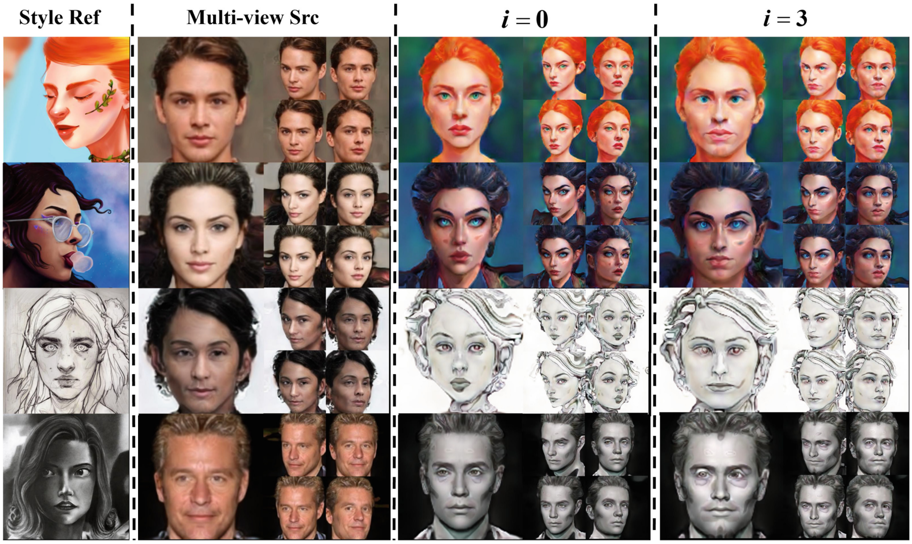


Fig. 1. Multi-view 3D-aware faces with arbitrary styles generated by our model. We evenly select 5 views within a reasonable range where pitch $\in [\frac{\pi}{2} - 0.2, \frac{\pi}{2} + 0.2]$ and yaw $\in [\frac{\pi}{2} - 0.4, \frac{\pi}{2} + 0.4]$. In our SBM module, setting $i = 0$ maximizes stylistic effects, while $i = 3$ strikes a balance between stylization and identity preservation.

Many existing 2D methods [1–3] encode source and reference images into content and attribution latent codes, then combine these codes for reconstruction or style transfer. While these methods support arbitrary reference images, they often yield results with low visual quality and inconsistent style. Additionally, none of these 2D methods can generate 3D-aware results. More recently, [16] proposes a domain-adaption framework and [17] presents a novel stylefield for 3D-aware image stylization. Although these methods achieve high visual quality in 3D-aware results, they are limited to fixed styles and face challenges like high GPU memory consumption during training owing to their 3D representations.

To overcome the limitations of existing methods, we propose **ArtNeRF**, a novel 3D-aware GAN framework for generating multi-view faces with arbitrary styles from given reference images. ArtNeRF features an expressive 3D-aware generator paired with a triple-branch discriminator, achieving rapid and high-quality face stylization. As for the generator, we design better implementation practices to reliably enhance the generation quality of pi-GAN [11]. Specifically, we design dense skip connection layers in the original backbone to strengthen the reuse of feature maps from different semantic layers and discard the progressive growing training strategy. We then design a neural rendering module comprising 1×1 convolutional layers and a skip connection mechanism for image super-resolution and faster rendering. For style representation extraction, we leverage a style encoder based on contrastive learning, effectively mapping reference images to low-dimensional style codes. To address training instability stemming from cross-domain learning discrepancies, we propose an adaptive style blending module that dynamically adjust the blending ratio of style control vectors to ensure a smooth training process. Finally, our triple-branch discriminator module consists of three discriminators with analogous architectures. The first two aid the generator in synthesizing faces adhering to the distributions of the source and target domains, while the third one with an embedder head is capable of improving style consistency between synthesized faces and the reference images. Our contributions can be summarized as follows:

- We propose a novel 3D-aware image arbitrary stylization task, where the synthesized results should emulate the style characteristics of the style reference image while maintaining strong multi-view consistency. Correspondingly, We design ArtNeRF, a framework based on 3D-aware GAN to realize this goal.

- We introduce a self-adaptive style blending module to inject style information into the generator and a triple-branch discriminator to guarantee style consistency. Incorporated with our two-stage training strategy, the cross-domain adaption process can be smoothed and stabilized effectively.

- By designing dense skip connections between sequential layers and incorporating a neural rendering module, we boost the generator backbone of pi-GAN, leading to efficient real-time rendering and better visual quality.

2 Related works

2.1 Style Transfer with 2D GAN

MUNIT [1], FUNIT [2], DRIT++ [3] and StarGANv2 [4] are seminal works that focus on reference-guided image style transfer using GAN [5]. Subsequently, several methods have been proposed to achieve style transfer in specific style domains. CartoonGAN [6] introduces various losses suitable for general photo cartoonization while ChipGAN [7] utilizes an adversarial loss for Chinese ink painting style transfer with constraints on strokes and ink tone. Some recent works combine expressive backbones with unique designs, further boosting the artistic effects of synthesized images. BlendGAN [8] proposes a style encoder and employs a style-conditioned discriminator to generate 2D faces with arbitrary styles. Pastiche Master [9] employs a dual-path style generation network and introduces multi-stage fine-tuning strategies, achieving facial cartoonization with fixed styles. However, none of these methods can generate vivid 3D-aware results.

2.2 3D-aware Image Synthesis

In the realm of 3D-aware image synthesis, we mainly focus on NeRF-based methods. GRAF [10], pi-GAN [11] and GIRAFFE [12] combine GAN with NeRF [13] to learn a 3D representation from 2D images, thereby enabling novel view synthesis. Other endeavors aim to narrow the gap in visual quality between 3D models and 2D GAN models. For instance, GRAM [14] introduces an implicit neural representation based on learnable 2D manifolds, enhancing the quality of synthesized images with reduced sampling points. EG3D [15] presents an effective tri-plane representation for high-quality 3D-aware image synthesis. More recently, some 3d-aware stylization works like 3DAvatarGAN [16] and DeformToon3D [17] are proposed to generate 3D-aware avatars with specific styles. Nevertheless, these methods incur high training costs and are limited in their ability to handle arbitrary styles with a single trained model.

3 Method

Given an identity code $\boldsymbol{z_f}$ sampled from a normal distribution, a reference style image $\boldsymbol{X_s}$ and camera poses $\boldsymbol{p}$, we aim to generate high-quality 3D-aware stylized faces which are supposed to maintain consistent across various views. We firstly give preliminaries in Sec 3.1. To solve the challenges discussed in the introduction, we leverage a style encoder to extract style embeddings of reference images in Sec 3.2, a novel generative radiance field to achieve efficient style blending and rendering in Sec 3.3, and a triple-branch discriminator network to supervise the 3D-aware generator and enhance style consistency in Sec 3.4.

3.1 Preliminaries

Neural Radiance Fields. A Neural Radiance Field (NeRF) implicitly represents the scene as a 5D function, enabling high-quality synthesis of novel views with multi-view consistency. Given a 3D point x, a radiance field g_θ is employed to map its position (x, y, z) and the viewing direction (θ, ϕ) to its RGB color $\boldsymbol{c}$ and volume density σ. To render a pixel, a ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ is cast from the camera origin $\boldsymbol{o}$ to the 3D space along the viewing direction $\boldsymbol{d}$, where $t \in [t_n, t_f]$ represents the distance from the sampling point to the camera origin. The color of the pixel can be rendered via volume rendering:

$$\boldsymbol{C}(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})dt,\ T(t) = e^{-\int_{t_n}^{t} \sigma(r(s))ds} \tag{1}$$

where $T(t)$ is the cumulative transmittance from t_n to t.

3.2 Self-supervised Style Encoder

Style encoder used to extract features is indispensable in style transfer tasks. However, utilizing an VGG-based encoder with randomly initialized parameters

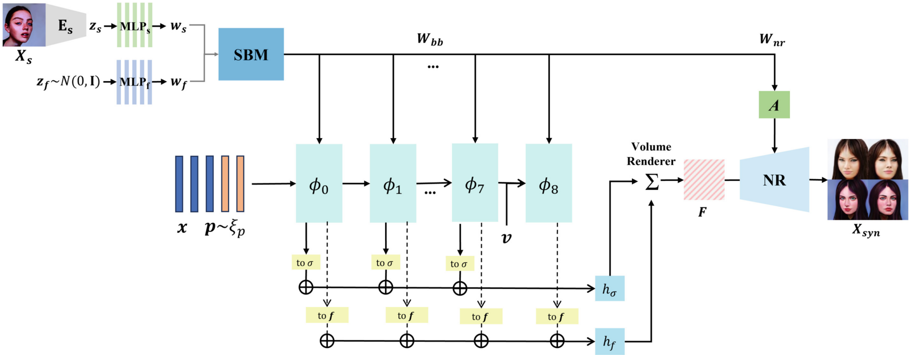


Fig. 2. The pipeline of the generator in ArtNeRF. Given an identity code $\boldsymbol{z_f}$ sampled from normal distribution and a style image $\boldsymbol{X_s}$, we first extract the style code using the style encoder E_s. Subsequently, dual mapping networks are utilized to map $\boldsymbol{z_f}, \boldsymbol{z_s}$ to $\boldsymbol{W_f}, \boldsymbol{W_s}$ in the W^+ space. The self-adaptive SBM module then blends $\boldsymbol{W_f}, \boldsymbol{W_s}$ based on a split index i and injects the style information into the 3D generator. Given camera poses $\boldsymbol{p}$, the sampled 3D points $\boldsymbol{x}$ in the camera coordinate system are first transformed to the world coordinate system and then modulated by a sequence of FiLM blocks incorporating style information. Dense skip connections are applied to accumulate the intermediate features output by these blocks, contributing to the volume density and features of the 3D points. Finally, low-resolution feature maps are synthesized through volume rendering and then up-sampled and refined by the neural rendering module (NR) to achieve real-time rendering of 3D-aware stylized faces.

may lead to inconsistent styles and confusion. Moreover, since we need to generate 3D-aware images, it is crucial to prevent the leakage of pose information from the reference images into the style latent codes. To tackle this problem, we leverage a style encoder with strong expressive capability following [8]. Note that the parameter of the style encoder is frozen after the training process is finished.

The overall structure of the style encoder is illustrated in Fig. 3. A pretrained VGG19 is utilized to extract style features $\boldsymbol{f_s}$ from input images, followed by a compress-CNN to reduce the dimension of $\boldsymbol{f_s}$ to 512. The 512-dim vectors serve as the style codes $\boldsymbol{z_s}$. To facilitate the contrastive learning process, a projection head is applied to further map $\boldsymbol{z_s}$ to their representation vectors $\boldsymbol{u_s}$. During training, each batch contains $2N$ images, where $\boldsymbol{X_i}, \boldsymbol{X_j}$ are positive samples ($\boldsymbol{X_i}$ is a style image and $\boldsymbol{X_j}$ is an augmented sample via affine transformation), and the remaining $2N-2$ images serve as negative samples. We use the following objective function to optimize the compress-CNN :

$$\ell_{i,j}^{CL} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{u_i}, \boldsymbol{u_j})/\tau)}{\sum\limits_{k=1, k\neq i, k\neq j}^{2N} \exp(\mathrm{sim}(\boldsymbol{u_i}, \boldsymbol{u_k})/\tau)} \tag{2}$$

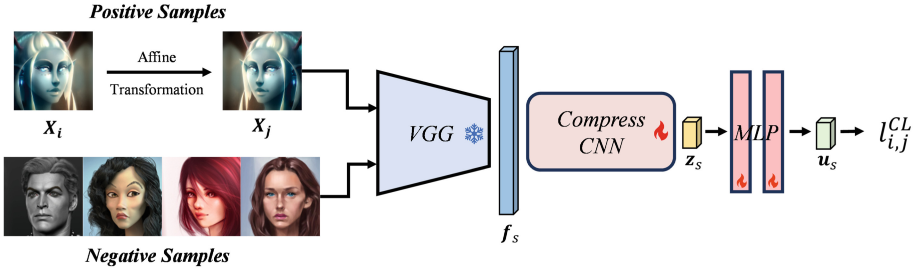


Fig. 3. The architecture of the style encoder E_s.

where sim(·, ·) is the cosine similarity between embeddings, τ is the temperature coefficient, and $(\boldsymbol{u_i}, \boldsymbol{u_j})$ represents the contrastive learning representations for $(\boldsymbol{X_i}, \boldsymbol{X_j})$. After training, augmented samples of the same style image will have style codes rich in style semantics but devoid of spatial structure, since they are pushed closer in the embedding space and their original spatial features are neutralized.

3.3 Conditional Generative Radiance Field

pi-GAN [11] is a NeRF-based 3D-aware face generation framework. We start from pi-GAN to design our generator considering its concise and effective backbone along with its relatively small training overhead. In this work, we extend and enhance pi-GAN to address the 3D-aware image stylization problem. Specifically, our improved neural generative radiance field comprises three main components: mapping network, style blending module (SBM), and conditional radiance field with dense skip connections.

Mapping network and SBM Module. Let $\boldsymbol{z_f}$ denote the identity latent code, and $\boldsymbol{z_s}$ represent the style code obtained from the style image. We first utilize two mapping networks with unshared parameters to respectively map $\boldsymbol{z}$ and $\boldsymbol{z_s}$ from z space to $\boldsymbol{w}$ and $\boldsymbol{w_s}$ in W^+ space to achieve feature decoupling.

To incorporate the two latent codes into our backbone, we design a style blending module SBM. Since different layers are responsible for learning facial semantic information at various levels, simply setting fixed blending weights for each layer is not advisable, which will inevitably cause mode collapse. Inspired by this, we first mix $\boldsymbol{w_f}$ and $\boldsymbol{w_s}$ using a learnable weight vector $\boldsymbol{\alpha}$, then feed the mixed code into our model to achieve style mixing of facial semantic information at multiple levels. The SBM module can be formulated as Eq.3. We omit the batch dimension for simplicity.

$$\begin{aligned} \boldsymbol{W_{fused}} &= \text{Concat}(\boldsymbol{W_{bb}}; \boldsymbol{W_{nr}}) \\ \boldsymbol{W_{bb}} &= \boldsymbol{\alpha}[: k] \odot \boldsymbol{w_s}[: k, :] + (1 - \boldsymbol{\alpha}[: k]) \odot \boldsymbol{w_f}[: k, :] \\ \boldsymbol{W_{nr}} &= \boldsymbol{\alpha}[k :] \odot \text{Trans}(\boldsymbol{w_s}[k :, :]) + (1 - \boldsymbol{\alpha}[k :]) \odot \text{Trans}(\boldsymbol{w_f}[k :, :]) \end{aligned} \quad (3)$$

where bb and nr denote the backbone (conditional generative radiance field introduced in Sec. 3.3) and the neural rendering module (which will be introduced in Sec. 3.4), $\mathrm{Concat}(\cdot;\cdot)$ and $\odot$ denote channel-wise concatenation and element-wise multiplication, [:] represent the slicing operation in PyTorch. In practice, k is the number of layers in the backbone, $\boldsymbol{\alpha}$ is a learnable weight vector with a shape of $[n]$, $\boldsymbol{w_f}$ and $\boldsymbol{w_s}$ both have a shape of $[n, 256]$, where n is the number of layers requiring style mixing. To flexibly adjust the degree of stylization, we introduce a split index i in SBM. When i is specified, $\boldsymbol{\alpha}[: i] = 1$ and $\boldsymbol{\alpha}[i :]$ remains unchanged. This strategy ensures layers with indices less than i only affected by $\boldsymbol{w_f}$, while layers with indices greater than i influenced by both $\boldsymbol{w_f}$ and $\boldsymbol{w_s}$. Consequently, we can perform style blending on the backbone and the neural rendering module (Sec 3.4). Note that directly injecting style vector into the neural rendering module may be improper as its feature space differs from the backbone. Hence, we apply projection operation to $\boldsymbol{w_s}$ and $\boldsymbol{w_f}$ to refine them before the injection operation, denoted as Trans.

Conditional Radiance Field with Dense Skip Connections. The proposed conditional radiance field takes as input not only 3D positions $\boldsymbol{x}$ in the camera coordinate system and a camera pose $\boldsymbol{p}$ but also a fused conditioning latent code $\boldsymbol{W_{bb}}$. Therefore, properly injecting $\boldsymbol{W_{bb}}$ into the backbone is crucial for the generation performance. As is shown in Fig.2, the backbone consists of two parts: a sequence of n FiLM layers and dense skip connections. The FiLM layer sequence can be formulated as follows:

$$\begin{aligned}\Phi(\boldsymbol{x}) &= \phi_{n-1} \circ \phi_{n-2} \circ ... \circ \phi_0(\boldsymbol{x}) \\ \phi_i(\boldsymbol{x_i}) &= \sin(\boldsymbol{\gamma_i} \cdot (\boldsymbol{W_i x_i} + \boldsymbol{b_i}) + \boldsymbol{\beta_i})\end{aligned} \tag{4}$$

where $\boldsymbol{x_i}$ is the input of the i-th FiLM layer, $\boldsymbol{W_i}, \boldsymbol{b_i}$ are learnable parameters and γ_i, β_i are modulation coefficients projected from $\boldsymbol{W_{bb}}$. Note that ϕ_{n-1} also takes the viewing direction $\boldsymbol{v}$ as input to model view-dependant appearance, we omit it for brevity.

To mitigate the ripple-like artifacts observed during training pi-GAN, we draw inspiration from StyleGAN2's [18] improvements to StyleGAN. We discard the progressive growing training strategy used in pi-GAN and optimize the generator with a structure featuring dense skip connections. This modification ensures that feature maps from different layers can mutually contribute to the final output, thereby increasing the strength of gradient back-propagation and preventing training collapse. Optimized formulas for volume density and feature calculation is shown in Eq.5:

$$\sigma(\boldsymbol{x}) = h_\sigma(\sum_{i=0}^{n-2} \lambda_i(\boldsymbol{M_i})), \boldsymbol{f}(\boldsymbol{x}) = h_f(\sum_{i=0}^{n-1} \mu_i(\boldsymbol{M_i})) \tag{5}$$

where $\boldsymbol{M_i}$ represents the output of ϕ_i, λ_i is the i-th volume density prediction layer (to σ block in Fig. 2) and μ_i is the i-th feature prediction layer inspired by [12] (to f block in Fig. 2). h_σ, h_f are used to clamp the volume density $\sigma_i \in \mathbb{R}^1$ and the feature values $\boldsymbol{f_i} \in \mathbb{R}^{M_f}$. Let $\{\boldsymbol{x_i}\}_{i=1}^{N_s}$ denote the N_s sampling points along a ray, with volume density σ_i and feature values $\boldsymbol{f_i}$ of each point, the volume rendering can be defined as follows:

$$\pi_{vol} : (\mathbb{R} \times \mathbb{R}^{M_f})^{N_s} \mapsto \mathbb{R}^{M_f}, \quad \{\sigma_i, \boldsymbol{f_i}\}_{i=1}^{N_s} \mapsto \boldsymbol{f} \tag{6}$$

By performing volume rendering to all rays, we can rapidly obtain the complete feature map $\boldsymbol{F} \in \mathbb{R}^{M_f \times 32 \times 32}$ with relatively small GPU overhead. We can intuitively consider $\boldsymbol{F}$ as a texture representation of the final image.

3.4 Neural Rendering Module

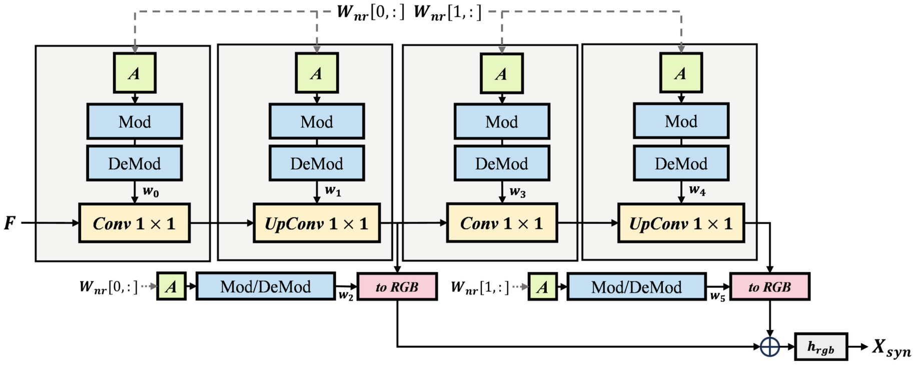


Fig. 4. The architecture of the neural rendering module.

The integration of the neural rendering module can remarkably enhance the expressiveness of the generator, enabling the synthesis of high-quality images at higher resolutions with faster inference speed. GIRAFFE [12] first introduces a neural renderer composed of 2D CNNs into the model. However, EG3D [15] indicates that overly deep 2D CNNs and excessive 3 × 3 convolution operations undermine the 3D consistency of the final results. Therefore, our proposed neural rendering module is composed of shallow 1 × 1 convolutions. 1 × 1 convolutions enhance the network's ability to model information between channels of 2D feature maps and help avoid the fusion of local spatial contexture in feature maps, further ensuring multi-view consistency. The proposed neural rendering module leverages ModConv and upsampling layers in StyleGAN2, with a to_rgb layer to facilitate the reuse of features between adjacent blocks. Introducing ModConv allows for the reuse of $\boldsymbol{W_{nr}}$ during super-resolution, which can refine the details of low-resolution results. Given a 2D low-resolution feature map $\boldsymbol{F} \in \mathbb{R}^{M_f \times 32 \times 32}$, we can generate the final synthesized image $\boldsymbol{X_{syn}} \in \mathbb{R}^{3 \times 128 \times 128}$ following Fig. 4, where the A blocks are affine transformations applied to inject $\boldsymbol{W_{nr}}$ into the neural rendering module, $\mathbf{w_0}$ to $\mathbf{w_5}$ are weight parameters for 1×1 convolutions and the h_{rgb} is a clamp operation.

3.5 Triple Discriminator Network

We employ three discriminators to guide the generator to synthesize 3D-aware stylized images decently and appropriately. D_r discriminates between fake natural faces and real natural faces, while D_s discriminates between fake stylized faces and real ones. Together, they supervise the generator to ensure the generated images conforming to the distributions of the respective domains. In order to further ensuring that the synthesized images match the style of the given reference images, we treat $\boldsymbol{w_s}$ as a sort of class labels inspired by [19]. The task of generating stylized images can be naturally transformed into a cGAN problem. Therefore, we leverage a conditional discriminator D_c, which provides an additional supervision to the generator. Specifically, we apply an embedder head at the end of D_c. Let's denote the output of the global sum pooling layer in D_c as $\boldsymbol{f_{gsp}}$. We first map a given style code to a feature embedding $\boldsymbol{f_{emb}}$ aligned with $\boldsymbol{f_{gsp}}$, then the dot product result of $\boldsymbol{f_{gsp}}$ and $\boldsymbol{f_{emb}}$ is added to the original output of D_c to form the final output. The structure of D_r and D_s is similar to D_c, but without the embedder head.

3.6 Loss Functions

Given a reference style image $\boldsymbol{X_s}$, we extract its style code $\boldsymbol{z_s}$ with E_s. We then sample an identity code $\boldsymbol{z_f}$ from a normal distribution and a camera pose $\boldsymbol{\xi}$ from a predefined distribution. On one side, we aim to synthesize fully stylized faces with the split index $i = 0$ in SBM. On the other side, to ensure that generated stylized faces retain the original face identity during cross-domain adaption process, the generator should generate fully natural faces with $i = 11$ in SBM. Additionally, a style consistency loss is leveraged to guarantee that the stylized faces share the same style as the reference images. During training, we maintain an embedding queue Q that stores style codes from $i-1$-th batch. When we process i-th batch, we first sample a code $\boldsymbol{z_s^-}$ from Q as a negative sample, we then instruct the generator to synthesize stylized face $\boldsymbol{X_s^-} = G_{i=0}(\boldsymbol{z_f}, \boldsymbol{z_s^-})$. We feed $(\boldsymbol{X_s^-}, \boldsymbol{z_s^-})$ into D_c as a pair of negative sample. The objective functions for D_s, D_r, D_c can be formulated as follows:

$$\begin{aligned}
\mathcal{L}_s &= \mathbb{E}_{\boldsymbol{z_f}, \boldsymbol{X}_s, \boldsymbol{\xi}} \left[f(D_s(G_{i=0}(\boldsymbol{z_f}, \boldsymbol{z_s}, \boldsymbol{\xi}))) \right] + \mathbb{E}_{\boldsymbol{X}_s} \left[f(-D_s(\boldsymbol{X_s})) + \lambda \|\nabla D_s(\boldsymbol{X_s})\|^2 \right] \\
\mathcal{L}_r &= \mathbb{E}_{\boldsymbol{z_f}, \boldsymbol{\xi}} \left[f(D_r(G_{i=11}(\boldsymbol{z_f}, \boldsymbol{\xi}))) \right] + \mathbb{E}_{\boldsymbol{X}_r} \left[f(-D_r(\boldsymbol{X_r})) + \lambda \|\nabla D_r(\boldsymbol{X_r})\|^2 \right] \\
\mathcal{L}_c &= \mathbb{E}_{\boldsymbol{z_f}, \boldsymbol{X}_s} \left[f(D_c(\boldsymbol{X_s^-}, \boldsymbol{z_s^-}) \right] + \mathbb{E}_{\boldsymbol{X}_s} \left[f(-D_c(\boldsymbol{X_s}, E_s(\boldsymbol{X_s}))) \right]
\end{aligned} \tag{7}$$

where $f(x) = -\log(1 + e^{-x})$. We adopt the non-saturating GAN objective [12] and R_1 gradient penalty to avoid mode collapse as well as stabilize the entire training process.

Finally, we need to ensure that all the generated faces are constrained within the same canonical space. To this end, the discriminator should predict the camera pose $\hat{\boldsymbol{\xi}} = (pitch, yaw)$ of the generated face and compute a pose consistency loss between $\hat{\boldsymbol{\xi}}$ and the previously sampled pose $\boldsymbol{\xi}$. We apply pose consistency loss for both natural faces (denoted as real) and stylized faces (denoted as style):

$$\begin{aligned}\mathcal{L}_{real-pose} &= \mathbb{E}_{\xi}\left\|\hat{\boldsymbol{\xi}}_{real} - \boldsymbol{\xi}_{real}\right\|^2 \\ \mathcal{L}_{style-pose} &= \mathbb{E}_{\xi}\left\|\hat{\boldsymbol{\xi}}_{style} - \boldsymbol{\xi}_{style}\right\|^2\end{aligned} \tag{8}$$

Given $\lambda_1, \lambda_2, \lambda_3$, which are weights to balance these objective functions, the entire training loss of ArtNeRF is:

$$\begin{aligned}&\mathcal{L}_D = \lambda_1(\mathcal{L}_{real} + \mathcal{L}_{real-pose}) + \lambda_2(\mathcal{L}_{style} + \mathcal{L}_{style-pose}) + \lambda_3\mathcal{L}_{style-latent} \\ &\mathcal{L}_G = -{\mathcal{L}_D}^{no-R1}\end{aligned} \tag{9}$$

where ${\mathcal{L}_D}^{no-R1}$ represents $\mathcal{L}_D$ without R_1 penalty term.

4 Experiments

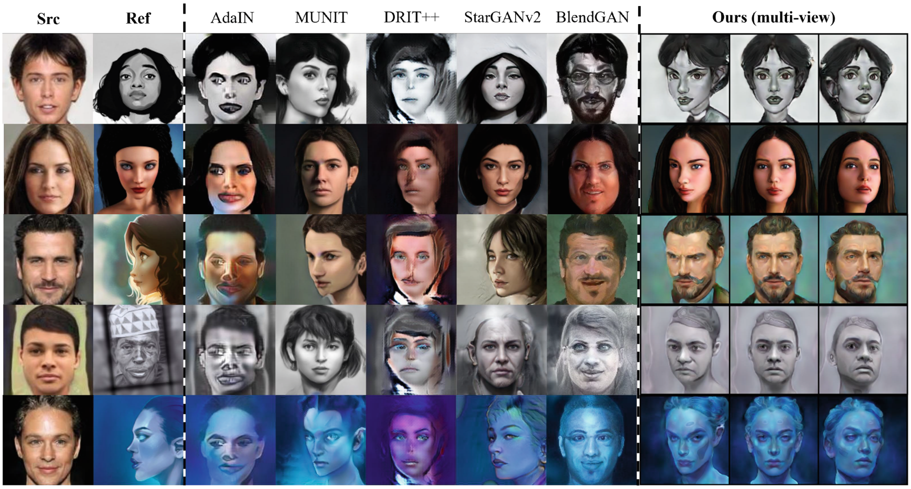


Fig. 5. Qualitative comparison of reference-guided face stylization results among several 2D methods and ours. Our model can not only generate high-quality stylized faces but also produce 3D-aware results with high multi-view consistency.

Datasets. We utilize CelebA [20], containing approximately 200k faces, as our source domain dataset. For the style domain, we employ AAHQ [8], an artistic dataset comprising around 24k high-quality stylized faces. All the images from the two datasets have been cropped and aligned properly, with a resolution of 128 × 128.

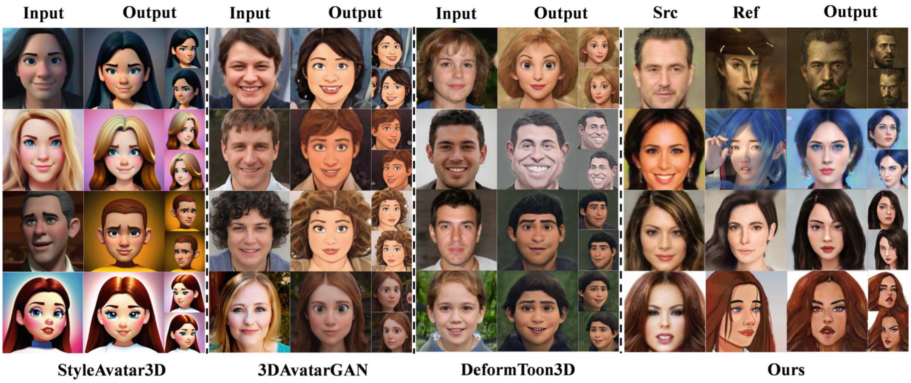


Fig. 6. Qualitative comparison of face stylization results among several 3D-aware methods and ours. StyleAvatar3D can only generate cartoonized faces from input cartoon images. 3DAvatarGAN creates faces with a fixed style after performing domain adaptation on a given dataset. DeformToon3D can synthesize faces with several fixed styles. Our method enables the direct generation of 3D-aware faces with diverse artistic styles given reference images, all without the need for fine-tuning on a specific style dataset.

Implementation details. We use a two-stage strategy to train our model: base model pre-training and fine-tuning within the style domain. We train 200k steps for stage1 with the generator and D_r using CelebA and 30k steps for stage2 with the generator, triple-branch discriminator, style encoder and SBM module using both the two datasets. After stage1, our model can sufficiently learn prior knowledge about the distribution of real natural faces and generate high-quality multi-view natural faces. In stage 2, the pretrained model we get after stage 1 will be decently guided to generate stylized faces in a cross-domain adaption manner. Our method is implemented in PyTorch. The entire training process is done on a NVIDIA RTX 2080Ti for about 3 days.

4.1 Comparisons

Qualitative results. Fig. 1 displays synthesized multi-view natural faces and their corresponding stylized faces at different levels ($i = 0$ and $i = 3$) within the SBM module. At $i = 0$, the results exhibit the highest degree of stylization but lose some identity features. At $i = 3$, a balance between stylization and identity preservation is achieved. More results can be found in our supplementary material. Fig. 5 provides a qualitative comparison with 2D methods which support one-shot face stylization with arbitrary reference style images (we set $i = 3$ in our SBM module). DRIT++ fails to learn the style consistency between generated faces and reference images, while AdaIN struggles with identity preservation. Although MUNIT and StarGANv2 produce reasonable results, they tend to overly inherit face poses from reference images. BlendGAN performs well in identity preservation and style consistency but lacks 3D-awareness. Our

Table 1. A thorough comparison of the functionality among several prevailing 2D or 3D face cartoonization or stylization methods.

Method	Year	Reference-guided	Arbitrary style	3D-aware
CartoonGAN [6]	CVPR18	×	×	×
AniGAN [22]	TMM21	✓	×	×
AdaIN [21]	ICCV17	✓	✓	×
MUNIT[1]	ECCV18	✓	✓	×
FUNIT[2]	ICCV19	✓	✓	×
DRIT++[3]	IJCV20	✓	✓	×
StarGANv2 [4]	CVPR20	✓	✓	×
BlendGAN [8]	NIPS21	✓	✓	×
JoJoGAN [23]	ECCV22	✓	×	×
DualStyleGAN [9]	CVPR22	✓	×	×
StyleAvatar3D [25]	ARXIV23	×	×	✓
3DAvatarGAN [16]	CVPR23	✓	×	✓
DeformToon3D [17]	ICCV23	✓	×	✓
ArtNeRF(Ours)	**2024**	✓	✓	✓

method excels in synthesizing 3D-aware images with robust multi-view consistency, achieving strong identity preservation and style consistency simultaneously. Fig. 6 compares several 3D-aware face stylization methods with ours. Existing methods only generate faces with one or several fixed styles like cartoon or caricature, lacking the capability of generalizing to more diverse artistic styles. They typically require domain adaptation for each style or the collection of paired training data, which limits their applicability. In contrast, our method supports reference-guided face stylization and not being restricted to a set of fixed styles. Additionally, thanks to our two-stage training strategy and SBM module, the results generated by our method naturally inherits the facial identity and hairstyle traits from the source faces, which is not achieved by other methods.

Quantitative results. Table 1 demonstrates our method's capability to generate reference-guided 3D-aware faces with arbitrary styles, a task not addressed by existing methods. Note that our method is the first to achieve 3D-aware face stylization with arbitrary artistic styles, hence our quantitative comparisons are drawn with existing 2D methods which can generate results in a single forward pass. We provide quantitative comparisons against reference-guided image synthesis baselines using FID, KID, and IS metrics on 20k generated stylized images and 20k style reference images. Besides, we assess image diversity using LPIPS. Given a specified identity code, we select 10 reference styles randomly and generate 10 stylized faces, we then evaluate the LPIPS scores between every 2 results. This process is repeated for 1000 identity codes and the average of all scores constitute the final LPIPS score. BlendGAN differs from ArtNeRF in training settings and their training code is unavailable, so we reproduce the latent-guided (an identity code and a style code is sampled) results of BlendGAN and downsample them to 128 × 128 for comparison.

Table 2. Quantitative evaluation of style-guided face synthesis. We compare with methods that support face stylization with arbitrary reference styles.

Method	FID ↓	KID↓	IS↑	LPIPS↑
AdaIN	86.87	0.084	2.14	0.237
MUNIT	56.99	0.046	1.98	0.241
DRIT++	89.79	0.069	2.02	0.231
StarGANv2	34.24	0.022	2.50	0.389
BlendGAN	39.45	0.037	**2.98**	0.239
Ours(i=3)	13.80	0.0066	2.89	0.377
Ours(i=0)	**12.09**	**0.0052**	2.96	**0.403**

Table 2 highlights our method's significant improvements over AdaIN, MUNIT, DRIT++, and StarGANv2 across quantitative metrics. Notably, we use multi-view stylized faces for evaluation while other methods use fix-pose faces. It manifests the 3D-aware faces generated by our method possess higher visual quality and diversity than the baseline methods. BlendGAN is the SOTA in 2D reference-guided image synthesis with arbitrary style and our method exhibits slightly inferior IS compared to BlendGAN, suggesting that there is still room for our method to narrow the visual quality gap between 2D and 3D-aware methods.

4.2 Ablation Study

In this section, we conduct extensive ablation studies to assess the impact of various modules on the model's generative capability and demonstrate their effectiveness. Experiments involving the base model (stage 1) are conducted using the CelebA dataset, whereas experiments focusing on the final stylization model (stage 2) are carried out using the AAHQ dataset. Due to space limitation, please refer to our supplementary material for more results.

Generator network. In Sec 3.3, we enhance the pi-GAN baseline by omitting the progressive growing strategy, integrating dense skip connections into the backbone, and introducing a neural rendering module. We assess their effectiveness by progressively incorporating them into the baseline model in stage 1 and comparing results in Table 3. Omitting the progressive growing strategy initially yields a slightly higher FID at the start of training but significantly lowers it towards the end. Adding dense skip connections further reduces the final FID. Finally, if the neural rendering module is applied, the visual quality of our results will be further refined during the entire training process.

Table 3. We improve the generation capability of the base model in a progressive way. PG, DSC and NR denote progressive growing training strategy, dense skip connections and neural rendering, respectively.

	FID↓ (10k)	FID↓ (20k)	FID↓ (40k)	FID↓ (100k)
base model	58.76	40.83	24.47	36.49
-PG	63.9	49.05	34.95	27.46
-PG, +DSC	61.59	40.39	24.61	17.28
-PG, +DSC, +NR (ours)	**53.97**	**33.97**	**22.29**	**14.42**

Table 4. The rendering speed (fps) with (w/) and without (w/o) neural rendering module under different (res, ns) pairs. OOM denotes CUDA out of memory error.

	ns=8		ns=16		ns=32		ns=48	
	w/o nr	w/ nr	w/o nr	w/ nr	w/o nr	w/ nr	w/o nr	w/ nr
res=64	33.45	**49.69**	41.20	**56.44**	26.33	**48.80**	18.26	**48.76**
res=128	25.57	**53.60**	14.11	**52.40**	7.29	**42.69**	OOM	**38.54**
res=256	7.22	**43.40**	3.25	**32.93**	OOM	**21.93**	OOM	**15.58**

Neural rendering. We analyze how neural rendering impacts inference speed across different resolutions (res) and samples per ray (ns). Experiments are conducted with our stylization model. As detailed in Table 4, a significant enhancement in inference speed across nearly all $(\mathrm{res}, \mathrm{ns})$ pairs can be achieved with neural rendering. Notably, when $\mathrm{res} = 256, \mathrm{ns} = 16$, neural rendering improves inference speed by 10× compared to the original structure. This capability allows the model to efficiently handle diverse $(\mathrm{res}, \mathrm{ns})$ settings, facilitating high-quality real-time rendering essential for VR/AR applications.

5 Conclusion

In this paper, we propose a novel 3D-aware image stylization method ArtNeRF, enabling the generation of faces with arbitrary styles. We achieve this goal by enhancing a NeRF-GAN baseline with dense skip connections and a neural rendering module, proposing an SBM module to integrate style control vectors into the generator and leveraging a triple-branch discriminator to improve style and multi-view consistency. Extensive experiments illustrate the effectiveness of ArtNeRF. However, our model still has limitations. Although reasonable faces can be generated with most camera viewpoints, our model cannot tackle with extreme views or synthesize 360° images of human heads due to dataset constraints. Future work will incorporate advanced 3D representations like 3D Gaussian Splatting [24] to further enhance image quality and rendering speed.

Acknowledgements. This work is partly supported by the National Key R&D Program of China (No. 2022ZD0161902), the National Natural Science Foundation of

China (No. 62202031), the Beijing Natural Science Foundation (No. 4222049), and the Fundamental Research Funds for the Central Universities.

References

1. Huang, X., Liu, M. Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In Proceedings of the European conference on computer vision (ECCV), pp. 172-189 (2018)
2. Liu, M. Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., Kautz, J.: Few-shot unsupervised image-to-image translation. In Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp. 10551-10560 (2019)
3. Lee, H.Y., et al.: DRIT++: Diverse Image-to-Image Translation via Disentangled Representations. Int. J. Comput. Vis. **128**, 2402–2417 (2020)
4. Choi, Y., Uh, Y., Yoo, J., Ha, J. W.: Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp. 8188-8197 (2020)
5. Goodfellow, I., et al.: Generative adversarial nets. Advances in neural information processing systems, 27 (2014)
6. Chen, Y., Lai, Y. K., Liu, Y. J.: Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), pp. 9465-9474 (2018)
7. He, B., Gao, F., Ma, D., Shi, B., Duan, L. Y.: Chipgan: A generative adversarial network for chinese ink wash painting style transfer. In Proceedings of the 26th ACM international conference on Multimedia, pp. 1172-1180 (2018)
8. Liu, M., Li, Q., Qin, Z., Zhang, G., Wan, P., Zheng, W.: Blendgan: Implicitly gan blending for arbitrary stylized face generation. Adv. Neural. Inf. Process. Syst. **34**, 29710–29722 (2021)
9. Yang, S., Jiang, L., Liu, Z., Loy, C. C.: Pastiche master: Exemplar-based high-resolution portrait style transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7693-7702 (2022)
10. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. Adv. Neural. Inf. Process. Syst. **33**, 20154–20166 (2020)
11. Chan, E. R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp. 5799-5809 (2021)
12. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11453-11464 (2021)
13. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (2021)
14. Deng, Y., Yang, J., Xiang, J., Tong, X.: Gram: Generative radiance manifolds for 3d-aware image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10673-10683 (2022)
15. Chan, E. R., Lin, C. Z., Chan, M. A., et al.: Efficient geometry-aware 3d generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp. 16123-16133 (2022)

16. Abdal, R., Lee, H. Y., Zhu, P., et al.: 3davatargan: Bridging domains for personalized editable avatars. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4552-4562 (2023)
17. Zhang, J., Lan, Y., Yang, S., et al.: Deformtoon3d: Deformable neural radiance fields for 3d toonification. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9144-9154 (2023)
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 (CVPR), pp. 8110-8119 (2020)
19. Miyato, T., Koyama, M.: cGANs with projection discriminator. arXiv preprint arXiv:1802.05637 (2018)
20. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 3730-3738 (2015)
21. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1501-1510 (2017)
22. Li, B., Zhu, Y., Wang, Y., Lin, C.W., Ghanem, B., Shen, L.: Anigan: Style-guided generative adversarial networks for unsupervised anime face generation. IEEE Trans. Multimedia **24**, 4077–4091 (2021)
23. Chong, M. J., Forsyth, D.: Jojogan: One shot face stylization. In European Conference on Computer Vision (ECCV), pp. 128-152 (2022)
24. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
25. Zhang, C., Chen, Y., Fu, Y., et al.: Styleavatar3d: Leveraging image-text diffusion models for high-fidelity 3d avatar generation. arXiv preprint arXiv:2305.19012 (2023)

Latent Behavior Diffusion for Sequential Reaction Generation in Dyadic Setting

Minh-Duc Nguyen, Hyung-Jeong Yang(✉), Soo-Hyung Kim, Ji-Eun Shin,
and Seung-Won Kim

Chonnam National University, Gwangju, South Korea
hjyang@jnu.ac.kr

Abstract. The dyadic reaction generation task involves synthesizing responsive facial reactions that align closely with the behaviors of a conversational partner, enhancing the naturalness and effectiveness of humanlike interaction simulations. This paper introduces a novel approach, the Latent Behavior Diffusion Model, comprising a context-aware autoencoder and a diffusion-based conditional generator that addresses the challenge of generating diverse and contextually relevant facial reactions from input speaker behaviors. The autoencoder compresses high-dimensional input features, capturing dynamic patterns in listener reactions while condensing complex input data into a concise latent representation, facilitating more expressive and contextually appropriate reaction synthesis. The diffusion-based conditional generator operates on the latent space generated by the autoencoder to predict realistic facial reactions in a non-autoregressive manner. This approach allows for generating diverse facial reactions that reflect subtle variations in conversational cues and emotional states. Experimental results demonstrate the effectiveness of our approach in achieving superior performance in dyadic reaction synthesis tasks compared to existing methods.

Keywords: Latent Diffusion · AutoEncoder · Dyadic interaction · Multiple appropriate reactions generation

1 Introduction

Dyadic interaction refers to the communication or relationship between two individuals, characterized by direct and reciprocal exchange. This form of interaction is fundamental in social and psychological studies, as it helps to understand interpersonal dynamics, mutual influence, and the development of social bonds. Referring to the Stimulus Organism Response (SOR) model [1], each human individual expresses reaction behavior influenced by the context in which they are situated [2]. Specifically, a speaker can significantly affect a listener through various factors such as tone of voice, choice of words, body language, and emotional expressiveness. These elements influence the listener's perceptions, emotions, and responses, thereby shaping the overall communication and interaction dynamics.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 233–248, 2025.
https://doi.org/10.1007/978-3-031-78389-0_16

In recent years, there has been an increasing number of studies focusing on the analysis of human-human dyadic interactions [3]. These studies aim to understand the intricacies of interpersonal communication by examining verbal and non-verbal cues, emotional exchanges, and the dynamics of social interactions within dyadic contexts. The automated generation of natural facial and bodily reactions, which mimic the behaviors of conversational partners, has been investigated extensively in several studies [4–8]. These studies primarily focused on replicating specific real facial reactions that correspond to the behavior of the input speaker. However, the potential divergence of non-verbal reaction labels for similar speaker behaviors during the training phase presents challenges for this approach.

When understanding and replicating the nuanced feedback from listeners presents a new and intriguing challenge, the Responsive Listening Head Generation task was introduced in the computer vision field by Zhou et al. [9]. Although studies such as [4,9] focused on the nonverbal facial feedback listeners provide to speakers during dyadic conversations, their primary aim was to generate reactions that mirror a ground-truth response and typically employed deterministic models to replicate precise reactions. To capture motion that represents the inherently non-deterministic nature of different perceptually plausible listeners, Learning2Listen [10] introduced a framework designed to model interactional communication in dyadic conversations. It processes multimodal inputs from a speaker and autonomously produces multiple potential listener motions in an autoregressive manner, however, the one-dimensional discrete codebook they used limited the diversity of motion and emotional representation. Later, a study by [11] introduced the novel concept of the Facial Multiple Appropriate Reaction Generation task, pioneering its definition within the literature. This study also presented novel objective evaluation metrics tailored to assess the appropriateness of generated reactions. Following the concepts introduced in [11], this research aims to advance the automatic generation of multiple appropriate non-verbal facial reactions that correspond to specific speaker behaviors.

To tackle this challenge, we propose a novel two-stage, non-autoregressive diffusion architecture for the synthesis of dyadic reactions, also known as Facial Multiple Appropriate Reaction Generation (fMARG). The primary contributions of our work include:

- Leveraging the power of the non-autoregressive Latent Diffusion Model [12] as our approach for dyadic reaction generation.
- We enhance the latent space representation through a context-aware autoencoder designed to learn spatio-temporal features of the lower facial representation features.
- We conduct extensive experiments on the REACT2024 dataset [13], demonstrating that our model significantly outperforms recent methods in generating facial reactions.

2 Related Works

2.1 Deterministic reaction synthesis

In recent decades, research on listening reaction modeling has focused on simulating engaged listeners' facial expressions and head movements. Gillies et al. [14] pioneered a data-driven approach to create an animated character capable of dynamically responding to the speaker's voice. Ahuja et al. [15] focus on generating non-verbal body behaviors. In contrast, Greenwood et al. [16] explore the synchronized motion of conversational agents in dyadic interactions, with adaptations based on speech. RealTalk [17] utilizes a large language model to retrieve potential videos of the listener's facial expressions. Huang et al. [5] trained a conditional Generative Adversarial Network [18] (GAN) to generate realistic facial reaction sketches of listeners based on the corresponding facial action units (AUs) of the speaker. Song et al. [7,8] suggest exploring person-specific networks tailored to individual listeners, enabling the reproduction of each listener's unique facial reactions.

Several studies have explored the generation of diverse non-verbal behaviors, such as hand gestures, posture, and facial reactions, in face-to-face interactions [4,19]. Zhou et al. [9] were the first to introduce the Responsive Listening Head Generation task, which involves generating a head video of a listener based on a talking-head video of the speaker and an image of the listener's face. They also developed the ViCo dataset to facilitate the evaluation of methods for this task. Their baseline approach utilized an LSTM-based model [20] to process visual and audio data from the speaker to generate facial 3D morphable model (3DMM) [21] coefficients for the listener. Although the former methods could generate listening reaction attributes based on specific speaker behavior inputs, as deterministic models, they lack diversity which is the key to real-world face-to-face scenarios.

2.2 Multiple Reaction Generation

When deterministic approaches grapple with the challenge of the 'one-to-many mapping' problem, where a single speaker behavior can evoke multiple distinct facial reactions, several studies have begun exploring the non-deterministic aspect of this problem, aiming to predict diverse facial reactions from the same input. Ng et al. [10] introduced a novel approach for modeling dyadic communication by predicting multiple realistic facial motion responses from speaker inputs using a motion-audio cross-attention transformer and a motion-encoding VQ-VAE [22] for non-deterministic prediction. This method advances beyond existing work by effectively capturing nonverbal interactions' multimodal and dynamic nature in dyadic conversations. However, expanding the one-dimensional codebook to a composition of several discrete codewords can limit motion and emotional representation diversity. Thus, the Emotional Listener Portrait (ELP) model in [23], proposed a discrete motion-codeword-based approach to generate natural and diverse non-verbal responses from listeners based on learned emotion-specific probability distributions and offering controllability.

On the other hand, according to [11], human facial reactions exhibit variability; identical or similar behaviors from speakers can prompt diverse facial responses, both across different individuals and within the same individual in different contexts. This variability poses challenges when training models to accurately reproduce the listener's facial reactions based solely on each speaker's behavior sequence. Therefore, Song et al. [11,13,24] defined the Facial Multiple Appropriate Reaction Generation (fMARG) task and introduced novel objective evaluation metrics to assess the appropriateness of generated reactions. Their framework aims to predict, generate, and evaluate multiple appropriate facial reactions and these models are successful in generating facial responses [25–28] by mapping from speaker behavior to a distribution of appropriate reactions. This paper provided an effective approach to the fMARG problem, we designed a Latent Diffusion Model for stochastic listener reaction behavior generation. Our Latent Behavior Diffusion Models accelerate sampling through diffusion in a low-resolution latent space trained by a robust context-aware auto-encoder. This approach achieves state-of-the-art performance in appropriateness, diversity, and synchrony aspects.

3 Proposed Method

Our method for generating multiple spatio-temporal reactions consists of facial Action Units, valence and arousal intensity, and facial emotion from speaker behavior with the same attributes. Diffusion models are more flexible in how they model data distributions, as they do not rely on adversarial training like GANs, or VAEs, which can suffer from mode collapse. We employ a two-stage process: a context-aware time autoencoder and a Latent Diffusion (LD) generator. The autoencoder updates global statistics iteratively during training to ensure precise reconstruction of future timestamps. Following this, the conditional LD generator utilizes a guidance mechanism to generate latent conditions, effectively incorporating relevant covariates.

3.1 Problem definition

Given the η_{th} frame ($\eta_{th} \in [t_1, t_2]$ frame) and its preceding frames expressed by the speaker S_n at the period $[t_1, t_2]$ with corresponding spatio-temporal behaviors $b_{S_n}^{t_1,t_2}$, we develop a generative model $\mathcal{G}$ that predicts each listener appropriate facial reaction frame $r_{L_n}\left(b_{S_n}^{t_1,t_2}\right)_i$. This can be formulated as:

$$r_f\left(b_{S_n}^{t_1,t_2}\right)_i^\eta = \mathcal{G}\left(b_{S_n}^{t_1,t_2}\right) \tag{1}$$

where $r_f\left(b_{S_n}^{t_1,t_2}\right)_i^\eta$ denotes the η_{th} predicted facial reaction frame of the i_{th} generated appropriate facial reaction in response to $b_{S_n}^{t_1,t_2}$; and $b_{S_n}^{t_1,\eta}$ denotes the speaker behaviour segment at the period $[t_1, \eta]$. Our approach aims to gradually generate all facial reaction frames, resulting in many appropriate spatio-temporal facial reactions (one-to-many) as described. Thus, the multiple M appropriate listener facial reaction sequences are presented as:

$$\begin{aligned} R_f\left(b_{S_n}^{t_1,t_2}\right)^\eta &= \left\{r_f\left(b_{S_n}^{t_1,t_2}\right)_1^\eta, \cdots, r_f\left(b_{S_n}^{t_1,t_2}\right)_M^\eta\right\} \\ r_f\left(b_{S_n}^{t_1,t_2}\right)_1^\eta &\neq r_f\left(b_{S_n}^{t_1,t_2}\right)_2^\eta \neq \cdots \neq r_f\left(b_{S_n}^{t_1,t_2}\right)_M^\eta \end{aligned} \tag{2}$$

Here, we simply define the set of spatio-temporal listener reaction sequences target $\mathcal{Y}$ and condition speaker behavior $\mathcal{C}$ in the time-space as:

$$\begin{aligned} \mathcal{Y} &= \left\{r_f\left(b_{S_n}^{t_1,t_1+w}\right)_M^\eta, \cdots, r_f\left(b_{S_n}^{t_2,t_2+w}\right)_M^\eta\right\}, \quad \mathcal{Y} \in \mathbb{R}^{\mathcal{T}\times d} \\ \mathcal{C} &= \{b_{S_n}^{t_1-w,t_1}, \cdots, b_{S_n}^{t_2-w,t_2}\}, \quad \mathcal{C} \in \mathbb{R}^{\mathcal{T}\times d} \end{aligned} \tag{3}$$

where $\mathcal{T}$ typically represents the number of time steps or temporal observations and d represents the dimensionality of the features at each time step.

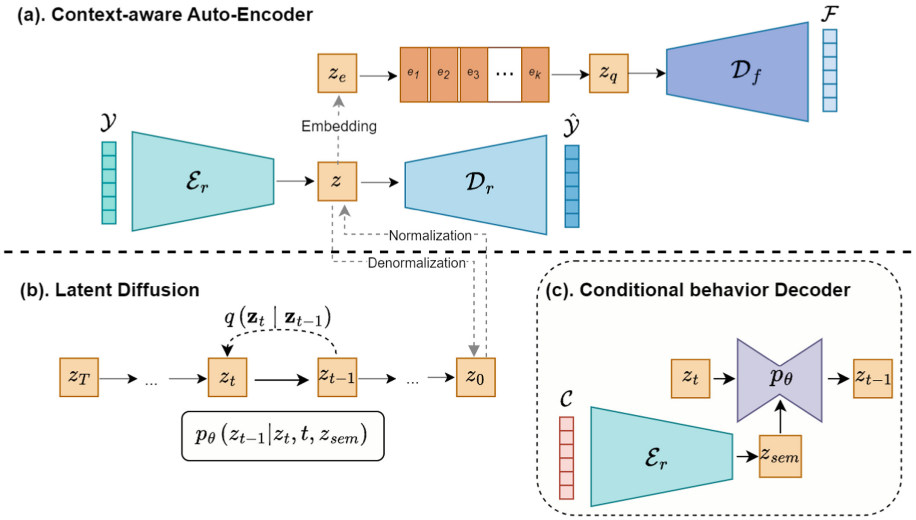


Fig. 1. Overview of our proposed Latent Diffusion Model (LDM) for generating multiple reactions. During the training phase, the autoencoder (AE) is first trained to encode the time series of listener reactions through a reconstruction task. Concurrently, the LDM is trained to predict future targets based on speaker behaviors $\mathcal{C}$. During the sampling phase, the latent representation of the time series is first generated by the LDM and then passed as input to the decoder $\mathcal{D}_r(z_0)$ to obtain the future targets.

3.2 Facial Reaction Compression

An Auto-Encoder uses backpropagation to generate an output vector similar to its input. It compresses input data into a lower-dimensional space and then reconstructs the original data from this compact representation. In our approach, we implement a context-aware Auto-Encoder (see Fig. 1a) with the ability to encode and decode sequences while maintaining awareness of the context or temporal dependencies inherent in the facial reaction sequence data.

The encoder transforms the input $\mathcal{Y}$ into a low-dimensional continuous latent space $V \subset \mathbb{R}^v$ (feature code) using a deterministic mapping function: $z = \mathcal{E}_r(\mathcal{Y})$. Samples $z \in V$ can be sampled and then reconstructed into the original facial reaction by a decoder $\mathcal{D}_r$ as $\hat{\mathcal{Y}} = \mathcal{D}_r(z)$. For generating listener 3DMM coordinates, a Vector Quantized technique is performed to produce discrete latent representation z_q by a codebook (see Fig. 1a). The 3DMM head motion coefficient set is generated as $\mathcal{F} = \mathcal{D}_f(z_q)$

Our Auto-Encoder is trained with a fix-length target listener facial reaction sequence w with Mean Square Error (L_2) regularization for $\mathcal{D}_r$ and $\mathcal{D}_f$ solely. Specifically, we use VQ-VAE loss that is composed of three components: reconstruction loss which optimizes the encoder and decoder; codebook loss to bypass the embedding as the codebook learning by L_2 error; and commitment loss to make sure the encoder commits to an embedding for $\mathcal{D}_f$.

$$\begin{aligned}
\mathcal{L}_{react} &= \sum_{t=1}^{T} \left\| r_f \left(b_{S_n}^{t,t+w}\right)_i^{\eta} - \hat{r_f} \left(b_{S_n}^{t,t+w}\right)_i^{\eta} \right\|_2 \\
\mathcal{L}_{face} &= \log p\left(\mathcal{Y} \mid z_q(\mathcal{Y})\right) \\
&\quad + \left\| \mathrm{sg}\left[z_e(\mathcal{Y})\right] - e \right\|_2^2 \\
&\quad + \beta \left\| z_e(\mathcal{Y})) - \mathrm{sg}[e] \right\|_2^2
\end{aligned} \tag{4}$$

In terms of decoding the listener's spatio-temporal facial reaction, our context-aware Auto-Encoder learns the prior distribution $p(z)$ without a standard Gaussian $\mathcal{N}(0,1)$ or 1-D codebook as it can only produce deterministic outcomes, however, it does not suffer posterior collapse and codebook collapse on a very high-dimension multivariate reaction time series and our posterior Diffusion model can compensate the ability with stochastic inference.

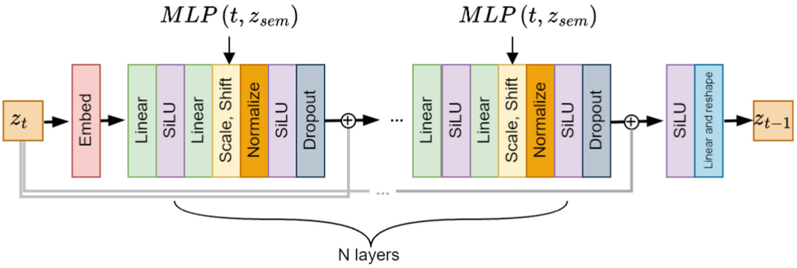


Fig. 2. Conditional Behavior Decoder architecture in Latent space.

3.3 Latent Behavior Diffusion

We propose a straightforward adaptation of conditional Latent Diffusion Models (LDMs) for reaction generation.

Conditional behavior Decoder. In the pursuit of a meaningful latent code, for each denoising step, a sequence of residual MLPs based on Diffusion autoencoders [29] is implemented as demonstrated in Fig. 1c. Each layer of the MLP has a skip connection from the input, which concatenates the input with the output from the previous layer (see Fig. 2). Hence, the conditional behavior decoder $p_\theta\left(z_{t-1}|z_t, t, z_{sem}\right)$ is conditioned by a semantic behavior encoder $z_{sem} = \mathcal{E}_r(\mathcal{C})$ which reemployed from our First stage Auto-Encoder. In practice, the learned $\mathcal{E}_r$ can deterministic map an input speaker behaviors $\mathcal{C}$ to a semantically meaningful z_{sem}. Here, our conditional behavior decoder takes the high-level semantic subcode z_{sem} and the low-level stochastic subcode z_T. In sampling process, our approach by reversing the generative process of Pseudo Linear Multi-step (PLMS) [30] to infer z_T.

Diffusion Models. Diffusion Models [31] are probabilistic generative models designed to learn the original data distribution $P(x)$ by progressively denoising variables sampled from a normal distribution. This process can be viewed as learning the reverse steps of a fixed Markov chain with step length $T \in \mathbb{N}^+$. During each step, the diffusion model employs a noise predictor to estimate the noise added in the forward Markov process and then denoise it, effectively refining the data towards its original distribution. The diffusion process is a Markov process, which incrementally applies the forward transition kernel:

$$\begin{aligned} & q\left(\mathbf{z}_{1:T} \mid \mathbf{z}_0\right) = \prod_{t=T}^{1} q\left(\mathbf{z}_t \mid \mathbf{z}_{t-1}\right) \\ & q\left(\mathbf{z}_t \mid \mathbf{z}_{t-1}\right) \sim \mathcal{N}\left(\sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t \mathbf{I}\right) \end{aligned} \tag{5}$$

where β_t determines the noise strength at each step, referred to as the variance schedule. T represents the total number of steps in the denoising process and $t = 0, 1, \cdots, T-1$.

The reverse process has a similar form to the diffusion process. During training, the network predicts $p_\theta(z_{t-1}|z_t)$ by reversing the Markov chain of length M and using a Conditional behavior Decoder $p_\theta(\cdot)$ presented as:

$$\begin{aligned} & p\left(\mathbf{z}_{0:T}\right) = p\left(\mathbf{z}_T\right) \prod_{t=t}^{T} p_\theta\left(z_{t-1}|z_t, t, z_{sem}\right) \\ & p\left(\mathbf{z}_T\right) = \mathcal{N}(0, \mathbf{I}) \\ & q\left(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathcal{Y}\right) = \mathcal{N}\left(\mu_t\left(\mathbf{z}_t, \mathcal{Y}\right), \sigma_t^2 \mathbf{I}\right) \end{aligned} \tag{6}$$

where $\mu_t\left(\mathbf{z}_t, \mathcal{Y}\right)$ has a closed-form solution and σ_t is a hyperparameter.

Objective Functions. The LDM's loss is the average of the Mean Absolute Error (MAE) in the latent space and the Mean Square Error (MSE) in the reconstructed space. The complete objective training loss function for our LDM is expressed as follows:

$$\begin{aligned}\mathcal{L}_{latent} &= \sum_{t=1}^{T} \underset{q(z_t|z_0)}{\mathbb{E}} \|p_\theta\left(z_{t-1}|z_t, t, z_{sem}\right) - \mathcal{E}_r(\mathcal{Y})\|_1 \\ \mathcal{L}_{rec} &= \sum_{t=1}^{T} \underset{q(z_t|z_0)}{\mathbb{E}} \|\mathcal{D}_r(p_\theta\left(z_{t-1}|z_t, t, z_{sem}\right)) - \mathcal{Y}\|_2 \\ \mathcal{L}_{ldm} &= \mathcal{L}_{latent} + \mathcal{L}_{rec}\end{aligned} \tag{7}$$

Latent Behavior Sampler. Pseudo Linear Multi-Step [30] (PLMS) is an improvement over DDIM . According to [30], a 50-step process can achieve higher quality than a 1000-step process in DDIM. We proposed a Latent Behavior Sampler that can be broken down into a series of formulas capturing the core logic of sampling $z_{t+\delta}$ from the model using the PLMS method. Following the forward Euler Method, for a certain differential equation satisfying $\frac{dx}{dt} = f(z, t)$. We represent PLMS in formulaic terms:

$$\begin{aligned} z_{t+\delta} &= z_t + \delta\left(f_t\right) && \text{if } k = 1 \\ z_{t+\delta} &= z_t + \frac{\delta}{2}\left(3f_t - f_{t-\delta}\right) && \text{if } k = 2 \\ z_{t+\delta} &= z_t + \frac{\delta}{12}\left(23f_t - 16f_{t-\delta} + 5f_{t-2\delta}\right) && \text{if } k = 3 \\ z_{t+\delta} &= z_t + \frac{\delta}{24}\left(55f_t - 59f_{t-\delta} + 37f_{t-2\delta} - 9f_{t-3\delta}\right) && \text{if } k = 4 \end{aligned} \tag{8}$$

where $f_t = f\left(z_t, t\right)$, k is the convergence order and δ is the step size. Our inference process leverages the reverse PLMS process when it reaches $T = 0$, obtaining the future target by $\mathcal{D}_r(z_0)$.

4 Experiments

4.1 Evaluation setup

Dataset. The REACT2024 dataset [11,13] is a comprehensive resource for analyzing dyadic video interactions with detailed facial attribute annotations. This dataset is constructed from two prominent video conference corpora: NoXi [32] and RECOLA [33]. Each audio-video clip from the NoXi and RECOLA datasets has been segmented into 30-second clips, resulting in conversational 5,919 clips. Specifically, the dataset includes extensive facial attribute annotations for each frame, derived using state-of-the-art models [34–36]: Action Units (AUs): 15

AUs are annotated, including AU1, AU2, AU4,... Facial Affects: Two continuous affective states, valence and arousal intensities, are provided; Facial Expression Probabilities: Eight facial expression probabilities are included, covering Neutral, Happy, Sad,... and further the extracted 3DMM parameters. The carefully segmented and cleaned clips, rich annotations, and special appropriateness label strategy provide a robust foundation for training and evaluating advanced machine-learning models in dyadic emotion recognition, facial expression analysis, interaction dynamics in video conferencing scenarios, and indeed capability for reaction generation.

Comparison Methods. We compare our method with two baseline approaches provided by [13,24]: TransVAE and Belfusion. The Trans-VAE baseline shares a similar architecture to the TEACH model proposed in [37], a Multimodal Transformer-based VAE that takes video facial and audio embedding from speaker to predict listener facial reaction features and 3DMM parameters. Belfusion is based on the work in [38], employing DDIM model with standard Gaussian Distribution as the prior. Furthermore, some generative methods were remarkable at REACT2024 Challenge [13] included in our comparison. In particular, Dam et al. [25] designed an architecture encouraged by [10], but leveraging Finite Scalar Quantization [39] to replace Vector Quantization. Besides, Liu et al. [26] introduced discrete latent variables to tackle this one-to-many mapping problem, to model the diversity of contextual factors, and to generate diverse reactions.

Implementation Details. We implement our model using PyTorch [40] and perform the training on a single Nvidia RTX 3080Ti GPU. In the first stage of our autoencoder structure, we implemented based on Transformer-based [41] architectures, all $\mathcal{E}_r$, $\mathcal{D}_f$, and $\mathcal{D}_r$ utilized two layers of Transformer encoders, each with 4 attention heads. The latent Conditional Behavior Decoder includes MLP + Skip with 10 layers and 1024 hidden nodes, detailed in Fig. 2. The AE models are solely trained with 1000 epochs for $\mathcal{D}_r$ and 200 epochs for $\mathcal{D}_f$ with a batch size of 32. The window size is set to 50, the learning rate is 1e-3, and the weight decay is 5e-4 with the AdamW optimizer. We adjusted the same optimizer parameters for the second stage and trained the LDM for 200 epochs. The denoising chain has 50 steps. Sampling was conducted with our fourth-order PLMS latent behavior sampler.

4.2 Evaluation metric

Following the standard protocols proposed in [11], we evaluate our method based on three key aspects of the generated facial reaction attributes as below:

Appropriateness: Facial reaction distance (FRDist) calculates the Dynamic Time Warping (DTW) distance between a generated facial reaction and its closest corresponding real facial reaction; Facial Reaction Correlation (FRCorr) computes the correlation between each generated facial reaction and its most similar corresponding real facial reaction.

Diversity: Facial Reaction Variance (FRVar) computing the variation across all frames; Diverseness among generated facial reactions (FRDiv); Diversity among facial reactions generated from different speaker behaviors (FRDvs).

Synchrony (FRSyn) is computed by first calculating the Time-Lagged Cross-Correlation (TLCC) scores between the input speaker behavior and each of its generated facial reactions.

Table 1. Comparision against various methods on Multiple Facial Reaction Generation on REACT2024 dataset.

Method	Appropriateness		Diversity			Synchrony
	FRCorr(↑)	FRDist(↓)	FRDiv(↑)	FRVar(↑)	FRDvs(↑)	FRSyn(↓)
Trans-VAE	0.07	90.31	0.0064	0.0012	0.0009	44.65
BeLFusion	0.12	94.09	0.0379	0.0248	0.0397	49.00
Dam et al. [25]	0.31	**84.94**	0.1167	0.0349	0.1165	47.43
Liu et al. [26]	0.22	88.32	0.1030	0.0387	0.1065	44.41
Ours	**0.37**	89.40	**0.1211**	**0.0653**	**0.1505**	**43.48**

Table 2. Comparisons on image-level in terms of quality and identity preservation.

Method	FID (↓)
GT	53.96
Trans-VAE	69.19
Belfusion	54.00
Ours	**50.95**

4.3 Results

Quantitative Results. Table 1 illustrates the quantitative evaluation results of our proposed LDM compared with other methods. The results demonstrate that our approach generates facial reactions with greater diversity and synchrony than those produced by the competitors. Trans-VAE and Belfusion got the worst performance among our comparison, these models relied on standard Gaussian distribution can easily lead to posterior collapse during training with Trans-VAE and prior VAE of Belfusion. Dam et al. [25] and Liu et al. [26] tackled this problem by applying discrete latent space. Nevertheless, our Latent Diffusion model with a Transformer AE prior can surpass the overall evaluation criteria, delivering superior performance in generating diverse and synchronized facial reactions. Moreover, the better diversity and synchrony metrics state that our Decoder can better generate responsive reactions and avoid jitters in the facial frames among generated fix-length segments. Although there was a trade-off regarding FRDist, its performance remains competitive and within a fair margin compared to other approaches.

To comprehensively evaluate the video-level performance, we use the Fréchet Inception Distance (FID) [42] as a commonly used metric for assessing the realism of generated human faces. We evaluate the realism of the facial reactions that PIRender [43] generated from 3DMM parameters by $\mathcal{D}_f$, results are shown in Table 2.

Qualitative Results. In this section, we present the qualitative results of the generated facial frames of multiple listeners. These results are illustrated in Fig. 3 and Fig. 4. Trans-VAE failed to preserve identity, generating unclear expressions and arbitrary motions. Belfusion's results show poor variation and less dynamic motions due to the weakness of its Gaussian latent space. In contrast, our model achieves the best natural and coherent results compared to these baseline methods.

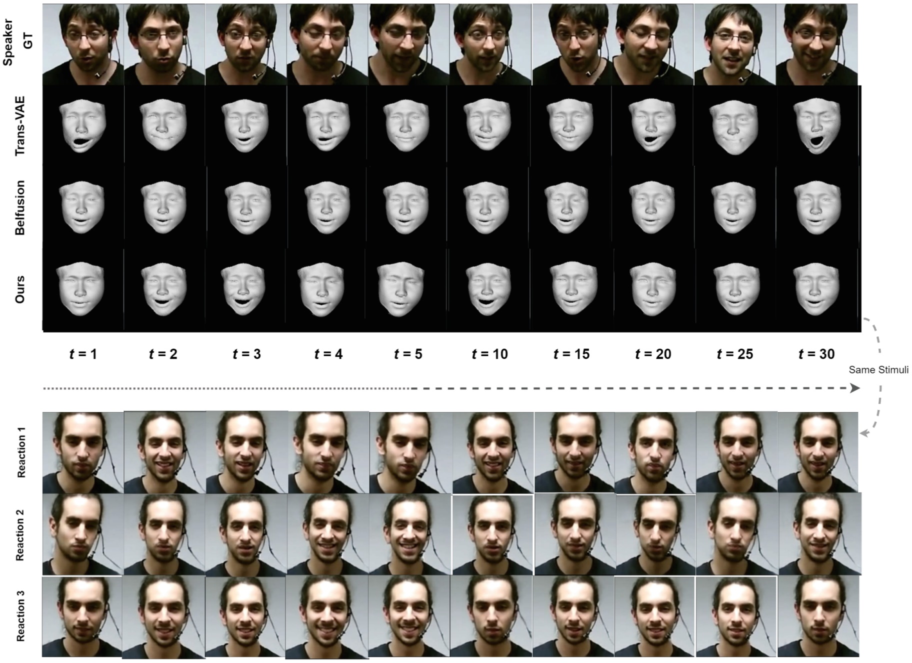


Fig. 3. The qualitative results of the generated facial frames of multiple listeners. Our comparison between other baselines by 3D rendering translation. The bottom of the figure shows our model-generated variants of reaction that are expressed from Speaker ground truth. The time t in second.

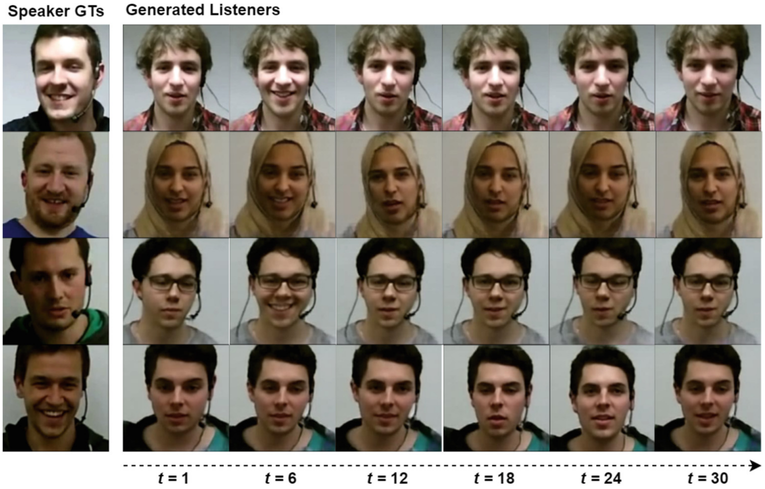


Fig. 4. Generated listener facial reactions from different speakers.

Table 3. Effect of ablating key settings of our method.

Method	Appropriateness		Diversity			Synchrony
	FRCorr(↑)	FRDist(↓)	FRDiv(↑)	FRVar(↑)	FRDvs (↑)	FRSyn(↓)
PLMS (k=2, T=10)	0.36	89.27	0.1157	0.0627	0.1434	44.16
PLMS (k=2, T=25)	0.37	91.99	0.1190	0.0699	0.1605	45.47
PLMS (k=1, T=50)	0.372	89.49	0.1211	0.0653	0.1505	43.88
PLMS (k=2, T=50)	0.372	89.48	0.1206	0.0651	0.1501	43.68
PLMS (k=3, T=50)	0.372	89.54	0.1208	0.0651	0.1501	43.69
PLMS (k=4, T=50)	**0.374**	89.40	**0.1211**	**0.0653**	**0.1505**	**43.48**
PLMS (k=2, T=100)	0.33	88.22	0.1064	0.0619	0.1405	45.03
DDIM (T=50)	0.35	88.27	0.1047	0.0579	0.1343	44.00
DPM (T=50)	0.32	**86.53**	0.1014	0.0546	0.1223	43.82

Note: T is the number of denoise steps and k is convergence order of PLMS

4.4 Ablation Study

We quantitatively analyze the effect of each of the settings in our method for the final LD model with Trans-AE prior. We perform ablation studies with different denoise chain step numbers $T \in (10, 25, 50, 100)$ and four convergence orders of PLMS method.

Table 3 demonstrates that increasing the number of denoising steps slightly improves the alignment of predicted reactions with the ground truth in terms of Dynamic Time Warping (DTW). However, this improvement negatively impacts the Correlation aspect. Through our experiments, we identified an optimal balance between Appropriateness and Diversity at 50 denoising steps. The impact of varying PLMS convergence orders is minimal, with our best results achieved using the fourth-order implementation. Furthermore, we applied DDIM and the origin DPM samplers with the same 50 denoising steps, and our PLMS achieved higher Correlation and Diversity in our comparison.

5 Conclusion

In this paper, our study introduces Latent Behavior Diffusion as a robust framework for generating diverse and contextually appropriate facial reactions in dyadic interactions. By combining a context-aware autoencoder with a diffusion-based conditional generator, our approach effectively compresses high-dimensional input features into a concise latent representation. Our approach attains superior performance in objective benchmarks of generated facial reactions compared to existing methods. We further aim to improve the appropriateness of DTW by refining denoising process for better long-term reaction sequence modeling.

Acknowledgements. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00219107). This work was also supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP- 2023-RS-2023-00256629) grant funded by the Korean government (MSIT).

References

1. Albert Mehrabian and James A Russell. *An approach to environmental psychology.* the MIT Press, 1974
2. Xuesong Zhai, Minjuan Wang, and Usman Ghani. The sor (stimulus-organism-response) paradigm in online learning: an empirical study of students' knowledge hiding perceptions. In *Cross Reality (XR) and Immersive Learning Environments (ILEs) in Education*, pages 48–63. Routledge, 2023
3. Peng, S., Dong, Y., Wang, W., Jieyi, H., Dong, W.: The affective facial recognition task: The influence of cognitive styles and exposure times. J. Vis. Commun. Image Represent. **65**, 102674 (2019)
4. German Barquero, Johnny Núnez, Zhen Xu, Sergio Escalera, Wei-Wei Tu, Isabelle Guyon, and Cristina Palmero. Comparison of spatio-temporal models for human motion and pose forecasting in face-to-face interaction scenarios supplementary material. 2022
5. Yuchi Huang and Saad M Khan. Dyadgan: Generating facial expressions in dyadic interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 11–18, 2017
6. Cristina Palmero, German Barquero, Julio CS Jacques Junior, Albert Clapés, Johnny Núnez, David Curto, Sorina Smeureanu, Javier Selva, Zejian Zhang, David Saeteros, et al. Chalearn lap challenges on self-reported personality recognition and non-verbal behavior forecasting during social dyadic interactions: Dataset, design, and results. In *Understanding Social Behavior in Dyadic and Small Group Interactions*, pages 4–52. PMLR, 2022

7. Zilong Shao, Siyang Song, Shashank Jaiswal, Linlin Shen, Michel Valstar, and Hatice Gunes. Personality recognition by modelling person-specific cognitive processes using graph representation. In *proceedings of the 29th ACM international conference on multimedia*, pages 357–366, 2021
8. Song, S., Shao, Z., Jaiswal, S., Shen, L., Valstar, M., Gunes, H.: Learning person-specific cognition from facial reactions for automatic personality recognition. IEEE Trans. Affect. Comput. **14**(4), 3048–3065 (2022)
9. Mohan Zhou, Yalong Bai, Wei Zhang, Ting Yao, Tiejun Zhao, and Tao Mei. Responsive listening head generation: a benchmark dataset and baseline. In *European Conference on Computer Vision*, pages 124–142. Springer, 2022
10. Evonne Ng, Hanbyul Joo, Liwen Hu, Hao Li, Trevor Darrell, Angjoo Kanazawa, and Shiry Ginosar. Learning to listen: Modeling non-deterministic dyadic facial motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20395–20405, 2022
11. Siyang Song, Micol Spitale, Yiming Luo, Batuhan Bal, and Hatice Gunes. Multiple appropriate facial reaction generation in dyadic interaction settings: What, why and how? *arXiv preprint* arXiv:2302.06514, 2023
12. Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022
13. Siyang Song, Micol Spitale, Cheng Luo, Cristina Palmero, German Barquero, Hengde Zhu, Sergio Escalera, Michel Valstar, Tobias Baur, Fabien Ringeval, et al. React 2024: the second multiple appropriate facial reaction generation challenge. *arXiv preprint* arXiv:2401.05166, 2024
14. Gillies, M., Pan, X., Slater, M., Shawe-Taylor, J.: Responsive listening behavior. Computer animation and virtual worlds **19**(5), 579–589 (2008)
15. Chaitanya Ahuja, Shugao Ma, Louis-Philippe Morency, and Yaser Sheikh. To react or not to react: End-to-end visual pose forecasting for personalized avatar during dyadic conversations. In *2019 International conference on multimodal interaction*, pages 74–84, 2019
16. David Greenwood, Stephen Laycock, and Iain Matthews. Predicting head pose in dyadic conversation. In *Intelligent Virtual Agents: 17th International Conference, IVA 2017, Stockholm, Sweden, August 27-30, 2017, Proceedings 17*, pages 160–169. Springer, 2017
17. Scott Geng, Revant Teotia, Purva Tendulkar, Sachit Menon, and Carl Vondrick. Affective faces for goal-driven dyadic communication. *arXiv preprint* arXiv:2301.10939, 2023
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Bing, X., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
19. Nguyen Tan Viet Tuyen and Oya Celiktutan. Context-aware human behaviour forecasting in dyadic interactions. In *Understanding Social Behavior in Dyadic and Small Group Interactions*, pages 88–106. PMLR, 2022
20. Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012
21. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In Seminal Graphics Papers: Pushing the Boundaries **2**, 157–164 (2023)
22. Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017

23. Luchuan Song, Guojun Yin, Zhenchao Jin, Xiaoyi Dong, and Chenliang Xu. Emotional listener portrait: Neural listener head generation with emotion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20839–20849, 2023
24. Siyang Song, Micol Spitale, Cheng Luo, German Barquero, Cristina Palmero, Sergio Escalera, Michel Valstar, Tobias Baur, Fabien Ringeval, Elisabeth Andre, et al. React2023: the first multi-modal multiple appropriate facial reaction generation challenge. *arXiv preprint* arXiv:2306.06583, 2023
25. Quang Tien Dam, Tri Tung Nguyen Nguyen, Dinh Tuan Tran, and Joo-Ho Lee. Finite scalar quantization as facial tokenizer for dyadic reaction generation
26. Zhenjie Liu, Cong Liang, Jiahe Wang, Haofan Zhang, Yadong Liu, Caichao Zhang, Jialin Gui, and Shangfei Wang. One-to-many appropriate reaction mapping modeling with discrete latent variable
27. Dang-Khanh Nguyen, Prabesh Paudel, Seung-Won Kim, Ji-Eun Shin, Soo-Hyung Kim, and Hyung-Jeong Yang. Multiple facial reaction generation using gaussian mixture of models and multimodal bottleneck transformer. In *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–5. IEEE, 2024
28. Minh-Duc Nguyen, Hyung-Jeong Yang, Ngoc-Huynh Ho, Soo-Hyung Kim, Seungwon Kim, and Ji-Eun Shin. Vector quantized diffusion models for multiple appropriate reactions generation. In *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–5. IEEE, 2024
29. Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10619–10629, 2022
30. Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint* arXiv:2202.09778, 2022
31. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
32. Angelo Cafaro, Johannes Wagner, Tobias Baur, Soumia Dermouche, Mercedes Torres Torres, Catherine Pelachaud, Elisabeth André, and Michel Valstar. The noxi database: multimodal recordings of mediated novice-expert interactions. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 350–359, 2017
33. Fabien Ringeval, Andreas Sonderegger, Juergen Sauer, and Denis Lalanne. Introducing the recola multimodal corpus of remote collaborative and affective interactions. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pages 1–8. IEEE, 2013
34. Cheng Luo, Siyang Song, Weicheng Xie, Linlin Shen, and Hatice Gunes. Learning multi-dimensional edge feature-based au relation graph for facial action unit recognition. *arXiv preprint* arXiv:2205.01782, 2022
35. Siyang Song, Yuxin Song, Cheng Luo, Zhiyuan Song, Selim Kuzucu, Xi Jia, Zhijiang Guo, Weicheng Xie, Linlin Shen, and Hatice Gunes. Gratis: Deep learning graph representation with task-specific topology and multi-dimensional edge features. *arXiv preprint* arXiv:2211.12482, 2022
36. Toisoul, A., Kossaifi, J., Bulat, A., Tzimiropoulos, G., Pantic, M.: Estimation of continuous valence and arousal levels from faces in naturalistic conditions. Nature Machine Intelligence **3**(1), 42–50 (2021)

37. Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. Teach: Temporal action composition for 3d humans. In *2022 International Conference on 3D Vision (3DV)*, pages 414–423. IEEE, 2022
38. German Barquero, Sergio Escalera, and Cristina Palmero. Belfusion: Latent diffusion for behavior-driven human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023
39. Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint* arXiv:2309.15505, 2023
40. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019
41. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017
42. Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017
43. Yurui Ren, Ge Li, Yuanqi Chen, Thomas H Li, and Shan Liu. Pirenderer: Controllable portrait image generation via semantic neural rendering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13759–13768, 2021

CFTS-GAN: Continual Few-Shot Teacher Student for Generative Adversarial Networks

Munsif Ali(✉), Leonardo Rossi, and Massimo Bertozzi

Department of Engineering and Architecture, University of Parma, Parma, Italy
{munsif.ali,leonardo.rossi,massimo.bertozzi}@unipr.it

Abstract. Few-shot and continual learning face two well-known challenges in GANs: overfitting and catastrophic forgetting. Learning new tasks results in catastrophic forgetting in deep learning models. In the case of a few-shot setting, the model learns from a very limited number of samples (e.g. 10 samples), which can lead to overfitting and mode collapse. So, this paper proposes a Continual Few-shot Teacher-Student technique for the generative adversarial network (CFTS-GAN) that considers both challenges together. Our CFTS-GAN uses an adapter module as a student to learn a new task without affecting the previous knowledge. To make the student model efficient in learning new tasks, the knowledge from a teacher model is distilled to the student. In addition, the Cross-Domain Correspondence (CDC) loss is used by both teacher and student to promote diversity and to avoid mode collapse. Moreover, an effective strategy of freezing the discriminator is also utilized for enhancing performance. Qualitative and quantitative results demonstrate more diverse image synthesis and produce qualitative samples comparatively good to very stronger state-of-the-art models.

1 Introduction

Continual Learning (CL) and Few-shot (FS) learning are two very important problems to take into account, which affect the applicability of deep learning systems in real cases [37,38]. Continual learning makes the model capable of learning new tasks without affecting the previously learned tasks. However, learning a new task in deep learning models leads to a well-known problem named catastrophic forgetting [15,31]. In a few-shot setting, a model learns only from a few samples (e.g. ≤ 10), which can result in memorizing the training set and causes overfitting and mode collapse [1]. FS is also a considerable matter because of the unavailability of the full dataset due the data privacy concerns. Moreover, FS is an important consideration for such applications where very limited samples are available such as healthcare, cyber security, etc. Furthermore, it is also essential due to huge resource consumption, training time, and low-power devices [37,38]. Individually, continual and few-shot learning are very well explored; conversely, considering both together have gained more popularity recently [1]. So, we also

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 249–262, 2025.
https://doi.org/10.1007/978-3-031-78389-0_17

take into consideration both together and present our contributions to the challenging tasks in this paper.

In CL and FS there are two very distinct areas of application of the techniques: the ones that apply to discriminative models, such as classifiers, segmentation models, etc., and the ones that apply to generative models, such as GAN models. E.g. a survey article in [34] collected the works that considered both the CL and FS together for classification and segmentation. Various techniques exist that focus on both few-shot and continual learning to mitigate forgetting and overfitting in classification models. However, generative models still need more attention and consideration regarding continual and few-shot learning and have gained attention recently [1]. Therefore, we consider the CL and FS which apply to generative models, especially to GANs.

The CL approaches are grouped into regularization [28], replay [17], or dynamics/expansion methods [27,35,43]. Regularization approaches grant to overcome forgetting but produce blurred samples after learning many tasks [35]. In replay approaches, memory consideration limits the scalability of these approaches, and data privacy is also an important consideration where data privacy is concerned [16]. Dynamic architectures add additional parameters for CL, do not need previous data samples, and also provide good results. However, designing such architecture needs careful attention due to increasing the number of parameters [35]. On the other hand, when limited samples are available for training, instead of learning the data distribution, the model memorizes the training samples and leads to overfitting and mode collapse. Earlier approaches for FS and limited data generation used transfer learning and fine-tuning for the target domain generation [2,23,39]. Another approach is data augmentation either for the data or feature level for gaining diversity in the target generation [40]. However, these approaches are not a good choice in the case of very limited training samples [9]. In the knowledge distillation, another model is used to make the deep learning model efficient considering the FS setting [24]. However, all these approaches do not consider both the CL and FS image generation. Less work has been done considering both few-shot and continual learning together for GANs [29].

Our work proposes a continual few-shot teacher-student model for GANs (CFTS-GAN) considering the FS and CL together. Our method uses knowledge distillation, adding regularization terms for the few-shot image generation. Our CFTS-GAN model takes inspiration from CAM-GAN [35] for continual learning and from [22] for few-shot learning. CAM-GAN injects adapter modules on the top of a generator model [19] for continual image generation. It trains only the adapters when a new task is available, preserving the ability to generate images for the previous tasks. The CAM-GAN consists of a simple generator architecture compared to StyleGAN2 [12], whereas StyleGAN2 has more control over the image generation due to latent space manipulation and adding noise in every stage. We extend the CAM-GAN training with the teacher-student architecture to improve its performance and with the CDC loss to preserve diversity and avoid mode collapse. Our final architecture consists of three generators: a source, a teacher, and a student. Starting from a source generator, previously trained on a large dataset, we train CFTS-GAN in two stages. In the first stage, the teacher model is trained on the current task, preserving the generator diversity with the help of the CDC loss [22] between the source model and itself. In the second stage, when the student model is trained on the current task, the student takes advantage of the teacher, squeezing teacher knowledge inside the adapters and, at the same time, preserving generator diversity using CDC loss between the source model and itself. By applying CDC loss to both, we decrease the probability that the generators lose the ability to generate different images. CFTS-GAN also utilizes a simple technique of freezing the student's discriminator to obtain much better results [20]. The quantitative results demonstrate that our approach gains more diversity and obtains comparatively quality samples compared with stronger models [29], [22], [41], and [45] which are derived from a more advanced architecture [12].

The main contributions of this paper are summarized in the following points.

- We propose a teacher-student model for continual few-shot image synthesis, to condense the knowledge of a generator into the adapters of a CAM-GAN model.
- To preserve the diversity of the image generated and prevent mode collapse due to memorization of few available examples, we employed the Cross-Domain Correspondence (CDC) loss [22] in both the teacher and student models training, introducing a source model pre-trained on a large dataset.
- To further refine the model performance, a simple strategy of freezing the discriminator in the student training is also used [20].
- To evaluate image quality and diversity, the performance of the CFTS-GAN is analyzed on different few-shot datasets using FID and B-LPIPS [29] metrics.

2 Literature Review

Continual Learning. Generative models have been recently analyzed to generate data continually and get rid of overriding new information on the older ones. A well-known approach in [13] is proposed for discriminative models where an additional loss term is added to retain the previously learned distribution. The idea of [13] is utilized and implemented in GAN [28] where an additional loss term is added only to the generator of the conditional GAN to prevent previously learned distribution from drifting and avoid forgetting. However, after too many tasks the model saturates and provides an unrealistic generation of images. Another notable approach is memory replay GAN (MeRGAN) [5] in which a previous sample is recalled during training for the current task. The previous data sample is generated using the generator which solves the issue but still provides unrealistic samples after many tasks. Moreover, rehearsal-based approaches are limited to label conditional generation [44]. To implement both label and image conditional GAN, the authors in [44] proposed an approach using knowledge distillation for a lifelong generation. However, the proposed lifelong model is shared across all tasks so the previous task generation quality is degraded as new tasks arrive. In parameter isolation-based approaches, the parameters of the previous tasks are kept unchanged while learning new tasks. One way of parameter isolation is to expand the network and each task has its own parameters [14,42]. Another way is to use the same architecture but allocate distinct parameters for each task [18,25]. For continual classification, a model is designed using universal and parameter vectors for learning shared and task-specific domains, respectively [26]. In [27], add additional layers for learning a new task and use previously learned knowledge, and fine-tune for better convergence. CAM-GAN authors [35] proposed a continual learning approach that is based on adding adapter modules for learning the upcoming tasks without affecting the previously learned distributions. Taking inspiration from them, we extended their model to work on a more challenging scenario of CL and FS, together. We also use their adapters concept on a Teacher-Student setting to enhance generation quality and CDC loss to preserve diversity of generated images.

Few-Shot Learning. Many methods exist for the few-shot GANs [6,30,32]. A simple fine-tuning FreezD method based on freezing the discriminator layers to achieve the few-shot image generation is presented in [20]. Transfer learning is also considered for the few-shot image synthesis in [39]. BSA [21] adds a small number of parameters to the source model based on the statistics of the feature map. CDC [22] preserves the diversity of the images by applying cross-domain correspondence between the source and target images. Other variants of CDC are also presented in [41] and [45]. Similar to CDC the DCL [45] uses the mutual information between the source and target domain. These approaches consider only few-shot learning while in our case we take into consideration both continual and few-shot learning.

Continual Few-Shot Learning. For discriminative models, a hyper-transformer is used for few-shot lifelong learning classification in [36]. The hyper-transformer generates the weights for learning the new task. The authors in [33] proposed another approach for the few-shot class incremental classification using a neural gas network. An expansion-based idea is presented in [46] for few-shot discrimination which tries to make the same sample closer and increase the space between different samples for few-shot settings. Conversely, generative models in terms of both continual and few-shot settings are not widely explored. According to the authors, the LFS-GAN [29] is the first approach that considers both setups together. LFS-GAN appends additional parameters for learning new tasks. The distance between generated fake samples and input noise is maximized to produce diverse images. A continual few-shot image translation is proposed in [4]. This method introduces new scale and shift parameters and modulates the older parameters to learn the newly introduced scale and shift parameters. We proposed different training approaches for continual few-shot learning utilizing the architecture of [35] introducing teacher-student architecture. Moreover, we also applied CDC loss in both training stages to obtain better diversity.

3 Method

In this section, we describe our approach for the continual few-shot image generation which is divided into several subsections. The sections 3.1 and 3.2 give details of the preliminaries of continual and few-shot learning and CDC loss. Section 3.3 describes the teacher-student model and their objectives.

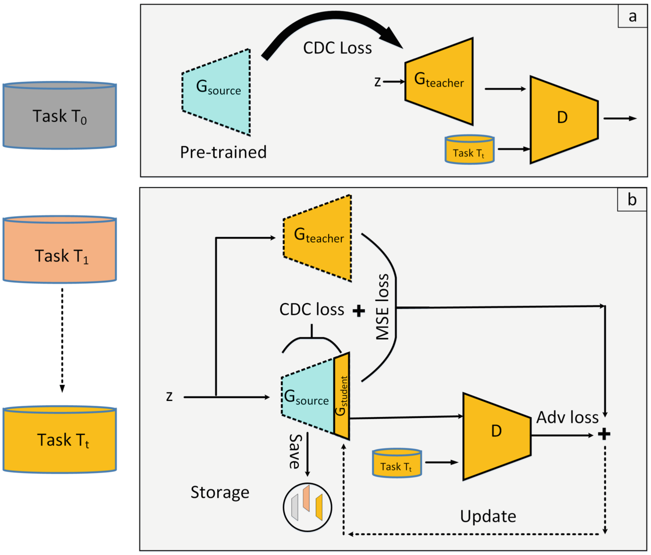


Fig. 1. The teacher-student model for continual learning GAN. The dashed lines show the frozen model. a) shows the teacher model training. b) demonstrates training of the student model.

3.1 Continual Few-Shot GAN

For continual few-shot image synthesis, a set of multiple tasks $T = \{T_0, .., T_t\}$ is considered (Figure 1). Each task T_i consists of a dataset with few samples $x_0, x_2, ...x_n$, where n is the total number of samples in the training set. If n is low (i.e. $n \leq 10$), we are in the case of few-shot learning. As in CL setting, when the model is trained on a new task T_i, previous training data are no longer available and then can not be used for this training step.

GANs are usually composed by two models: a discriminator D and generator G. The objective of D is to discern real data from synthetic samples generated by G. While the objective of G is to generate samples that look real to fool D. The generator learns parameters θ from a probability distribution $p_{data}(x)$ for a given dataset to produce synthetic data. These fake images are generated from a random Gaussian distribution $p(z)$. Both D and G are trained together in an adversarial manner to achieve the following objective [7]

$$\min_G \max_D \mathbb{E}_{x \sim p_{\mathrm{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_{\mathrm{z}}(z)}[1 - \log D(G(z))] + R(\theta_d) \tag{1}$$

where $p_{data}(x)$ is the distribution of real data and $p_z(z)$ is a random Gaussian distribution noise, $R(.)$ is a regularization term [19].

For each task T_i, our training is composed by two stages. In the first stage, a teacher model is trained on the FS dataset for task T_i. In the second stage, to exploit the knowledge distillation [3] using the teacher, the student model is trained. To enhance the diversity of the generated examples and avoid mode collapse, the Cross Domain Consistency (CDC) loss is employed for both training, teacher, and student, in addition to adversarial loss, as explained in the following paragraphs.

In the case of FS, due to limited samples, the discriminator part of the GAN also overfits and affects the generator training. So, freezing the discriminator is one of the strategies to avoid the overfitting of it [20].

3.2 Cross Domain Consistency Loss

Considering that few-shot learning leads to mode collapse in GANs, data augmentation is one possible solution for diverse image generation but it works better when the training samples are in a much larger number [9]. Since we are in a much more restrictive situation, to preserve the source diversity and avoid mode collapse, we took inspiration from the [22]. During the training process of the generator (teacher or student, depending if we are in the first or second stage) on the dataset for the task T_i, another frozen generator, called source, is incorporated into the training procedure.

As previously mentioned, we consider two generators in this scenario: G_{source}, trained on large dataset (in our case, trained on the FFHQ dataset), and G_{target}, which is trained using a few-shot learning method on dataset for the task T_i. At each training step i, two noise vectors z_j and z_k are sampled and passed to both the G_{source} and G_{target}. The probability distribution between different noise vectors for the source and target is given as

$$\begin{aligned} y_{source} &= \operatorname{Softmax}\left(\left\{\operatorname{sim}\left(G_{source}\left(z_j\right), G_{source}\left(z_k\right)\right)\right\}_{\forall j \neq k}\right) \\ y_{target} &= \operatorname{Softmax}\left(\left\{\operatorname{sim}\left(G_{target}\left(z_j\right), G_{target}\left(z_k\right)\right)\right\}_{\forall j \neq k}\right) \end{aligned} \tag{2}$$

where y_{source} and y_{target} are the probability of the source and target, sim denotes the cosine similarity, and $G_{source}(z)$ and $G_{target}(z)$ are the output layer for the source and target generators. We use the KL divergence to encourage the target generator to have a similar distribution of the source generator. This loss is called CDC loss and is given as:

$$\mathcal{L}_{\mathrm{cdc}}\left(G_{source}, G_{target}\right) = KL\left(y_{target} \| y_{source}\right). \tag{3}$$

In our case, our target generator G_{target} will be the teacher or the student, depending if we are in the first or second stage. In the following sections, we will use the notation $G_{teacher}$ and $G_{student}$ instead of the target G_{target} for the corresponding teacher and student training.

3.3 Teacher Student Model

Our generator model, based on CAM-GAN [35], injects adapter modules on the top of GP-GAN [19] for continual learning. The CAM-GAN consists of global weights θ_{global} and adapter weights $\theta_{adapter}$. The latter are used for learning the upcoming tasks while freezing the global weights, so as not to affect the previous tasks generation.

Our continual few-shot image generation training is composed of two stages, and takes into account three generators: source, teacher, and student. All of them have the same architecture but each one is used in a particular setting and different purpose. At the beginning, the source is trained on a very large dataset. As show in Figure 1(a), in the first stage the teacher model is cloned from the source model and trained all its weights $\theta_{teacher}$. Because also the teacher is trained in a few-shot setting, to enforce the diversity, we support its training with the CDC loss [22], in addition to its adversarial loss.

As show in Figure 1(b), in the second stage the student model is trained. To achieve the continual image generation objective, we considered the student model weights $\theta_{student}$ as the combination of global weights θ_{global} and adapter weights $\theta_{adapter}$, initially cloned from the source. To learn the current few-shot task T_i, only the adapters' weights are trained, while freezing the global weights. To make the student more effective in learning the current few-shot task T_i, it takes advantage from the source using CDC loss [22], and from the teacher model, previously trained with the same dataset, using a loss of knowledge distillation [3].

Teacher objective. The teacher learning is shown in Figure 1(a). The teacher model objective uses the adversarial loss as given in the equation 1. For diverse image generation, the CDC loss is used along with the adversarial loss. So, the objective of the teacher is the sum of the adversarial loss and CDC loss for the teacher $G_{teacher}$ and is given by

$$\mathcal{L}_{teacher} = \mathcal{L}_{adv} + w_t \mathcal{L}_{\text{cdc}} \left(G_{source}, G_{teacher} \right), \tag{4}$$

where w_t is a scalar weight factor for the CDC loss. Unlike starting from scratch, the teacher model's weights are initialized with the source parameters.

Student Objective. The training process of the student model is depicted in Figure 1(b). To train the student model $G_{student}$, the knowledge from the teacher model $G_{teacher}$ is transferred to the student. To transfer the knowledge from the teacher model to the student we utilized the Mean Square Error (MSE) loss [3], termed as $\mathcal{L}_{kd}$, applied to the output. This loss minimizes the loss between the teacher and student and its objective is given by

$$\mathcal{L}_{kd} = \frac{1}{N} \sum_{i=1}^{N} ||G_{teacher}(z) - G_{student}(z)||^2. \tag{5}$$

Moreover, the objective of the student adapter modules is to fool the discriminator, minimizing the standard GAN loss

$$\mathcal{L}_{adv} = \frac{1}{N} \sum_{i=1}^{N} \log(1 - D(G_{student}(z))), \tag{6}$$

where D means discriminator network of the GANs. Lastly, the $\mathcal{L}_{kd}$ and the standard GAN loss are combined with the CDC loss to obtain more diversity. The final objective of the student is given by

$$\mathcal{L}_{student} = \mathcal{L}_{adv} + \alpha \mathcal{L}_{kd} + w_s \mathcal{L}_{\text{cdc}} \tag{7}$$

where α is the weight of the loss $\mathcal{L}_{kd}$ and w_s is the weight of the CDC loss for the student.

4 Experiments

Datasets. This section provides details of the performed experiments and shows the qualitative and quantitative results of the CFTS-GAN for continual few-shot image synthesis. The datasets taken into consideration for the experiments are sketches [22], female [10], sunglasses [22], male [10], and babies [22]. These datasets are used as the target datasets for evaluating the efficacy of our model for the few-shot continual generation of images. Each of these datasets consists of 10 samples. While the source model is pre-trained on a large FFHQ dataset [11].
Evaluation metrics. The state-of-the-art CDC [22], RSSA [41], DCL [45], CAM-GAN [35] and LFS-GAN [29] approaches are considered for comparison. The CAM-GAN model appends adapter modules on the top of [19] for continual image generation. The LFS-GAN adds more weights for the subsequent tasks using few-shot learning. Where LFS-GAN derives its model from a very strong model StyleGAN2 [12]. Moreover, it also utilizes the patch discriminator [22] to improve the target generation. While others compared state-of-the-art models are also based on a stronger model [12]. We evaluated our CFTS-GAN teacher-student model for the few-shot continual learning, which produces diverse images and has a quality near to the stronger LFS-GAN model. The Frchet inception distance (FID) [8] and B-LPIPS [29] are used as quantitative metrics. The FID score provides how much the synthesis images are close to the real images. The lower FID means that the generator produces samples closer to the real images. While the B-LPIPS shows the diversity of the images, the higher score shows a more diverse image generation.
Training. The source model is pre-trained on the source task using a large dataset using the FFHQ dataset. The FFHQ contains 70k high-resolution image

samples. The input image dimension fed into the model is 256×256. The other datasets mentioned above are considered as subsequent tasks. Therefore, the teacher and student model are trained to generate new tasks continually. For the training, only the data from the current task is available while the previous data is not available. The teacher model is trained on a few shot images. The knowledge is then transferred from the teacher to the student model using knowledge distillation while also maintaining the source diversity. So the teacher-student model correspondingly produces quality and diverse samples without affecting the previous knowledge.

4.1 Qualitative Results

The qualitative results are shown in Figure 2. Our model is able to generate different, diverse, and comparatively quality samples of images continually. The model produces the current data samples without affecting previously learned samples. A few generated samples from each task are presented in the paper.

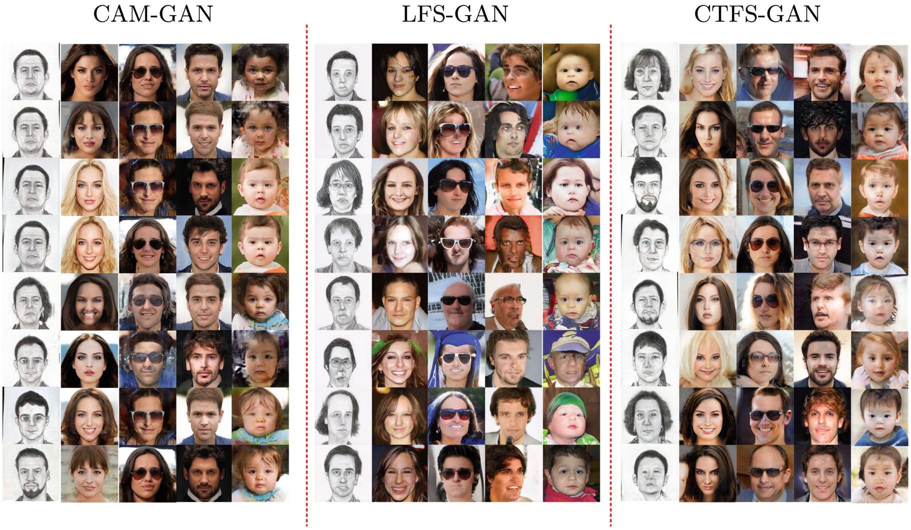


Fig. 2. Qualitative results: Generated samples. In each group, the first column is for sketches (Task 1), and the second, third, fourth, and fifth are for females (Task 2), sunglasses (Task 3), males (Task 4), and babies (Task 5).

4.2 Quantitative Results

The comparison in terms of FID and B-LPIPS scores with the state-of-the-art models is shown in Table 1. The bold value represents the best results while the underlined values represent the second-best results. The CFTS-GAN represents the best performance than CAM-GAN, CDC, RSSA, and DCL in terms of both B-LPIPS and FID, maintaining the same number of weights and same architecture on evaluation as CAM-GAN. Except for the FID of babies, the CDC is the second best. While compared with LFS-GAN the teacher-student model also gives the best performance in terms of B-LPIPS. In terms of FID, the LFS-GAN [29] performs best, probably because of utilizing a very strong baseline model StyleGAN2 [12].

Table 1. Quantitative results comparison in terms of FID (↓) and B–LPIPS (↑) for each task with state-of-the-art methods.

	Sketches (T_1)		**Female** (T_2)		**Sunglasses** (T_3)		**Male** (T_4)		**Babies** (T_5)		**Average**	
	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS
CDC [22]	322.72	0.205	197.40	0.427	244.94	0.463	277.00	0.381	69.98	0.454	208.41	0.386
RSSA [41]	308.00	0.285	175.20	0.440	207.58	0.484	205.49	0.405	76.70	0.481	194.59	0.419
DCL [45]	297.73	0.307	170.31	0.435	191.54	0.490	194.42	0.443	77.22	0.487	186.25	0.432
CAM-GAN [35]	91.81	0.293	85.68	0.332	86.81	0.333	82.83	0.312	146.20	0.181	98.66	0.290
LFS-GAN [29]	**34.66**	0.354	**29.59**	0.481	**27.69**	0.584	**35.44**	0.472	**41.48**	0.556	**33.77**	0.489
CFTS-GAN (ours)	82.49	**0.399**	62.10	**0.707**	36.03	**0.966**	66.23	**0.760**	96.62	**1.02**	68.69	**0.770**

4.3 Ablation Study

To show the effectiveness of the CFTS-GAN we analyzed each component of our proposed method. The details for each component are given below.

Effect of CDC on the teacher. We analyzed the effect of CDC loss using different values of w_t which are shown in Table 2. From the ablation, it is concluded that it increases diversity as we give more value to it. However, increasing w_t after some point leads to a higher FID score. So, in our experiments, we use the optimal value of $w_t = 40$ for training the teacher model for all of the tasks. It is because of gaining better FID at this point.

Table 2. Ablation study for the teacher using w_t for sunglasses dataset.

w_t	10	20	30	40	50	60	70
FID	79.41	63.70	46.94	**46.76**	54.21	63.39	65.76
B-LPIPS	0.245	0.516	0.856	0.930	0.943	0.956	**0.976**

Effect of CDC on the student. We also inspect the CDC loss terms and see its effect on the student model. Giving more weight to the CDC loss obtain more diversity and less weight leads to less diversity. However, increasing its effect leads to degrading the FID as we analyzed for the teacher. We performed some ablation studies on the sunglasses dataset for different values of w_s which is shown in Table 3. Assigning the $w_s = 20$ works better for gaining more diversity with better FID. So, for all the tasks, we used $w_s = 20$ for the student model.

Effect of $\mathcal{L}_{kd}$ on the student. The experiments are also performed by assigning different values to α. We observe that giving the value of 2 leads to better results as shown in Table 3. Giving more value to it has more impact on diversity. Assigning greater value to it leads to more diversity and light improvement in the FID. From this ablation study, we found when $\alpha = 2$, we have better FID. Therefore, we use the mentioned value for the rest of the tasks.

Table 3. Ablation study for the student on α and w_s for the sunglasses dataset.

	$\alpha = 0$				$\alpha = 2$				$\alpha = 5$				$\alpha = 10$			
w_s	10	20	30	40	10	20	30	40	10	20	30	40	10	20	30	40
FID	56.35	39.92	40.98	53.6	50.66	**37.54**	42.07	54.31	50.08	38.55	39.67	47.87	46.59	42.80	39.72	46.94
B-LPIPS	0.588	0.942	0.990	1.01	0.768	0.944	**1.02**	1.01	0.851	0.993	0.991	1.00	0.917	0.997	1.00	1.01

Freezing student discriminator. We also analyzed the student model by freezing the layers of the discriminator of our CFTS-GAN. We inspect that training the last 24 layers of the discriminator leads to better FID scores and more diverse image generation instead of training all the layers (total layers are 36) of the discriminator. The ablation study for the student model with a frozen discriminator is shown in Table 4. For this ablation study, we consider the teacher with the best value as given in Table 2. From this analysis, it is concluded that instead of training all the layers, if some portion of the discriminator is kept frozen it leads to better results.

Table 4. Ablation study for the student with freezing discriminator for sunglasses dataset.

Number of last trained layers	6	12	18	24	30	36
FID	51.22	39.20	38.98	**36.03**	40.38	37.54
B-LPIPS	0.902	0.910	0.922	0.966	**0.971**	0.944

5 Conclusion

This article proposes a continual learning few shots generative adversarial network CFTS-GAN. The CFTS-GAN considers the challenging tasks of catastrophic forgetting and overfitting problems in GANs. We used the teacher-student model for the challenging task of continual few shots image generation. For continual learning, the CFTS-GAN uses adapter modules as a student for learning the new task while preserving the previously learned knowledge.

The teacher model helps the student to produce better and more diverse image generation. The CDC is used by both the teacher and student to preserve the source diversity and prevent mode collapse. Moreover, we used a simple effective strategy of freezing the discriminator for more improvements. To show the performance of the CFTS-GAN model, it is analyzed on different datasets. Which shows better results and produces diverse images than the state-of-the-art models.

References

1. Abdollahzadeh, M., Malekzadeh, T., Teo, C.T., Chandrasegaran, K., Liu, G., Cheung, N.M.: A survey on generative modeling with limited data, few shots, and zero shot. arXiv preprint arXiv:2307.14397 (2023)
2. Abuduweili, A., Li, X., Shi, H., Xu, C.Z., Dou, D.: Adaptive consistency regularization for semi-supervised transfer learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6923–6932 (2021)
3. Aguinaldo, A., Chiang, P.Y., Gain, A., Patil, A., Pearson, K., Feizi, S.: Compressing gans using knowledge distillation. arXiv preprint arXiv:1902.00159 (2019)
4. Chen, P., Zhang, Y., Li, Z., Sun, L.: Few-shot incremental learning for label-to-image translation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3697–3707 (2022)
5. Chenshen, W., HERRANZ, L., Xialei, L., et al.: Memory replay GANs: Learning to generate images from new categories without forgetting [C]. In: The 32nd International Conference on Neural Information Processing Systems, Montréal, Canada. pp. 5966–5976 (2018)
6. Duan, Y., Niu, L., Hong, Y., Zhang, L.: Weditgan: Few-shot image generation via latent space relocation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 1653–1661 (2024)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
8. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
9. Israr, S.M., Zhao, F.: Customizing gan using few-shot sketches. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 2229–2238 (2022)
10. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
11. 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)
12. 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)
13. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proc. Natl. Acad. Sci. **114**(13), 3521–3526 (2017)

14. Kumar, A., Chatterjee, S., Rai, P.: Bayesian structural adaptation for continual learning. In: International Conference on Machine Learning. pp. 5850–5860. PMLR (2021)
15. Le, C.P., Dong, J., Aloui, A., Tarokh, V.: Mode-aware continual learning for conditional generative adversarial networks. arXiv preprint arXiv:2305.11400 (2023)
16. Lesort, T., Caselles-Dupré, H., Garcia-Ortiz, M., Stoian, A., Filliat, D.: Generative models from the perspective of continual learning. In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2019)
17. Li, X., Tang, B., Li, H.: Adaer: An adaptive experience replay approach for continual lifelong learning. Neurocomputing **572**, 127204 (2024)
18. Mallya, A., Davis, D., Lazebnik, S.: Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 67–82 (2018)
19. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: International conference on machine learning. pp. 3481–3490. PMLR (2018)
20. Mo, S., Cho, M., Shin, J.: Freeze the discriminator: a simple baseline for fine-tuning gans. arXiv preprint arXiv:2002.10964 (2020)
21. Noguchi, A., Harada, T.: Image generation from small datasets via batch statistics adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2750–2758 (2019)
22. Ojha, U., Li, Y., Lu, J., Efros, A.A., Lee, Y.J., Shechtman, E., Zhang, R.: 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)
23. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2009)
24. Park, K.H., Song, K., Park, G.M.: Pre-trained vision and language transformers are few-shot incremental learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23881–23890 (2024)
25. Rajasegaran, J., Hayat, M., Khan, S.H., Khan, F.S., Shao, L.: Random path selection for continual learning. Advances in Neural Information Processing Systems **32** (2019)
26. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Efficient parametrization of multi-domain deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8119–8127 (2018)
27. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
28. Seff, A., Beatson, A., Suo, D., Liu, H.: Continual learning in generative adversarial nets. arXiv preprint arXiv:1705.08395 (2017)
29. Seo, J., Kang, J.S., Park, G.M.: LFS-GAN: Lifelong Few-Shot Image Generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11356–11366 (2023)
30. Shi, J., Liu, W., Zhou, G., Zhou, Y.: Autoinfo gan: Toward a better image synthesis gan framework for high-fidelity few-shot datasets via nas and contrastive learning. Knowl.-Based Syst. **276**, 110757 (2023)
31. Song, X., Shu, K., Dong, S., Cheng, J., Wei, X., Gong, Y.: Overcoming catastrophic forgetting for multi-label class-incremental learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2389–2398 (2024)

32. Sushko, V., Wang, R., Gall, J.: Smoothness similarity regularization for few-shot gan adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7073–7082 (2023)
33. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12183–12192 (2020)
34. Tian, S., Li, L., Li, W., Ran, H., Ning, X., Tiwari, P.: A survey on few-shot class-incremental learning. Neural Netw. **169**, 307–324 (2024)
35. Varshney, S., Verma, V.K., Srijith, P., Carin, L., Rai, P.: Cam-gan: Continual adaptation modules for generative adversarial networks. Adv. Neural. Inf. Process. Syst. **34**, 15175–15187 (2021)
36. Vladymyrov, M., Zhmoginov, A., Sandler, M.: Few-shot incremental learning using hypertransformers (2022)
37. Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: theory, method and application. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
38. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: A survey on few-shot learning. ACM computing surveys (csur) **53**(3), 1–34 (2020)
39. Wang, Y., Wu, C., Herranz, L., Van de Weijer, J., Gonzalez-Garcia, A., Raducanu, B.: Transferring gans: generating images from limited data. In: Proceedings of the European conference on computer vision (ECCV). pp. 218–234 (2018)
40. Wang, Z., Jiang, Y., Zheng, H., Wang, P., He, P., Wang, Z., Chen, W., Zhou, M., et al.: Patch diffusion: Faster and more data-efficient training of diffusion models. Advances in Neural Information Processing Systems **36** (2024)
41. 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)
42. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3014–3023 (2021)
43. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. arXiv preprint arXiv:1708.01547 (2017)
44. Zhai, M., Chen, L., Tung, F., He, J., Nawhal, M., Mori, G.: Lifelong gan: Continual learning for conditional image generation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2759–2768 (2019)
45. Zhao, Y., Ding, H., Huang, H., Cheung, N.M.: A closer look at few-shot image generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9140–9150 (2022)
46. Zhou, D.W., Wang, F.Y., Ye, H.J., Ma, L., Pu, S., Zhan, D.C.: Forward compatible few-shot class-incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9046–9056 (2022)

T2R-GAN: A CGAN-based model for rural thematic road extraction

Zixiang Ni[1] and Weixin Zhai[1,2](✉)

[1] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
zhaiweixin@cau.edu.cn

[2] Key Laboratory of Agricultural Machinery Monitoring and Big Data Application, Ministry of Agriculture and Rural Affairs, Beijing 100083, China

Abstract. Rural roads extracted from agricultural machinery trajectories have significant research value. Due to the intricate network of rural roads and the large difference in the density of agricultural machinery trajectories, traditional road extraction methods struggle to perform effectively on rural roads with complex topology and on agricultural machinery trajectories with obscure road features. Therefore, this paper proposes a CGAN-based model named T2R-GAN (Trajectory to Road-GAN) for rural thematic road extraction, which learns the trajectory-to-road feature mapping through continuous adversarial training between the ELAU-Net generator and the PatchGAN discriminator to adapt to trajectories of various densities. ELAU-Net is an efficient network that utilizes an encoder-decoder structure and ELA modules to enhance the capture of obscure road features between sparse trajectories. To enhance model performance and reduce the risk of overfitting, bilateral hinge loss is designed to train our model to enhances the discriminator's discriminative ability to facilitate the generator to generate more realistic roads improve the generalization of the model. To verify the effectiveness of T2R-GAN in extracting roads from agricultural trajectory data, this paper selects the real agricultural trajectory data from Henan Province, China in June 2021 as the dataset for verification. The experimental results show that the proposed method achieves 79.23% $F1_{score}$, which is 5.53% higher than the previous state-of-art. The proposed T2R-GAN provides a novel and effective approach for extracting rural thematic roads from agricultural machinery trajectories for the first time.

Keywords: Rural thematic based road extraction · Agricultural machinery trajectory · Conditional Generative Adversarial Network · ELAU-Net

1 Introduction

Agricultural machinery trajectories refer to a series of geospatial coordinates recorded by the Global Navigation Satellite System (GNSS) terminals installed

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 263–276, 2025.
https://doi.org/10.1007/978-3-031-78389-0_18

on agricultural machines. These trajectories encompass a wealth of road-related information, indicating the paths and routes taken by agricultural machinery [1]. Since most of the movement area of agricultural machinery is in rural areas, rural thematic road extraction refers to extracting road routes from agricultural machinery trajectories. Rural thematic roads extracted using agricultural machinery trajectories can not only be applied in the fields of geographic information systems and precision agriculture [2–4], but also can build a road network in rural areas to fill the gaps in rural maps from a geographical point of view, which is of great significance in research [5,6].

Extracting roads from agricultural machinery trajectories is highly challenging, and traditional road extraction methods, including remote sensing image-based approaches and trajectory-based approaches, struggle to yield positive results. Since the roads in rural areas are typically low-quality dirt roads that are narrow and often obscured by forest canopy complicating their representation in remote sensing images. This obstruction hampers the effectiveness of traditional remote sensing image-based road extraction methods [7,8]. Moreover, since the quality of agricultural machines carrying GNSS terminals is mostly variable, the data sampling frequency is relatively low, and the density difference between trajectories is large. The traditional trajectory-based road extraction method [9–11], which makes extensive use of semantic segmentation models and is oriented towards urban vehicle trajectories with high sampling frequency and high trajectory density, is facing a major challenge on the road extraction in the context of agricultural machinery trajectories.

To the best of our knowledge, there is no road extraction method for agricultural machinery trajectories, which motivates us to develop an efficient road extraction method for agricultural machinery trajectories. To overcome the varios density and obscure road features of agricultural machinery trajectories, we base our proposed model on the CGAN and propose T2R-GAN for rural thematic based road extraction from agricultural machinery trajectories. Firstly, T2R-GAN generates a fake road image infinitely close to the real road through adversarial training between the generator and the discriminator within the model, and constrains the model with the trajectory image so that the model is able to capture the complex feature mapping between the trajectory and the road. Secondly, an ELAU-Net is designed as the generator of T2R-GAN, which accurately generates road lines through an encoder-decoder structure and ELA module for more efficient feature extraction. And the PatchGAN is proposed as the discriminator to improve the model's attention to details. Thirdly, we design bilateral hinge loss as the loss function of our model to improve the performance of the discriminator and reduce the overfitting risk of the model on small volume datasets. The main contributions of our work are summarized as follows.

- A novel image generation model T2R-GAN is proposed, which overcome the low density and obscure road features of agricultural machinery trajectories by generating realistic road images through learning trajectory-to-road feature mappings through adversarial training between the designed ELAU-Net

generator and the PatchGAN discriminator. It is the first road extraction model for agricultural machinery trajectories.
- An ELAU-Net is proposed, which fuses different levels of semantic information through encoder-decoder structure and ELA module to accurately extract roads. The novel ELA module connecting the encoder and decoder can help the network overcome the noise in the low-level semantics in the trajectories and enhance the capture of road features between sparse trajectories.
- Bilateral hinge loss is designed to train our model to enhance the discriminator and reduce the overfitting risk of the model.
- We use real agricultural machinery trajectories collected in June 2021 in Henan Province for our experiments. Experimental results show that the proposed method can automatically effectively extract road from agricultural machinery trajectories.

2 Related Work

The existing related research work on road extraction can be classified into artificial approaches, remote sensing image-based approaches and trajectory-based approaches. Artificial approaches refers to extract roads by human experts based on visual interpretation to extract roads from a large number of trajectories or remotely sensed images, which will consume a lot of human and material resources and is less efficient [12,13].

Remote sensing image-based approaches refers to extract roads from remotely sensed images based on topological, geometric, textural and other features of roads by machine learning or deep learning models [14]. Yu et al. designed MSAU-Net, which uses U-Net with a multi-attention mechanism to achieve accurate extraction of roads in high resolution remote sensing images, and Candy operator is used to extract the edge features of roads, which makes the extracted roads smoother [7]. Xu et al. designed a road extraction model named IDANet, which bases on D-LinkNet with an attention mechanism and iterative training to improve the accuracy of extracted roads [8]. Remote sensing image-based approaches are widely used, but the method faces challenges in rural road extraction. Because the rural roads through which agricultural machinery passes are usually low-quality and narrow dirt roads, and are easily obscured by forest canopy, making their representation in remote sensing imagery very complex.

Trajectory-based approaches convert trajectory points into grid points, transforming the trajectory into a raster image, and then extract road information from the raster image. This transforms trajectory-based road extraction into a problem of generating road images based on trajectory raster images. Deep learning models in image translation are then used to extract the road information. Ruan et al. proposed the DeepMG method for generating roads using vehicle trajectories, which uses cab trajectories and urban road network data as training datasets, and then uses a deep convolutional network named T2RNet to extract the centerline of roads [9]; Eftelioglu et al. designed a GPS trajectory-based road extraction model named RING-Net to cope with the problem of drastic GPS trajectory noise by applying a spatial self-attention layer to the bottom layer of the network, which makes the method robust on different trajectory data [10]. However, most of the existing methods are oriented towards vehicle trajectories with high trajectory density and clearly visible road features, which are suitable for the semantic segmentation model widely used by the above methods. While for agricultural machinery trajectories with low trajectory density and uneven sampling frequency, these methods face challenges.

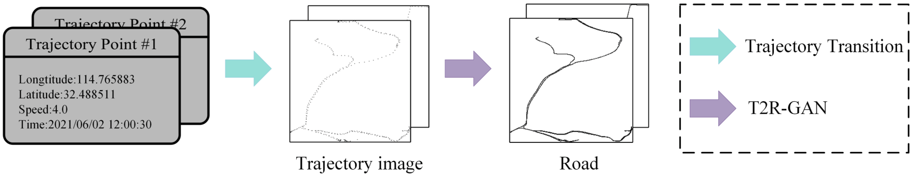


Fig. 1. The overall workflow of the road extraction method based on agricultural machinery trajectories. The trajectory transition part includes mapping the trajectory points to grid points, transforming the trajectory into a raster image, and obtaining the dataset for model training through data augmentation. Then T2R-GAN is built to train on the dataset for road extraction.

The road extraction method used in this paper is similar to trajectory-based approaches, and the overall workflow is shown in Fig. 1. In order to make the model able to extract road information from complex agricultural machinery trajectories, we construct the road extraction model T2R-GAN based on the powerful image generation model CGAN. The proposed model has excellent performance on trajectories with different densities and is capable of road extraction based on agricultural machinery trajectories.

3 Methodology

3.1 Overview of T2R-GAN

T2R-GAN is an image generation model that excels in extracting roads with complex trajectories based on the CGAN architecture [15]. We proposed an ELAU-Net as its generator and an PatchGAN as its discriminator, as shown in Fig. 2. ELAU-Net is an encoder-decoder network with ELA modules. PatchGAN is a efficient discriminator. Bilateral hinge loss is designed to train the two subnetworks.

3.2 ELAU-Net Generator and PatchGAN Discriminator

Our ELAU-Net is a novel encoder-decoder network that connects encoders and decoders with advanced ELA modules [16] to efficiently transfer road information at different levels, thus improving model performance. Fig. 3 illustrates the

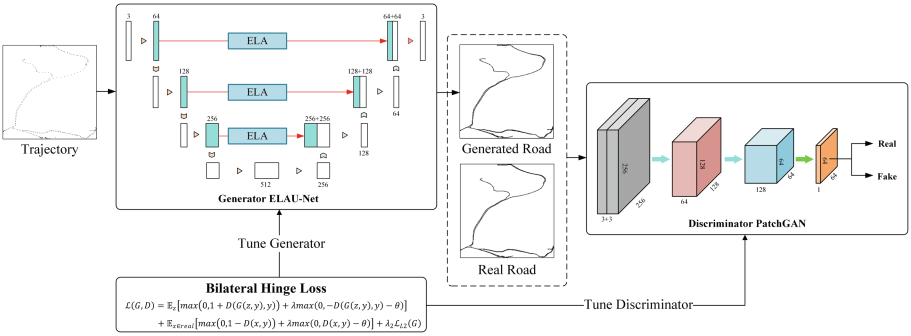


Fig. 2. The structure of the T2R-GAN. It consists of a generator and a discriminator, the generator is the ELAU-Net and the discriminator is the PatchGAN. The generator and the discriminator will conduct adversarial training according to the bilateral hinge loss.

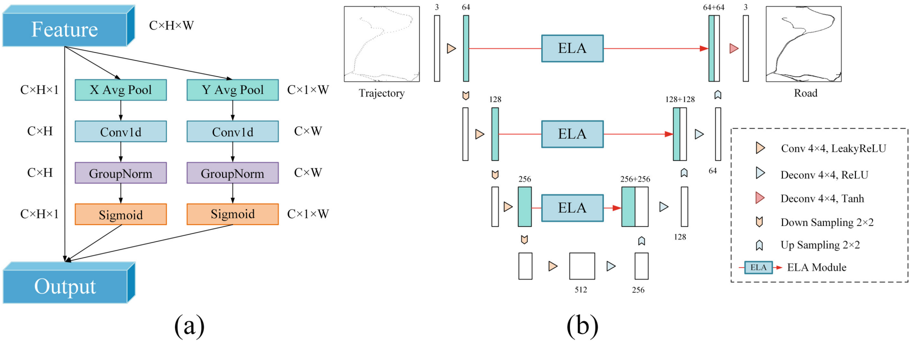


Fig. 3. The structure of ELA module (a) and the structure of ELAU-Net (b).

structure of the proposed model. ELAU-Net comprises two fundamental modules: an encoder (on the left side of Fig. 3) and a decoder (on the right side of Fig. 3). The encoder applies convolutional operations, batch normalization, and LeakyReLU activation functions for feature extraction, while the decoder utilizes deconvolution operations, batch normalization, and ReLU activation functions to generate images layer by layer. The ELA module between corresponding layers of the encoder and decoder amalgamate feature maps, feeding them into the subsequent layer to extract contextual information.

Although skip-connections in traditional U-Net [17] can effectively convey low-level semantic information of images, they also convey a lot of noise affecting the quality of generated roads. To overcome these noises, we replace skip-connections with the novel ELA (Efficient Local Attention) module, which is able to fully maintain the low-level semantic features of the trajectory images, is good at capturing the road features between long-distance trajectories, and overcomes the effects of noise in the trajectories. The left side of Fig. 3 demonstrates the structure of ELA. ELA uses strip pooling [18] in the spatial dimension to obtain feature vectors in the horizontal and vertical directions. Then ELA applies 1D convolution to locally interact with the two feature vectors separately, and the obtained feature vectors are processed by group normalization [19] and nonlinear activation functions to produce positional attention predictions in both directions. The final output features are obtained by multiplying the positional attention in both directions with the input features. The output of a convolutional block is denoted as $\mathbb{R}^{H\times W\times C}$, where H, W, and C represent the height, width, and channel dimension, respectively. Strip pooling averages each channel over two spatial scales: horizontally $(H, 1)$ and vertically $(1, W)$. The output of the cth channel at height h and the cth channel at width w is represented by the following mathematical equation.

$$z_c^h(h) = \frac{1}{H} \sum_{0\le i<H} x_c(h, i) \tag{1}$$

$$z_c^w(w) = \frac{1}{W} \sum_{0\le j<W} x_c(j, w) \tag{2}$$

z_h and z_w not only capture the global sensory fields but also contain precise positional information. We apply 1D convolutions F_h and F_w to enhance the position information in the horizontal and vertical directions. Subsequently, the group normalization will be used to process the augmented position information to obtain the positional attention representation in both horizontal and vertical directions:

$$y^h = \sigma\left(G_n\left(F_h\left(z_h\right)\right)\right) \tag{3}$$

$$y^w = \sigma\left(G_n\left(F_w\left(z_w\right)\right)\right) \tag{4}$$

where σ denotes the nonlinear activation function, here sigmoid. the convolution kernel size for F_h and F_w is set to 7, and the number of groups for the 1D convolution is set to $in\_channels/8$. The final output features are obtained by multiplying the positional attentions in both directions:

$$Y \ = \ x_c \times y^h \times y^w. \tag{5}$$

ELA maintains a narrow kernel shape by strip pooling in the horizontal and vertical directions to capture dependencies between long-distance trajectories and filters out noise in extraneous regions to produce rich target location features in the respective directions. 1D convolution preserves channel dimensionality and reduces model complexity, and group normalization helps to improve the generalization performance of the model compared to batch normalization. This enables ELAU-Net to efficiently integrate different levels of road information to generate more realistic roads.

The discriminator is responsible for checking whether the road maps generated by the generator are realistic or not, and in this way promoting the generator to generate more realistic road maps. PatchGAN is a commonly used discriminator network that enhances the detailed quality of generated images through local discrimination. Fig. 4 illustrates the structure of PatchGAN, which consists of a convolutional neural network comprising several convolution modules. It performs multiple convolutions on the input image to generate feature maps, and then evaluates the reality or falsity of each element in the feature maps, taking the arithmetic average as the probability that the input image is considered true or false. PatchGAN's local discrimination helps the network to focus on details such as road connectivity, prompting the network to generate detailed and realistic roads.

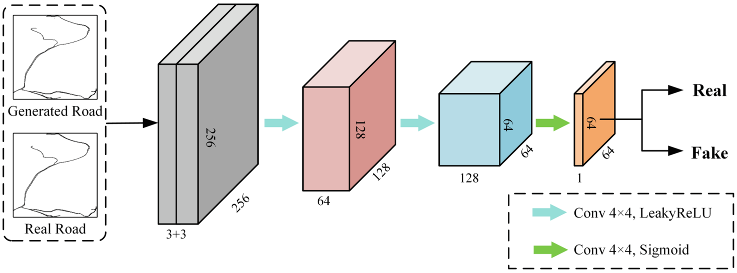


Fig. 4. The structure of PatchGAN.

3.3 Bilateral Hinge Loss

Bilateral hinge loss is an improvement of traditional hinge loss. The traditional hinge loss [20] hopes that the output of the discriminator D for the real data x is greater than 1 as much as possible, while the output of the generated data $G(z, y)$ is less than -1 as much as possible, i.e., it penalizes the case that the absolute value of the output is too small. In this study, to reduce the risk of overfitting brought by PatchGAN, bilateral hinge loss is designed to also penalize the case where the absolute value of the output is too large, and the specific formula is as follows:

$$\mathcal{L}\left(D_{\text{real}}\right) = \max(0, 1 - D(x, y)) + \lambda \max(0, D(x, y) - \theta) \tag{6}$$

$$\mathcal{L}\left(D_{\text{fake}}\right) = \max(0, 1 + D(G(z, y), y)) + \lambda \max(0, -D(G(z, y), y) - \theta) \tag{7}$$

Where x represents the real data, i.e., the real road image, y represents the condition variable, i.e., the trajectory image, z represents the noise, and D_{real} and D_{fake} represent the outputs of the discriminator for the real data x and for the generated data $G(z, y)$, respectively, subject to the constraints of y. λ and θ are hyperparameters.

During the training process of our T2R-GAN, the generator continually learns the distribution of real road images to create realistic new ones, while the discriminator simultaneously aims to distinguish between the generated road images and real road images. These two components engage in an adversarial train, continuously adjusting until they reach a dynamic equilibrium. In this state, the generated road images closely resemble the distribution of real road images, making them nearly indistinguishable from real ones by the discriminator, and T2R-GAN is able to learn how to generate realistic roads based on trajectories. Combine bilateral hinge loss and add the $L2$ paradigm for regularization, the loss function during this adversarial train is as follows:

$$\begin{aligned}\mathcal{L}\left(G, D\right) = \mathbb{E}_{z}\left[\max(0, 1 + D(G(z, y), y)) + \lambda \max(0, -D(G(z, y), y) - \theta)\right] \\ + \mathbb{E}_{x \in \text{real}}\left[\max(0, 1 - D(x, y)) + \lambda \max(0, D(x, y) - \theta)\right] + \lambda_2 \mathcal{L}_{L2}\left(G\right)\end{aligned} \tag{8}$$

where λ, θ, and λ_2 are hyperparameters set to 0.5, 2 and 10. Constraining PatchGAN by using bilateral hinge loss can effectively improve the model performance, enabling T2R-GAN to efficiently capture key road features to generate more realistic roads, while avoiding overfitting caused by its excessive learning.

4 Experiment

4.1 Experimental Setting

Dataset. The trajectory points used in this paper are from the agricultural machinery trajectories dataset collected in Henan Province, China in June 2021. Since the operation area of agricultural machinery contains fields, the trajectory points of agricultural machinery are divided into field points and road points [21–25], and the road points can be used for road extraction. Among the 140 data samples acquired, the total number of road points is 117,489. Mapping these trajectory points into 2D space by trajectory rasterization [26], we obtain 140 trajectory images. These samples were labeled to obtain 140 trajectory-road image pairs. After data augmentation such as rotation and mirror rotation, as demonstrated in Fig. 5, the image dataset capacity is expanded from 140 to 1120. Then, the dataset will be randomly divided into train set and test set for model training in the ratio of 9:1. In addition, we selected 788 centralized data samples collected in Nanyang City, Henan Province, China in June 2021 as validation examples for road network construction in Sec 5. Finally, the capacity of the train set and test set for model training is 980 and 140, respectively, and the capacity of the validation examples for road network construction is 788.

Experimental Environment. The model in this study is implemented using PyTorch 2.0.0, and the programming environment is Python 3.8. All models were trained and tested on an NVIDIA RTX 4090 with 24GB of memory. During the model training process, we employed the Adam optimizer with an initial learning rate of 8e-5 to update the parameters of both the generator and the discriminator [27].

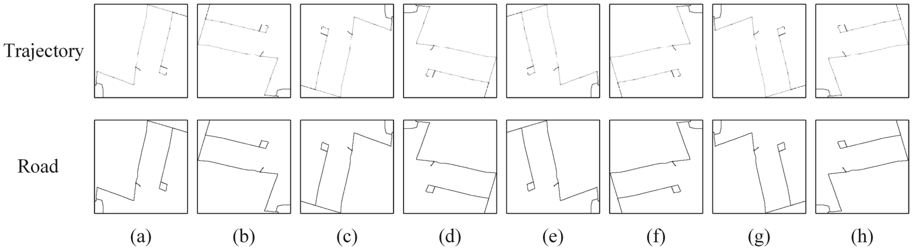


Fig. 5. Visualization of data augmentation. From left to right: (a) Initial, (b) Rotation by 90° (counterclockwise direction), (c) Rotation by 180°, (d) Rotation by 270°, (e) Mirror (left and right), (f) Mirror Rotation by 90°, (g) Mirror Rotation by 180°, (h) Mirror Rotation by 270°.

Evaluation metrics. To evaluate the accuracy of extracted roads, this study establishes a 3-metre buffer around the target roads and classifies the roads generated by the model into three categories: true positives (TP) within the buffer zone, false positives (FP) outside the buffer zone, and false negatives (FN) not generated by the model. Three metrics are derived to assess the accuracy of the extracted roads. $Precision = \frac{TP}{TP+FP}$ represents the ratio of the length of true roads to the total length of constructed roads, $Recall = \frac{TP}{TP+FN}$ represents the ratio of the length of true roads to the total length of target roads, and $F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$ represents the overall similarity between the constructed road and the target road, with higher values indicating that the constructed road is more realistic. Moreover, this study uses the PCC (Pearson Correlation Coefficient [28]) metric for multivariate analysis of the quality of the generated roads, which is widely used in the task of image generation to assess the similarity between the generated image and the real image in terms of details and structure. In this study, the metric is used to assess the quality of generated roads including details such as road connectivity. The definition of PCC is specified as follows:

$$PCC = \frac{\sum_{x=1}^{H} \sum_{y=1}^{W} (I_{xy} - \bar{I})(I'_{xy} - \bar{I}')}{\sqrt{\sum_{x=1}^{H} \sum_{y=1}^{W} (I_{xy} - \bar{I})^2 \sum_{x=1}^{H} \sum_{y=1}^{W} (I'_{xy} - \bar{I}')^2}} \tag{9}$$

where I_{xy} and I'_{xy} prime denote the pixel values of the target road image and the generated road image at point (x, y), respectively.

Baseline. T2RNet, SPBAM-LinkNet, and AD-LinkNet [9,29,30] are state-of-the-art semantic segmentation models used to extract roads from trajectories. Morever, to verify the efficiency and generalization performance of bilateral hinge loss, ELAU-Net paired with traditional CGAN loss named ELA-GAN will also participate in the comparison experiment.

5 Results

We use the validation example introduced in subsection 4.1 for road network construction to test the road extraction effect of T2R-GAN. Fig. 6 shows the superimposed effect diagram of the rural roads extracted by the model and the real remote sensing impact. As can be seen from the figure, the roads extracted by the T2R-GAN construct can reflect the rural roads better, which is practical to a certain extent. Some of the discrete lines in the figure are caused by field points that were not split cleanly. With the optimization of the field and road segmentation algorithm, the road points obtained by segmentation will be cleaner and the above problems can be solved.

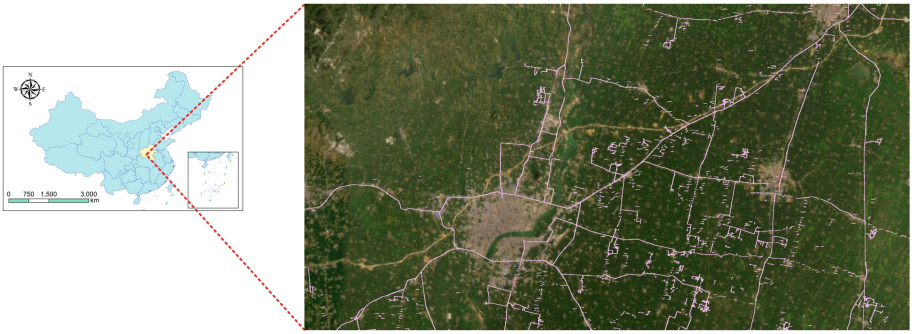


Fig. 6. Road extraction effects in rural areas of Nanyang City, Henan Province. The pink highlighted lines in the figure ind0icate the roads extracted based on agricultural machinery trajectories.

We evaluate T2R-GAN compared with baseline models on the test set introduced in subsection 4.1. All models aretrained for 500 epochs with pre-trained weights. The test results of different models are shown in Table 1. From the table, it can be seen that compared to T2RNet, SPBAM-LinkNet, AD-LinkNet, and ELA-GAN, our proposed T2R-GAN achieved the highest precision of 83.61%, the highest recall of 75.29%, the highest PCC of 70.26%, and the highest F1-Score of 79.23%, which shows that the roads extracted by our proposed model are the most realistic. The F1-Score is 5.53% higher than the previous state-of-art AD-LinkNet. It proves that our proposed model is stronger than the semantic segmentation model widely used for vehicle trajectories on the agricultural machinery trajectories dataset. Although ELA-GAN achieves a high recall, its precision is lower than AD-LinkNet and T2R-GAN, which indicates that although ELA-GAN generates roads that cover the real roads relatively well, it also generates many redundant roads. This is due to the fact that ELA-GAN undergoes overfitting and recognizes trajectory points that are on different roads as being on the same road, so many redundant roads are generated. Whereas, T2R-GAN has the highest precision, which shows that bilateral hinge loss can effectively prevent overfitting.

Table 1. Comparison results on the test set of different models

Model	Precision	Recall	F1-Score	PCC
T2RNet	65.97%	61.59%	63.71%	52.68%
SPBAM-LinkNet	74.24%	67.28%	70.59%	54.23%
AD-LinkNet	76.79%	70.85%	73.70%	57.09%
ELA-GAN	76.00%	72.02%	73.96%	62.48%
T2R-GAN	**83.61%**	**75.29%**	**79.23%**	**70.26%**

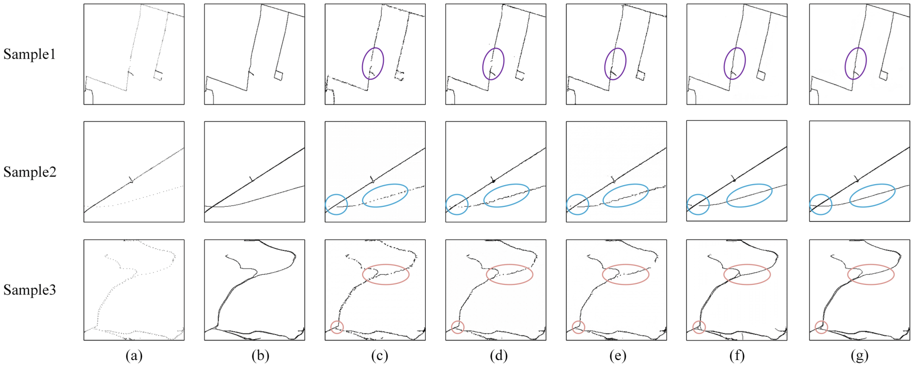


Fig. 7. Comparison of road quality extracted by different models. From left to right: (a) Trajectory, (b) Real road, (c) T2RNet, (d) SPBAM-LinkNet, (e) AD-LinkNet, (f) ELA-GAN, (g) T2R-GAN.

Fig. 7 illustrates the visual comparison of road quality extracted by different models, further demonstrating the superiority of T2R-GAN. For samples with high trajectory density (partial straight roads in Sample 1 and Sample 2), all five models can extract roads effectively. However, in sparse trajectory regions (marked by circles in Sample 1, Sample 2, and Sample 3), only ELA-GAN and T2R-GAN are able to extract roads. This indicates that the road reconstruction performance of these three state-of-the-art semantic segmentation models in the sparse trajectory regions is weaker than that of our CGAN-based model. Also, the visual comparison shows that ELA-GAN tends to produce redundant roads. For instance, at the road intersection marked in Sample 3, ELA-GAN erroneously generates non-existent roads, leading to a reduction in model precision. Furthermore, ELA-GAN performs poorly at the intersection of dense and sparse trajectories, as seen at the road intersection marked on the left side of Sample 2, where it fails to extract the road where sparse trajectories are present. In contrast, T2R-GAN demonstrates superior performance in handling trajectories of varying densities, successfully extracting roads in sparse trajectory regions, and achieving the best road connectivity. Meanwhile, T2R-GAN has a lower risk of overfitting compared to ELA-GAN. The probability of generating redundant roads by T2R-GAN is significantly reduced. This highlights the efficiency of T2R-GAN in addressing trajectories of different densities at intersections, showcasing its effectiveness.

6 Conclusion

We propose an image generation model named T2R-GAN to extract rural thematic roads from agricultural machinery trajectories. This model utilizes adversarial training between the ELAU-Net generator and the PatchGAN discriminator to extract roads. An ELAU-Net is proposed as the generator, which efficiently generates road images based on trajectory images, overcoming the low density of agricultural machinery trajectories and obscure road features. Bilateral hinge loss is designed to further enhance the discriminative capabilities of PatchGAN discriminator and improve the generalization performance of the model. We constructed a dataset of trajectory-road image pairs based on real agricultural machinery trajectories and evaluated T2R-GAN on this dataset. Experimental results demonstrate that our model can extract roads from agricultural machinery trajectories, outperforming other state-of-art road extraction models in terms of precision, completeness, generalization, and overall performance.

Acknowledgements. This research was financially supported by the National Natural Science Foundation of China (32301691), National Key R&D Program of China and Shandong Province, China (2021YFB3901300) and the National Precision Agriculture Application Project (Grant/Contract number: JZNYYY001).

References

1. Wu, C., Li, D., Zhang, X., et al.: Application note: China's agricultural machinery operation big data system[J]. Comput. Electron. Agric. **205**, 107594 (2023)
2. Molari, G., Mattetti, M., Lenzini, N., et al.: An updated methodology to analyse the idling of agricultural tractors[J]. Biosys. Eng. **187**, 160–170 (2019)
3. Sopegno, A., Calvo, A., Berruto, R., et al.: A web mobile application for agricultural machinery cost analysis[J]. Comput. Electron. Agric. **130**, 158–168 (2016)
4. Pagare, V., Nandi, S., Khare, D.K.: Appraisal of Optimum Economic Life for Farm Tractor: A Case Study[J]. Econ. Aff. **64**(1), 117–124 (2019)
5. Ang, K.L.M., Seng, J.K.P.: Big data and machine learning with hyperspectral information in agriculture[J]. IEEE Access **9**, 36699–36718 (2021)
6. Li, D., Zheng, Y., Zhao, W.: Fault analysis system for agricultural machinery based on big data[J]. Ieee Access **7**, 99136–99151 (2019)
7. Pan, D., Zhang, M., Zhang, B.: A generic FCN-based approach for the road-network extraction from VHR remote sensing images-using OpenStreetMap as benchmarks[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **14**, 2662–2673 (2021)
8. Xu B, Bao S, Zheng L, et al. IDANet: Iterative D-LinkNets with attention for road extraction from high-resolution satellite imagery[C]//Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Cham: Springer International Publishing, 2021: 140-152

9. Ruan S, Long C, Bao J, et al. Learning to generate maps from trajectories[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(01): 890-897
10. Eftelioglu E, Garg R, Kango V, et al. RING-Net: Road inference from gps trajectories using a deep segmentation network[C]//Proceedings of the 10th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data. 2022: 17-26
11. Lu, C., Sun, Q., Zhao, Y., et al.: A Road Extraction Method Based on Conditional Generative Adversarial Nets[J]. Geomatics and Information Science of Wuhan University **46**(6), 807–815 (2021)
12. Karimi, H.A., Kasemsuppakorn, P.: Pedestrian network map generation approaches and recommendation. Int. J. Geograph. Inf. Sci. **27**(5), 947–962 (2013)
13. Wei, Y., Tinghua, A.I.: Road centerline extraction from crowdsourcing trajectory data[J]. Geography and Geo-Information Science **32**(3), 1–7 (2016)
14. Lian, R., Wang, W., Mustafa, N., et al.: Road extraction methods in high-resolution remote sensing images: A comprehensive review[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **13**, 5489–5507 (2020)
15. Isola P, Zhu J Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1125-1134
16. Xu W, Wan Y. ELA: Efficient Local Attention for Deep Convolutional Neural Networks[J]. arxiv preprint arxiv:2403.01123, 2024
17. Ronneberger, O., Fischer, P., Brox, T., U-net: Convolutional networks for biomedical image segmentation[C], , Medical image computing and computer-assisted intervention-MICCAI,: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18. Springer International Publishing **2015**, 234–241 (2015)
18. Hou Q, Zhang L, Cheng M M, et al. Strip pooling: Rethinking spatial pooling for scene parsing[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 4003-4012
19. Wu Y, He K. Group normalization[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19
20. George E, Murray M, Swartworth W, et al. Training shallow ReLU networks on noisy data using hinge loss: when do we overfit and is it benign?[J]. Advances in Neural Information Processing Systems, 2024, 36
21. Zhai, W., Mo, G., Xiao, Y., et al.: GAN-BiLSTM network for field-road classification on imbalanced GNSS recordings[J]. Comput. Electron. Agric. **216**, 108457 (2024)
22. Xiao, Y., Mo, G., Xiong, X., et al.: DR-XGBoost: An XGBoost model for field-road segmentation based on dual feature extraction and recursive feature elimination[J]. International Journal of Agricultural and Biological Engineering **16**(3), 169–179 (2023)
23. Zhai, W., Xiong, X., Mo, G., et al.: A Bagging-SVM field-road trajectory classification model based on feature enhancement[J]. Comput. Electron. Agric. **217**, 108635 (2024)
24. Chen Y, Quan L, Zhang X, et al. Field-road classification for GNSS recordings of agricultural machinery using pixel-level visual features[J]. 2023
25. Chen, Y., Li, G., Zhang, X., et al.: Identifying field and road modes of agricultural Machinery based on GNSS Recordings: A graph convolutional neural network approach[J]. Comput. Electron. Agric. **198**, 107082 (2022)

26. Gengeç, N.E., Tarı, E., Performance evaluation of gps trajectory rasterization methods[C], , Computational Science and Its Applications-ICCSA,: 21st International Conference, Cagliari, Italy, September 13–16, 2021, Proceedings, Part I 21. Springer International Publishing **2021**, 3–17 (2021)
27. Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arxiv preprint arxiv:1412.6980, 2014
28. Neto A M, Victorino A C, Fantoni I, et al. Image processing using Pearson's correlation coefficient: Applications on autonomous robotics[C]//2013 13th International Conference on Autonomous Robot Systems. IEEE, 2013: 1-6
29. Liu, Z., He, J., Zhang, C., et al.: Vehicle trajectory extraction at the exit areas of urban freeways based on a novel composite algorithms framework[J]. Journal of Intelligent Transportation Systems **27**(3), 295–313 (2023)
30. Yang, X., Fan, X., Su, Y., et al.: TR2RM: an urban road network generation model based on multisource big data[J]. International Journal of Digital Earth **17**(1), 2344596 (2024)

d-Sketch: Improving Visual Fidelity of Sketch-to-Image Translation with Pretrained Latent Diffusion Models without Retraining

Prasun Roy[1](✉), Saumik Bhattacharya[2], Subhankar Ghosh[1], Umapada Pal[3], and Michael Blumenstein[1]

[1] University of Technology Sydney, Ultimo, NSW 2007, Australia
{prasun.roy,subhankar.ghosh}@student.uts.edu.au,
michael.blumenstein@uts.edu.au

[2] Indian Institute of Technology, Kharagpur 721302, WB, India
saumik@ece.iitkgp.ac.in

[3] Indian Statistical Institute, Kolkata 700108, WB, India
umapada@isical.ac.in

Abstract. Structural guidance in an image-to-image translation allows intricate control over the shapes of synthesized images. Generating high-quality realistic images from user-specified rough hand-drawn sketches is one such task that aims to impose a structural constraint on the conditional generation process. While the premise is intriguing for numerous use cases of content creation and academic research, the problem becomes fundamentally challenging due to substantial ambiguities in freehand sketches. Furthermore, balancing the trade-off between shape consistency and realistic generation contributes to additional complexity in the process. Existing approaches based on Generative Adversarial Networks (GANs) generally utilize conditional GANs or GAN inversions, often requiring application-specific data and optimization objectives. The recent introduction of Denoising Diffusion Probabilistic Models (DDPMs) achieves a generational leap for low-level visual attributes in general image synthesis. However, directly retraining a large-scale diffusion model on a domain-specific subtask is often extremely difficult due to demanding computation costs and insufficient data. In this paper, we introduce a technique for sketch-to-image translation by exploiting the feature generalization capabilities of a large-scale diffusion model without retraining. In particular, we use a learnable lightweight mapping network to achieve latent feature translation from source to target domain. Experimental results demonstrate that the proposed method outperforms the existing techniques in qualitative and quantitative benchmarks, allowing high-resolution realistic image synthesis from rough hand-drawn sketches.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78389-0_19.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 277–292, 2025.
https://doi.org/10.1007/978-3-031-78389-0_19

Keywords: Sketch-to-Image translation · Latent diffusion models

1 Introduction

Freehand sketches provide simple and intuitive visual representations of natural images, allowing humans to understand and envision complex objects with a few sparse strokes. The convenience of modifying such minimalistic stroke-based representations to conceptualize semantic image manipulation is one key motivation for researchers to explore sketch-to-image translation. There are two primary objectives for such conditional generation – the synthesized image should be *visually realistic* and *structurally consistent* with the input sketch, enabling perceptually appealing image generation from hand-drawn sketches irrespective of the artistic expertise of users. However, the intriguing premise becomes substantially challenging due to the practically unavoidable ambiguities in freehand sketches. For example, sketches of a specific object drawn by different persons can widely differ in stroke density and structural adherence depending on artistic abilities, as illustrated in Fig. 1 with samples from the Sketchy dataset [43].

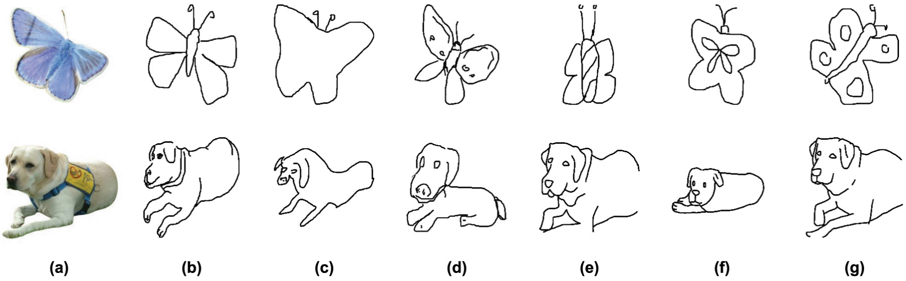


Fig. 1. Structural ambiguity in hand-drawn sketches. **(a)** Subject image. **(b)**–**(g)** Freehand sketches drawn by different users. The examples are from the Sketchy dataset [43].

Consequently, this problem forces the generative algorithms to balance the trade-off between visual realism and intended shape. Existing GAN-based [19,31] methods primarily address sketch-to-image translation in two ways – a direct mapping between domains with conditional GANs [10,11,17,18,22,25,26,28,54,62] or modification in latent space using GAN inversions [3,61]. However, such techniques often require application-specific data, optimization objectives, and complex learning strategies but occasionally fail to produce stable outcomes. Additionally, these methods operate on limited sets of task-specific object classes, resulting in poor generalization for unseen categories.

More recently, denoising diffusion probabilistic models [16,21,33,46] have demonstrated unprecedented improvements in the perceptual quality of general image synthesis. With sufficiently large annotated datasets [44,45], text-conditioned diffusion models [38–40,42] have achieved state-of-the-art results across multiple vision tasks, such as image generation, super-resolution, and inpainting. However, due to high structural ambiguities in hand-drawn sketches and the lack of sufficient paired sketch-image data, large-scale diffusion models have seen limited success in sketch-to-image translation. Furthermore, training such architectures from scratch is often computationally demanding and heavily infrastructure-dependent, which limits the scope of adopting the rich generative capabilities of latent diffusion models into sketch-to-image translation.

In this paper, we propose a novel method for photorealistic image generation from freehand sketches leveraging the learned feature space of a pre-trained latent diffusion model [40]. We achieve this by introducing a learnable lightweight feature mapping network to perform latent code translation between source (*sketch*) and target (*image*) domains. The proposed approach provides a more stable optimization than GANs without requiring to train the latent diffusion model, thus mitigating the instability of GANs and the high computational overhead of large-scale diffusion models. Furthermore, unlike the existing methods, the proposed technique generalizes well beyond task-specific data distribution, significantly improving the generative performance on unseen object categories.

Contributions: The main contributions of the proposed work are as follows.

1. We introduce an efficient method of photorealistic image generation from freehand sketches by providing structural guidance to a pre-trained latent diffusion model without retraining.
2. The proposed approach achieves significantly better generalization beyond the observed data distribution, outperforming existing task-specific methods.

The remainder of the paper is organized as follows. Sec. 2 provides a brief overview of existing sketch-to-image translation techniques. Sec. 3 discusses the background and technical details of the proposed method, followed by the experimental analyses in Sec. 4. We conclude the paper by summarizing our findings and discussing the potential scopes in Sec. 5.

2 Related Work

Conditional GANs: Image-to-Image translation using conditional GANs is a widely explored method of directly transforming freehand sketches into images. Early architectural improvements introduced a Markovian discriminator [22] for better retention of high-frequency correctness in paired image-to-image translation. A subsequent approach [62] extended the idea to unpaired data by enforcing cycle consistency between source and target domains. In [54], the authors used coarse-to-fine generators, multi-scale discriminators, and an additional feature-matching loss for generating higher-resolution images. In [34], the authors achieved generational improvements in semantic image manipulation by introducing spatially-adaptive normalization. The initial work exclusively on multi-class sketch-to-image translation proposed a masked residual unit [11], accommodating fifty object categories. Another approach proposed a contextual GAN [28] to learn the joint distribution of the sketch and corresponding image.

Researchers also explored interactive generation [18] using a gating mechanism to suggest the probable completion of a partial sketch, followed by rendering the final image with a pre-trained image-to-image translation model [54]. In [17], the authors proposed a multi-stage class-conditioned approach for object-level and scene-level image synthesis from freehand sketches, improving the perceptual baseline over direct generations [22], contextual networks [28], and methods based on scene graphs [4,23] or layouts [58]. In [52], the authors achieved similar goals with an unsupervised approach by introducing a standardization module and disentangled representation learning.

GAN inversions: The main objective of GAN inversion is to find a latent embedding of an image such that the original image can be faithfully reconstructed from the latent code using a pre-trained generator. Existing strategies for such inversions can be learning-based [3,6,35,61], optimization-based [1,2,14,27,29,37,51], or hybrid [7,60]. In a learning-based inversion, an encoder learns to project an image into the latent space, minimizing reconstruction loss between the decoded (reconstructed) and original images. An optimization-based inversion estimates the latent code by directly solving an objective function. In a hybrid approach, an encoder first learns the latent projection, followed by an optimization strategy to refine the latent code. The rich statistical information captured by deep generative networks from large-scale data provides effective *priors* for various downstream tasks, including sketch-to-image translation. In [3], the authors adopted a learning-based GAN inversion strategy using a multi-class deep generative network [8], pre-trained on the large-scale ImageNet dataset [15], as *prior* to achieve sketch-to-image translation for multiple categories. In [55], the authors introduced a framework for generalizing image synthesis to *open-domain* object categories by jointly learning two *in-domain* mappings (image-to-sketch and sketch-to-image) with *random-mixed* strategy.

Diffusion models: A Denoising Diffusion Probabilistic Model (DDPM) [21,46] is a parameterized Markov chain that learns to generate samples similar to the original data distribution after a finite time. In particular, DDPMs use variational inference to learn to iteratively reverse a stepwise *diffusion* (noising) process. In [47], the authors introduced Denoising Diffusion Implicit Models (DDIMs) by generalizing DDPMs using non-Markovian diffusion processes with the same learning objective, leading to a deterministic and faster generative process. Recent advances [16,33] have shown that diffusion models can achieve generational improvements in the visual quality and sampling diversity over GANs while providing a more stable and straightforward optimization objective. The most prolific application of diffusion models in recent literature is text-conditioned image generation [38–40,42] and modification [5,9,20,32], utilizing a pre-trained language-image model [36] to embed the conditioning prompt. In [13], the authors guided the generative process with an iterative latent variable refinement to produce high-quality variations of a reference image. In [41], the authors introduced a class-specific prior preservation loss to finetune an existing text-to-image diffusion model for *personalized* manipulation of a specific subject image from a few observations. Emerging alternative approaches also involved

Stochastic Differential Equations (SDEs) to guide the generative process following score-based [30] or energy-based [56,59] objectives. More recent attempts for sketch-to-image translation involved multiple objectives [53], multi-dimensional control [12], or latent code optimization [50]. In [53], the authors used an additional network to reconstruct the input sketch from the generated image. The denoising process was optimized using a cumulative objective function consisting of the *perceptual similarity* (between the input and reconstructed sketches) and *cosine similarity* (between the input and generated images) measures. In [12], the authors provided three-dimensional controls over image synthesis from the strokes and sketches to manipulate the balance between *perceptual realism* and *structural faithfulness* during the conditional denoising process. In [50], the authors introduced a lightweight mapping network for providing structural guidance to a pre-trained latent diffusion model [40]. While the method avoided training a dedicated diffusion network, the *differential guidance* made sampling images computationally even more demanding than a large-scale model itself.

3 Method

3.1 Preliminaries

Diffusion models: A Denoising Diffusion Probabilistic Model (DDPM) defines a Markov chain that learns to generate samples to match the input data distribution over a finite time. The process consists of *forward diffusion* that iteratively perturbs an input by adding noise according to a scheduler, followed by *backward denoising* that learns to reverse the mapping to recover the original input from noise. Given a data distribution $x_0 \sim q(x_0)$, the *forward diffusion* defines an iterative noising process q that adds Gaussian noise over T finite steps, gradually perturbing the input sample x_0 to produce latents $\{x_1, ..., x_T\}$ as follows.

$$q(x_1, ..., x_T|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{1}$$

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}\; x_{t-1}, \beta_t \mathbf{I}) \tag{2}$$

where $\beta_t \in (0,1)$ denotes the variance of the Gaussian noise at time $t \sim [1,T]$. Rewriting Eq. 2 with $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, Ho *et al.* [21] deduced a closed-form expression to sample an arbitrary step of the noising process, directly estimating x_t from x_0 as the following marginal distribution.

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}\; x_0, (1-\overline{\alpha}_t)\mathbf{I}) \tag{3}$$

With sufficiently large T and a well-defined schedule of β_t, the latent x_T closely resembles a Gaussian distribution. If the reverse distribution $q(x_{t-1}|x_t)$ is known, sampling $x_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively running the process in reverse can yield a sample from $q(x_0)$. However, as $q(x_{t-1}|x_t)$ depends on the entire data distribution, the *backward denoising* process can be approximated by a learnable network, parameterized with θ, as follows.

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{4}$$

Ho *et al.* [21] also observed that learning to predict the added noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ worked best for estimating x_0 with the following formulation.

$$x_0 = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \overline{\alpha}_t}} \, \epsilon \right) \tag{5}$$

Most implementations adopt a U-Net architecture (parameterized with θ) to predict the added noise, minimizing mean squared error as the learning objective.

$$\mathcal{L}_{DM} = \mathbb{E}_{t \sim [1,T],\ x_0 \sim q(x_0),\ \epsilon \sim \mathcal{N}(0,\mathbf{I})} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right] \tag{6}$$

3.2 Latent Code Translation Network (LCTN)

We propose a learnable Latent Code Translation Network (LCTN) to shift the input latent space toward the target domain by exploiting the learned feature representations of a pre-trained Latent Diffusion Model (LDM) [40]. LCTN is trained with edge maps [49] instead of hand-drawn sketches to mitigate the structural ambiguities that arise from freehand sketches. Our experiments show that LCTN trained on edge maps works appreciably well during inference with freehand sketches. Given an image x, corresponding edge map e, and object class name c, we use the pre-trained image encoder $\mathcal{E}$ and text encoder $\mathcal{T}$ of LDM to compute the initial latent codes as, $\overline{x} = \mathcal{E}(x)$, $\overline{e} = \mathcal{E}(e)$, and $\overline{c} = \mathcal{T}(c)$. The input feature space F is computed from the intermediate activation maps of LDM U-Net ϵ_θ, rescaled to have the same spatial dimensions, with a single denoising pass of $\overline{e}$ at timestep $t = 0$ using $\overline{c}$ as conditioning, $F = f_{\epsilon_\theta}(\overline{e}, \overline{c}, t)$. LCTN learns to project F into the target latent code z_0 by minimizing the mean squared error, $\mathcal{L}_{LCTN} = \|z_0 - \overline{x}\|^2$. Architecturally, LCTN consists of a sequence of fully connected (FC) layers with 512, 256, 128, and 64 nodes, with each FC layer followed by ReLU activation and batch normalization. We illustrate the proposed training strategy for LCTN in Fig. 2.

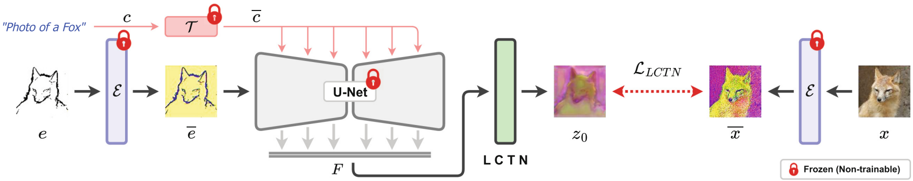


Fig. 2. Proposed training strategy for the Latent Code Translation Network (LCTN).

Ideally, if the domain translation by LCTN is accurate, we can readily decode z_0 into a high-quality photorealistic image using the pre-trained LDM decoder

$\mathcal{D}$. However, due to high sparsity in the input edge maps (or sketches), LCTN-projected latent code lacks sufficient subtlety, leading to unrealistic images from direct decoding. We address the issue by first perturbing z_0 to z_k over $k \sim [1, T]$ steps, where $1 < k < T$, followed by T denoising iterations to get $\overline{z}_0$ from z_k. With a sufficiently large value of k, z_k is close to an isotropic Gaussian distribution, $z_k \approx z_T \sim \mathcal{N}(0, \mathbf{I})$. However, strictly enforcing $k < T$ ensures minimal structural elements are retained in z_k. We observe that starting the backward denoising from z_k instead of z_T as the initial latent, followed by decoding the final latent $\overline{z}_0$, can produce photorealistic images while retaining the intended structural resemblance with the input edge map (or sketch). In our experiments, $0.7 \leqslant \frac{k}{T} \leqslant 0.9$ works best for most cases. We illustrate the proposed sampling strategy for LCTN in Fig. 3.

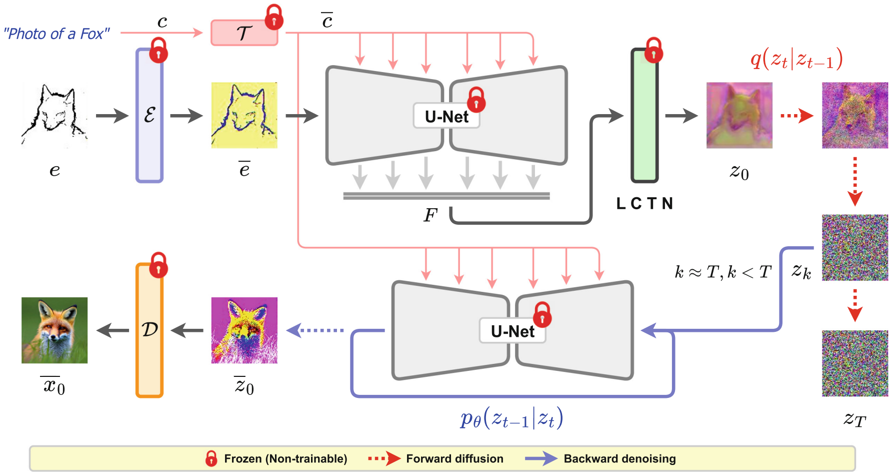


Fig. 3. Proposed sampling strategy for the Latent Code Translation Network (LCTN).

4 Experiments

Datasets: We evaluate the performance of the proposed method against existing sketch-to-image translation techniques [22,50,55,62] on three following datasets. **(a) Scribble:** The Scribble dataset [18] contains 256×256 image-sketch pairs of ten object classes (basketball, chicken, cookie, cupcake, moon, orange, pineapple, soccer, strawberry, and watermelon) having uniform white backgrounds. While the images in the dataset do not feature complex backgrounds, 60% of the object classes share nearly identical circular shapes, which introduces significant ambiguities to the generative algorithms. We use 1512 image-sketch pairs [55] (1412 train + 100 test) to train and evaluate all competing methods.

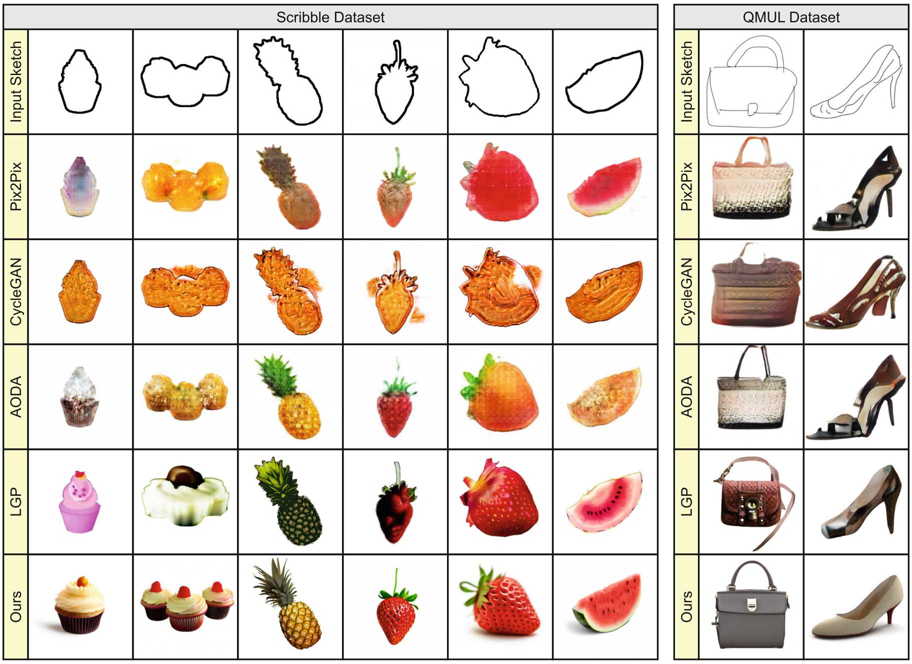


Fig. 4. Qualitative comparison of the proposed method with existing sketch-to-image translation techniques – Pix2Pix [22], CycleGAN [62], AODA [55], and LGP [50] on Scribble [18] and QMUL [48,57] datasets.

(b) QMUL: The QMUL dataset is a compilation [55] of image-sketch pairs from three object categories – shoe [57], chair [57], and handbag [48] with uniform white backgrounds. Due to the structural ambiguities in the provided hand-drawn sketches, the dataset poses a substantial challenge to the generative algorithms. Following [55], we use 7850 freehand sketches of 3004 images for training and 691 freehand sketches of 480 images for evaluation.
(c) Flickr20: While Scribble [18] and QMUL [48,57] datasets provide significant structural challenges to the learning algorithms, the images do not contain perceptual complexities of natural backgrounds. To investigate the generative performances in such cases, we introduce a new dataset by collecting 10K (9500 train + 500 test) high-resolution images from Flickr, equally distributed over 20 animal classes – bird, cat, cow, deer, dog, dolphin, elephant, fox, frog, giraffe, goat, horse, lion, monkey, pig, polar bear, rabbit, sheep, tiger, and zebra. The edge maps for these images are estimated with a pre-trained edge detector [49].
Implementation and experimental details: The LCTN architecture consists of a sequence of four fully connected (FC) hidden layers having 512, 256, 128, and 64 nodes, with each FC layer followed by ReLU activation and batch normalization. A final FC layer projects the last hidden layer output to a 4D latent vector, representing a single spatial position in the 4-channel latent space

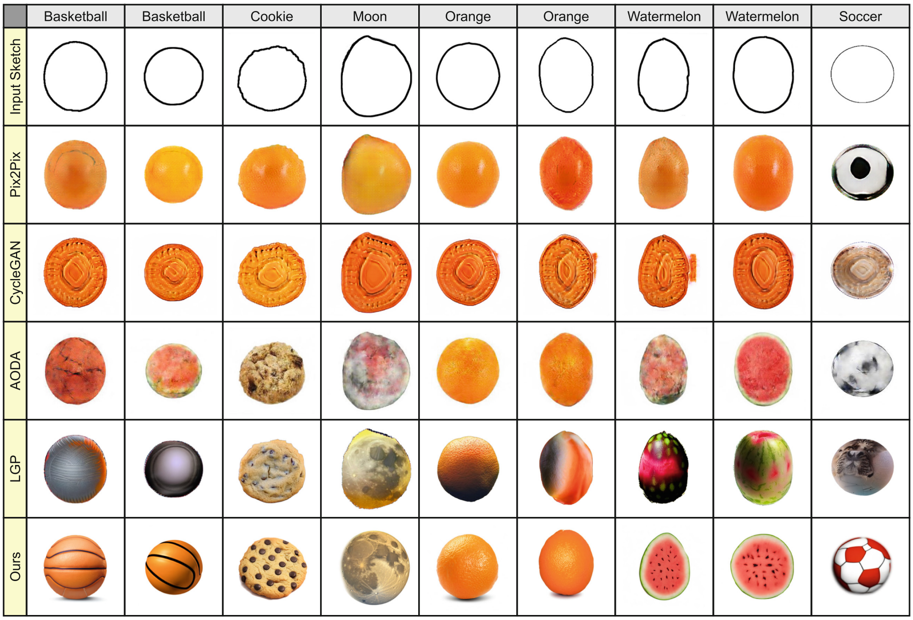


Fig. 5. Qualitative comparison for distinct object classes with nearly identical shapes. The proposed method can produce high-quality, visually distinguishable objects in contrast to the ambiguous results generated by existing sketch-to-image translation techniques – Pix2Pix [22], CycleGAN [62], AODA [55], and LGP [50].

z_0. We use the *Stable Diffusion v2.1* (SD2.1) distribution for pre-trained text encoder, VAE and U-Net. LCTN is trained for 50000 iterations at a constant learning rate of 0.001 with 100 initial warm up steps on a single NVIDIA Quadro RTX 6000 GPU with a batch size of 4 and FP16 mixed precision. We keep the default image size of SD2.1 (768 × 768) throughout all our experiments. LCTN is initialized with a normal distribution $\mathcal{N}(0, 0.02)$. We optimize the parameters of LCTN using stochastic Adam optimizer [24] having β-coefficients (0.9, 0.999). For reproducibility, the code is officially available at https://github.com/prasunroy/dsketch. We have included the full-resolution visual results in the ***supplementary material***.

Visual analysis: For analyzing the perceptual quality of the generated images by our method, we perform a visual comparison with existing GAN-based [22,55,62] and diffusion-based [50] sketch-to-image translation techniques. Fig. 4 demonstrates a qualitative comparison of the proposed method against Pix2Pix [22], CycleGAN [62], AODA [55], and LGP [50] on Scribble [18] and QMUL [48,57] datasets. Our method can generate highly detailed and perceptually appealing samples that are visibly superior to existing approaches while maintaining the intended structural resemblance with the input sketches.

Visual analysis on ambiguous classes: Occasionally, multiple visually distinguishable objects can have identical shapes. For example, 60% object classes in the Scribble dataset [18] have an identical circular structure (basketball, cookie, moon, orange, soccer, and watermelon), leading to nearly indistinguishable sketches for visibly distinguishable object categories. Therefore, it poses a substantial challenge to the generative algorithms for producing class-conditioned distinctive visual features in such ambiguous cases. Fig. 5 shows a qualitative comparison of the proposed method against existing approaches [22,50,55,62] on ambiguous classes from the Scribble dataset [18]. Pix2Pix [22] and CycleGAN [62] mostly fail to produce distinguishable objects. AODA [55] and LGP [50] achieve limited success in producing photorealistic results. In contrast, our method can generate high-quality and visibly distinctive images with class-specific visual attributes of intended objects from virtually identical sketches.

Evaluation metrics: We measure seven metrics to quantitatively evaluate the perceptual quality, structural consistency, and class accuracy in the generated images. Fréchet Inception Distance (**FID**) measures the feature space similarity between real and generated images. Inception Score (**IS**) estimates the Kullback-Leibler (KL) divergence between the label and marginal distributions to measure

Table 1. Quantitative analysis of the proposed method on Scribble [18] dataset.

Method	FID ↓	IS ↑	PSNR ↑	SSIM ↑	LPIPS ↓	ACC ↑	MOS ↑
Pix2Pix [22]	333.1872	3.8027	13.3208	0.6082	0.3635	0.24	0.02
CycleGAN [62]	322.6855	3.6737	13.4177	0.5804	0.3003	0.33	0.01
AODA [55]	353.9626	4.0133	12.4880	0.5588	0.3761	0.19	0.01
LGP [50]	207.8677	8.4247	5.6862	0.3171	0.5667	0.72	0.24
Ours	**163.8978**	**9.9132**	**13.8737**	**0.6406**	**0.2839**	**0.75**	**0.72**

Table 2. Quantitative analysis of the proposed method on QMUL [48,57] dataset.

Method	FID ↓	IS ↑	PSNR ↑	SSIM ↑	LPIPS ↓	ACC ↑	MOS ↑
Pix2Pix [22]	189.7064	**5.3261**	9.2383	0.5328	0.4013	0.6151	0.04
CycleGAN [62]	146.3326	5.1030	9.5792	0.6050	0.3198	0.4486	0.01
AODA [55]	216.7982	5.0196	9.8943	0.5784	0.4152	0.6208	0.01
LGP [50]	108.1720	5.1159	5.4842	0.1710	0.6943	0.8770	0.35
Ours	**63.9208**	4.3687	**11.8780**	**0.6677**	**0.3126**	**0.9899**	**0.59**

Table 3. Quantitative analysis of the proposed method on Flickr20 dataset.

Method	FID ↓	IS ↑	PSNR ↑	SSIM ↑	LPIPS ↓	ACC ↑	MOS ↑
Pix2Pix [22]	122.4473	8.5337	10.1246	0.1553	0.7136	0.430	0.02
CycleGAN [62]	162.6837	6.6324	10.6105	0.1261	0.7848	0.242	0.00
AODA [55]	150.0852	7.4056	10.0145	0.1478	0.7325	0.332	0.01
LGP [50]	81.4195	14.9779	9.3839	0.1109	0.7553	0.794	0.42
Ours	**72.5475**	**15.7383**	**10.9339**	**0.2113**	**0.6811**	**0.876**	**0.55**

the visual quality and class diversity of generated images. Peak Signal-to-Noise Ratio (**PSNR**) assesses the quality of generated images by estimating the deviation from real images. Structural Similarity Index Measure (**SSIM**) estimates the structural consistency in the generated images against the ground truth by considering image degradation as the perceived change in structural information. Learned Perceptual Image Patch Similarity (**LPIPS**) quantifies the perceptual similarity between real and generated images using the spatial feature maps obtained from a pre-trained deep convolutional network such as SqueezeNet in our experiments. We also estimate the classification accuracy (**ACC**) using a multi-class image classifier to measure the correctness of intended object classes in generated samples.

Human evaluation: Although the said metrics are widely used in the literature, perceptual quality assessment is an open challenge in computer vision. Therefore, we conducted an opinion-based user assessment among 45 individuals, where the volunteers were asked to select the most visually realistic sample that had the closest resemblance to a given sketch from a pool of images generated by the competing methods. The Mean Opinion Score (**MOS**) is the average fraction of times a method received user preference over other methods. Tables 1, 2, and 3 summarize the evaluation scores of different methods on the Scribble [18], QMUL [48,57], and Flickr20 datasets, respectively. In most cases, the proposed method achieves a better score than the existing sketch-to-image translation techniques [22,50,55,62] across different datasets, indicating superior perceptual quality, structural consistency, and class accuracy in the generated images.

Analyzing the optimal value of k: In the proposed method, $k \sim [1, T]$ is a crucial control parameter for balancing the trade-off between structural consistency and visual realism in the generated samples. As discussed in Sec. 3.2, directly decoding the LCTN-projected latent z_0 through the image decoder $\mathcal{D}$ produces virtually unusable images $\mathcal{D}(z_0)$. For substantially lower values of k, the generated image $\overline{x}_0$ retains high structural accuracy but lacks photorealism. With increasing values of k, perceptual quality of $\overline{x}_0$ gradually improves at the expense of structural consistency. While the optimal value of k varies among different datasets, $0.7 \leqslant \frac{k}{T} \leqslant 0.9$ works best for most cases in our experiments. Fig. 6 illustrates a visual analysis of balancing the trade-off between structural consistency and photorealism by selecting an optimal value of $k \approx T$, $k < T$.

Visual attribute control in the generated images: One key advantage of the proposed method is the ability to control visual attributes in the generated images for general image editing and manipulation. As the architecture does not require retraining the LDM, we can use the pre-trained LDM as a learned prior for visual modifications alongside LCTN to impose structural constraints. Fig. 7 shows a few examples where we render a specific object in multiple visual styles by providing different text prompts to the pre-trained LDM while keeping a consistent shape across different styles as intended in the input sketch.

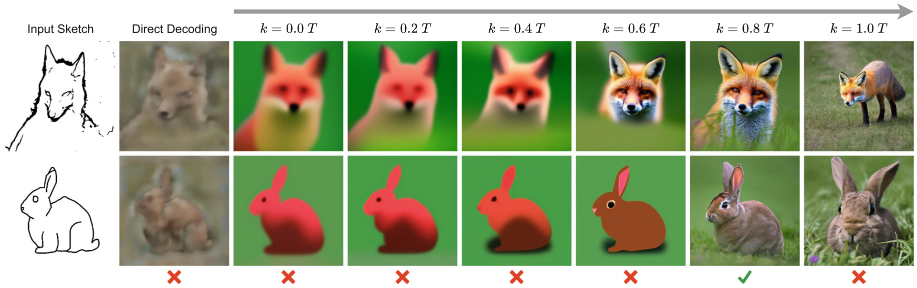


Fig. 6. Visual analysis of balancing the trade-off between structural consistency and perceptual quality by selecting an optimal value of k on the proposed Flickr20 dataset.

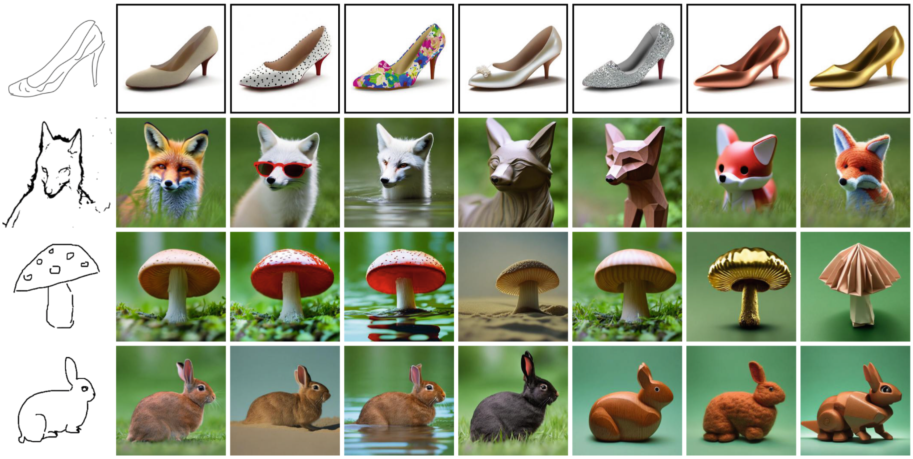


Fig. 7. Visual attribute control in the proposed sketch-to-image translation method. Training and sampling can exclusively use freehand sketches (**first row**) or edge maps (**second row**). Alternatively, training can be performed on edge maps while sampling uses freehand sketches of unseen (**third row**) or known (**fourth row**) object classes.

5 Conclusions

In this paper, we introduce a novel sketch-to-image translation technique that uses a learnable lightweight mapping network (LCTN) for latent code translation from sketch to image domain, followed by k forward diffusion and T backward denoising steps through a pre-trained text-to-image LDM. We show that by selecting an optimal value for $k \sim [1, T]$ near the upper threshold ($k \approx T$, $k < T$), it is possible to generate highly detailed photorealistic images that closely resemble the intended structures in the given sketches. Our experiments demonstrate that the proposed technique outperforms the existing methods in most visual and analytical comparisons across multiple datasets. Additionally, we show that the proposed method retains structural consistency across different visual styles, allowing photorealistic style manipulation in the generated images.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2StyleGAN: How to embed images into the StyleGAN latent space? In: The IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
2. Abdal, R., Qin, Y., Wonka, P.: Image2StyleGAN++: How to edit the embedded images? In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
3. An, Z., Yu, J., Liu, R., Wang, C., Yu, Q.: SketchInverter: Multi-class sketch-based image generation via GAN inversion. In: The IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2023)
4. Ashual, O., Wolf, L.: Specifying object attributes and relations in interactive scene generation. In: The IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
5. Bar-Tal, O., Ofri-Amar, D., Fridman, R., Kasten, Y., Dekel, T.: Text2LIVE: Text-driven layered image and video editing. In: The European Conference on Computer Vision (ECCV) (2022)
6. Bau, D., Zhu, J.Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., Torralba, A.: Inverting layers of a large generator. In: The International Conference on Learning Representations (ICLR) Workshop (2019)
7. Bau, D., Zhu, J.Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., Torralba, A.: Seeing what a GAN cannot generate. In: The IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
8. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: The International Conference on Learning Representations (ICLR) (2019)
9. Brooks, T., Holynski, A., Efros, A.A.: InstructPix2Pix: Learning to follow image editing instructions. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
10. Chen, S.Y., Su, W., Gao, L., Xia, S., Fu, H.: DeepFaceDrawing: Deep generation of face images from sketches. ACM Transactions on Graphics (TOG) (2020)
11. Chen, W., Hays, J.: SketchyGAN: Towards diverse and realistic sketch to image synthesis. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
12. Cheng, S.I., Chen, Y.J., Chiu, W.C., Tseng, H.Y., Lee, H.Y.: Adaptively-realistic image generation from stroke and sketch with diffusion model. In: The IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2023)
13. Choi, J., Kim, S., Jeong, Y., Gwon, Y., Yoon, S.: ILVR: Conditioning method for denoising diffusion probabilistic models. In: The IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
14. Creswell, A., Bharath, A.A.: Inverting the generator of a generative adversarial network. IEEE Transactions on Neural Networks and Learning Systems (TNNLS) (2018)
15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2009)

16. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: The Conference on Neural Information Processing Systems (NeurIPS) (2021)
17. Gao, C., Liu, Q., Xu, Q., Wang, L., Liu, J., Zou, C.: SketchyCOCO: Image generation from freehand scene sketches. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
18. Ghosh, A., Zhang, R., Dokania, P.K., Wang, O., Efros, A.A., Torr, P.H., Shechtman, E.: Interactive sketch & fill: Multiclass sketch-to-image translation. In: The IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: The Conference on Neural Information Processing Systems (NeurIPS) (2014)
20. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-or, D.: Prompt-to-Prompt image editing with cross-attention control. In: The International Conference on Learning Representations (ICLR) (2023)
21. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: The Conference on Neural Information Processing Systems (NeurIPS) (2020)
22. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-Image translation with conditional adversarial networks. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
23. Johnson, J., Gupta, A., Fei-Fei, L.: Image generation from scene graphs. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: The International Conference on Learning Representations (ICLR) (2015)
25. Li, Y., Chen, X., Wu, F., Zha, Z.J.: LinesToFacePhoto: Face photo generation from lines with conditional self-attention generative adversarial networks. In: The ACM International Conference on Multimedia (MM) (2019)
26. Li, Y., Chen, X., Yang, B., Chen, Z., Cheng, Z., Zha, Z.J.: DeepFacePencil: Creating face images from freehand sketches. In: The ACM International Conference on Multimedia (MM) (2020)
27. Lipton, Z.C., Tripathi, S.: Precise recovery of latent vectors from generative adversarial networks. In: The International Conference on Learning Representations (ICLR) Workshop (2017)
28. Lu, Y., Wu, S., Tai, Y.W., Tang, C.K.: Image generation from sketch constraint using contextual GAN. In: The European Conference on Computer Vision (ECCV) (2018)
29. Ma, F., Ayaz, U., Karaman, S.: Invertibility of convolutional generative networks from partial measurements. In: The Conference on Neural Information Processing Systems (NeurIPS) (2018)
30. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: The International Conference on Learning Representations (ICLR) (2021)
31. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
32. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
33. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: The International Conference on Machine Learning (ICML) (2021)

34. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
35. Perarnau, G., Van De Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional GANs for image editing. In: The Conference on Neural Information Processing Systems (NeurIPS) Workshop (2016)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: The International Conference on Machine Learning (ICML) (2021)
37. Ramesh, A., Choi, Y., LeCun, Y.: A spectral regularizer for unsupervised disentanglement. In: The International Conference on Machine Learning (ICML) (2019)
38. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125 (2022)
39. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: The International Conference on Machine Learning (ICML) (2021)
40. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
41. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
42. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. In: The Conference on Neural Information Processing Systems (NeurIPS) (2022)
43. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The Sketchy database: Learning to retrieve badly drawn bunnies. ACM Transactions on Graphics (TOG) (2016)
44. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: LAION-5B: An open large-scale dataset for training next generation image-text models. In: The Conference on Neural Information Processing Systems (NeurIPS) (2022)
45. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021)
46. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: The International Conference on Machine Learning (ICML) (2015)
47. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: The International Conference on Learning Representations (ICLR) (2021)
48. Song, J., Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: The IEEE/CVF International Conference on Computer Vision (ICCV) (2017)
49. Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q., Pietikäinen, M., Liu, L.: Pixel difference networks for efficient edge detection. In: The IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
50. Voynov, A., Aberman, K., Cohen-Or, D.: Sketch-guided text-to-image diffusion models. In: The ACM SIGGRAPH Conference Proceedings (2023)

51. Voynov, A., Babenko, A.: Unsupervised discovery of interpretable directions in the GAN latent space. In: The International Conference on Machine Learning (ICML) (2020)
52. Wang, J., Jeon, S., Yu, S.X., Zhang, X., Arora, H., Lou, Y.: Unsupervised scene sketch to photo synthesis. In: The European Conference on Computer Vision (ECCV) (2022)
53. Wang, Q., Kong, D., Lin, F., Qi, Y.: DiffSketching: Sketch control image synthesis with diffusion models. In: The British Machine Vision Conference (BMVC) (2022)
54. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
55. Xiang, X., Liu, D., Yang, X., Zhu, Y., Shen, X., Allebach, J.P.: Adversarial open domain adaptation for sketch-to-photo synthesis. In: The IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2022)
56. Xing, X., Wang, C., Zhou, H., Hu, Z., Li, C., Xu, D., Yu, Q.: Inversion-by-Inversion: Exemplar-based sketch-to-photo synthesis via stochastic differential equations. arXiv preprint arXiv:2308.07665 (2023)
57. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
58. Zhao, B., Meng, L., Yin, W., Sigal, L.: Image generation from layout. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
59. Zhao, M., Bao, F., Li, C., Zhu, J.: EGSDE: Unpaired image-to-image translation via energy-guided stochastic differential equations. In: Advances in Neural Information Processing Systems (2022)
60. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain GAN inversion for real image editing. In: The European Conference on Computer Vision (ECCV) (2020)
61. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: The European Conference on Computer Vision (ECCV) (2016)
62. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: The IEEE/CVF International Conference on Computer Vision (ICCV) (2017)

Semantically Consistent Person Image Generation

Prasun Roy[1](✉), Saumik Bhattacharya[2], Subhankar Ghosh[1], Umapada Pal[3], and Michael Blumenstein[1]

[1] University of Technology Sydney, Ultimo, NSW 2007, Australia
{prasun.roy,subhankar.ghosh}@student.uts.edu.au,
michael.blumenstein@uts.edu.au

[2] Indian Institute of Technology, Kharagpur 721302, WB, India
saumik@ece.iitkgp.ac.in

[3] Indian Statistical Institute, Kolkata 700108, WB, India
umapada@isical.ac.in

Abstract. We propose a data-driven approach for context-aware person image generation. Specifically, we attempt to generate a novel person image such that the synthesized instance can blend into a complex scene. In our method, the position, scale, and appearance of the generated person instance are semantically conditioned on the existing persons in the scene. The proposed technique consists of three sequential steps. At first, an image-to-image translation model infers a coarse semantic mask that represents the new person's spatial location, scale, and potential pose. Next, we introduce a data-centric approach to select the closest representation from a precomputed cluster of fine semantic masks. Finally, we use a multi-scale, attention-guided rendering network to transfer the appearance attributes from an exemplar image. The proposed strategy enables us to synthesize high-quality, semantically coherent, realistic human instances that can blend into an existing scene without altering the global context. We conclude our findings with relevant qualitative and quantitative evaluations.

Keywords: Person instance generation · Semantic consistency · GAN

1 Introduction

Person image generation is a challenging yet necessary task for many recent computer vision applications. Though the problem has been primarily addressed by utilizing different generative algorithms, often, the generation quality does not meet the requirements of the practical applications. Moreover, the existing person image generation algorithms rely on two main factors. First, they

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78389-0_20.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 293–309, 2025.
https://doi.org/10.1007/978-3-031-78389-0_20

heavily utilize the appearance and pose attributes of the target to generate the final image. This approach indirectly demands intricate supervision from users in the form of keypoints [5,18,19,25,29,30,36], parsing masks [20,31,35], or text inputs [34]. We can assume these attributes as *local attributes* or *local contexts* as they are only associated with the target person instance. Secondly, the local context-driven generation techniques are unsuitable for introducing a novel human instance in a complex scene due to the lack of imposed global semantic constraints, such as other existing people and objects in the environment. Consequently, the resulting target instances fail to blend convincingly into the given scene image. In this paper, we have addressed the exciting yet challenging task of novel person instance insertion, maintaining the global context of the scene. Additionally, the proposed method circumvents the necessity of user-specified local context information by introducing a data-driven distillation mechanism to automatically determine the best possible local attributes from the initial coarse estimation (Fig. 1).

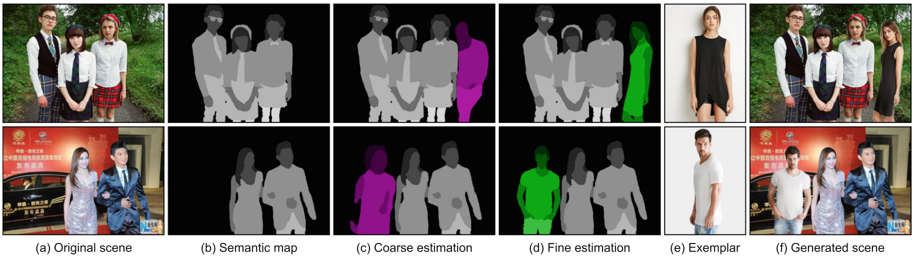


Fig. 1. Overview of the proposed method. **(a)** Original scene. **(b)** Semantic maps of existing persons in the scene. **(c)** Coarse estimation of the target person's location, scale, and potential pose. **(d)** Data-driven refinement of the coarse semantic map. **(e)** An exemplar of the target person. **(f)** Generated scene with the rendered target person.

Contributions: We summarize the main contributions of our work as follows.

1. The proposed technique uses global scene context followed by local appearance attributes, which allows us to synthesize human images that can blend into a complex scene with multiple existing persons.
2. The proposed technique utilizes a data-driven refinement strategy, significantly improving the perceptual quality and visual realism of the generated images.
3. The data-driven approach provides crude control over appearance variations through multiple fine semantic maps retrieved within a similarity score tolerance.
4. The proposed approach achieves state-of-the-art results in most qualitative and quantitative benchmarks.

2 Related Work

Person image generation: Generating high-quality realistic human images is a fundamental computer vision problem that directly and indirectly impacts multiple application domains, such as pose transfer, virtual try-on, and person re-identification. While the task is intriguing, the high degree of possible structural variations (poses) makes the problem inherently challenging. With the recent advances in generative modeling with generative adversarial networks (GANs) and diffusion models, the perceptual quality of synthesized images has significantly improved. Most works on person image generation focus on generating a person in a target pose given a source image and target pose attributes. The target pose attributes are given as keypoints [1,5,18,19,23–25,29,30,32,36], parsing masks [3,20,31,35], 3D surface maps [16,21], or text [34]. In [18], the proposed generation framework consists of novel pose synthesis followed by image refinement. The initial stage uses a UNet-based model to generate a coarse image, followed by refinement with another generative model in the second stage. In [19], the authors propose a two-stage architecture with a multi-branched generation network. Three mapping functions adversarially learn to map a random Gaussian noise into the relevant embedding feature space for targeted manipulation of the synthesized person image. Zhu et al. [36] have proposed a keypoint-based pose transfer method by incorporating a progressive attention transfer technique to divide the complex task of the generation into multiple repetitive simpler stages. Researchers have also explored 3D surface maps constructed from the DensePose [7] UV coordinates as the conditioning attribute for person image generation. In [21], the authors propose an end-to-end model incorporating surface-based pose estimation and a generative model to perform the pose transfer task. Li et al. [16] have estimated dense and intrinsic appearance flow between the poses to guide the pixels during the generation process. More recently, researchers have achieved significant visual improvements with carefully crafted attention mechanisms [23,24], pose transformers [32], and denoising diffusion models [1,3].

Semantically conditioned person image generation: Although several algorithms are proposed for person image generation, they require extensive information about the target pose for the generation process. Moreover, most existing algorithms consider the local attributes in the process, which makes them unsuitable for complex scenes with existing persons. Previously, researchers have introduced a relevant *random* person instance into a user-defined location [33] or a probabilistically estimated potential area [14,28] by performing a background context-conditioned instance-level search followed by image composition. In contrast, we aim to introduce a *specific* person instance into an optimally estimated scene location such that the new person contextually blends in with the existing persons. Recently, in [6], the authors have incorporated both local and global attributes for the person insertion problem in a disentangled GAN-based approach. In [13], the authors adopt an end-to-end conditional inpainting technique by finetuning a pretrained latent diffusion model to achieve similar goals.

3 Method

We propose a three-stage sequential architecture to address the problem. In the first stage, we estimate the potential location and pose of the target person from the global geometric context of the existing persons in the scene. The generated coarse semantic map performs appreciably in providing an estimate of the target location and scale. However, such a crude semantic map performs extremely poorly while attempting to transfer appearance attributes from an exemplar to render the final target. To mitigate this issue, we have taken a data-agnostic refinement strategy in the second stage to retrieve a representative semantic map for the target from an existing knowledge base. Finally, we render the target semantic map in the third stage by transferring appearance attributes from an exemplar of the target person. We show an overview of the proposed architecture in Fig. 2. Additionally, optional post-processing by image harmonization [2,11] can reduce blending inconsistencies between the foreground and background.

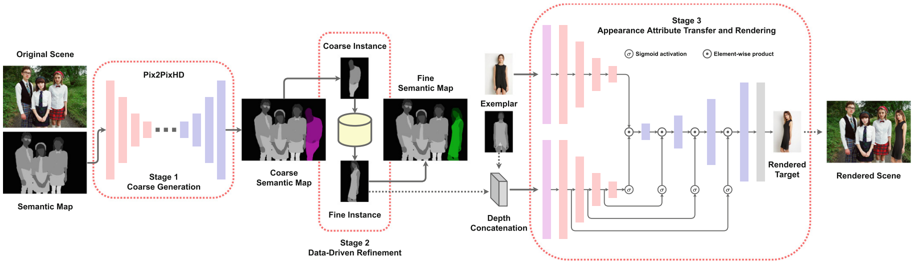


Fig. 2. The architecture of the proposed method consists of three sequential stages. **(a)** Initial coarse semantic map estimation from the global scene context in stage 1. **(b)** Data-driven refinement of the initially estimated coarse semantic map in stage 2. **(c)** Rendering the refined semantic map by transferring appearance attributes from an exemplar image in stage 3.

3.1 Coarse Generation Network

We use an encoder-decoder architecture to generate a rough estimate of the target person's position, scale and pose. This network performs an image-to-image translation from a semantic map S containing N persons to another semantic map T having the $(N+1)$-th person. The network aims to generate a coarse semantic map for a new person such that the new person is contextually relevant to the existing persons in the scene. We show a few examples of the coarse generation network in Fig. 3.

Both S and T are single-channel semantic maps containing eight labels corresponding to eight regions of a human body. This reduced set of label groups simplifies the semantic map generation while retaining sufficient information for high-quality image synthesis in the following stages. The reduced set of semantic label groups contains – background (0), hair (1), face (2), torso and upper limbs (3), upper body wear (4), lower body wear (5), lower limbs (6), and shoes (7). In [6], the authors also provide one channel for the face and another optional channel to specify the region boundary for the target. In contrast, we do not consider these additional channels due to our different approaches to refinement and rendering in later stages.

The coarse generation network adopts the default encoder-decoder architecture of Pix2PixHD [22]. We use a spatial dimension of 368 × 368 for the semantic maps. The original semantic maps are resized while maintaining the aspect ratio and then padded with zeros to have the desired square dimension. We use nearest-neighbor interpolation when resizing to preserve the number of label groups in the semantic maps. The only modification we apply to the default Pix2PixHD architecture is disabling the VGG feature-matching loss because it is possible to have a wide variation in the target person's location, scale, and pose, which leads to significant uncertainty in the generated semantic map.

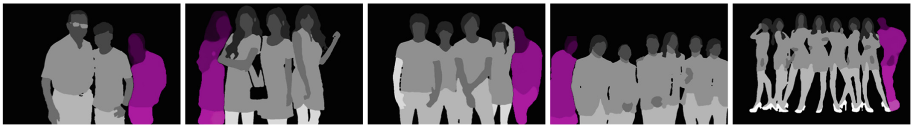


Fig. 3. Qualitative results of the coarse generation in stage 1. Semantic maps of existing persons are marked in gray, and the coarse estimation of the target semantic map is marked in purple.

3.2 Data-Driven Refinement Strategy

The rough semantic map provides a reasonable estimate for the target person, which is contextually coherent with the global semantics of the scene. While the spatial location and scale of the target are immediately usable to localize a new person into the scene, the semantic map itself is not sufficiently viable to produce realistic results. In [6], the authors use a multi-conditional rendering network (MCRN) on the roughly estimated semantic map, followed by a face refinement network (FRN) on the rendered target. While this approach produces some decent results, it is limited in scope due to solely relying on the initially generated rough semantic map from the essence generation network (EGN). We notice two crucial issues in this regard. Firstly, the use of a coarse semantic map highly affects the visual realism of the generated image. Secondly, it is not easy to achieve control over the appearance of the generated target with a fixed semantic representation. For example, if EGN produces a semantic map that appears to be a man while the intended exemplar is a woman. The subtle difference in core appearance attributes between the estimated semantic map and exemplar poses a significant challenge in practically usable generation results. We attempt to improve visual quality and appearance diversity in the generated results by introducing a data-driven refinement strategy with a clustered knowledge base.

We collect a set of finely annotated semantic maps of high-quality human images to construct a small database having a diverse range of natural poses. This database works as a knowledge base for our method. To optimally split the knowledge base into several clusters, we first encode the individual semantic maps using a VGG-19 [26] model pretrained on ImageNet [4]. The semantic maps are resized to a square grid of size 128 × 128, maintaining the aspect ratio and using zero padding. The resampling uses nearest-neighbor interpolation. After passing the resized image through the VGG-19 network, the final feature extraction layer produces an output of dimension $512 \times 4 \times 4$. To avoid too many features during clustering, we apply adaptive average pooling to map the feature space into a dimension of $512 \times 1 \times 1$. The pooled feature space is flattened to a 512-dimensional feature vector. We perform K-means clustering on the encoded feature vectors corresponding to the samples in the knowledge base. From our ablation study in Sec. 6, we have found 8 clusters work best for our case. After the algorithm converges, we split the knowledge base by the algorithm-predicted class labels.

During refinement, the coarse semantic map is center-cropped and resized to dimension 128 × 128, maintaining the aspect ratio. The resampling uses the same nearest-neighbor interpolation as earlier. The resized coarse semantic map is then similarly encoded and passed to the K-means algorithm for inference. After receiving a cluster assignment, we measure the cosine similarity between the encoded coarse semantic map and every sample previously classified as a cluster member. The refinement returns one or more existing samples by the similarity score-based ranking. The retrieved selection acts as the refined semantic map of the target person.

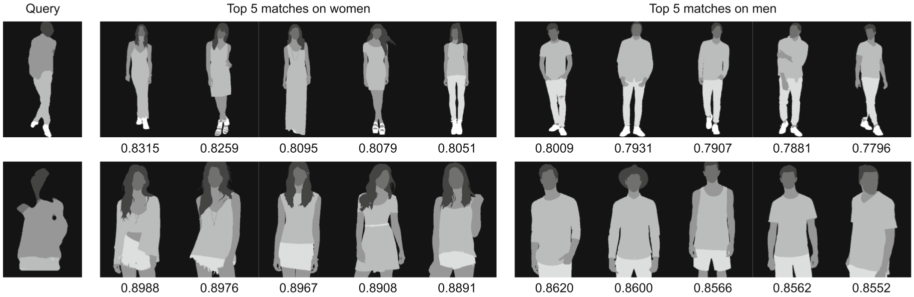


Fig. 4. Qualitative results of refinement in stage 2. The first column shows a coarse semantic map as the query, and the following columns show the top-5 refined semantic maps retrieved for both genders. The cosine similarity score for each retrieval is shown below the respective sample.

As we perform a nearest-neighbor search in the semantic feature space of samples in pre-computed clusters, given a coarse semantic map, we can dynamically select a refined candidate for either *women* or *men* as per requirements. This step can be automated if the gender of the exemplar is either known or estimated using a trained classifier. In Fig. 4, we show the top-5 matches for both *women* and *men* samples given a coarse semantic map as the query.

3.3 Appearance Attribute Transfer and Rendering

In [6], the authors train the rendering network on single instances extracted from multi-person images. In contrast, we impose the rendering task as a pose-transfer problem to transfer the appearance attributes conditioned on the pose transformation. Let us assume a pair of images I_A and I_B of the same person but with different poses P_A and P_B, respectively. We aim to train the network such that it renders a realistic approximation $\hat{I}_B$ (generated) of I_B (target) by conditioning the pose transformation (P_A, P_B) on the appearance attributes of I_A (exemplar). We represent each pose with a semantic map consisting of 7 label groups – background (0), hair (1), face (2), skin (3), upper body wear (4), lower body wear (5), and shoes (6). For effective attribute transfer on different body regions, the semantic map P is converted into a 6-channel binary heatmap (0 for the background and 1 for the body part) H where each channel indicates one specific body region. We use a spatial dimension of $3 \times 256 \times 256$ for I_A, I_B, and $\hat{I}_B$. Consequently, the same for H_A and H_B is $6 \times 256 \times 256$. We utilize a multi-scale attention-based generative network for rendering. The generator $\mathcal{G}$ takes the exemplar I_A and the depth-wise concatenated heatmaps (H_A, H_B) as inputs to produce an estimate $\hat{I}_B$ for the target I_B. The discriminator $\mathcal{D}$ takes the channel-wise concatenated image pairs, either (I_A, I_B) (real) or $(I_A, \hat{I}_B)$ (fake), to estimate a binary class probability map for 70×70 receptive fields (input patches).

The generator $\mathcal{G}$ has two separate but identical encoding pathways for I_A and (H_A, H_B). At each branch, the input is first mapped to a $64 \times 256 \times 256$ feature space by convolution (3×3 kernel, stride=1, padding=1, bias=0), batch normalization, and ReLU activation. The feature space is then passed through 4 consecutive downsampling blocks, where each block reduces the spatial dimension by half while doubling the number of feature maps. Each block consists of convolution (4×4 kernel, stride=2, padding=1, bias=0), batch normalization, and ReLU activation, followed by a basic residual block [8]. The network has a single decoding path that upsamples the combined feature space from both the encoding branches. We have 4 consecutive upsampling blocks in the decoder, where each block doubles the spatial dimension while compressing the number of feature maps by half. Each block consists of transposed convolution (4×4 kernel, stride=2, padding=1, bias=0), batch normalization, and ReLU activation, followed by a basic residual block. We apply an attention mechanism at every spatial dimension to preserve both coarse and fine appearance attributes in the generated image. Mathematically, for the first decoder block at the lowest resolution, $k = 1$,

$$I_1^D = D_1(I_4^E \odot \sigma(H_4^E)) \tag{1}$$

and for the subsequent decoder blocks at higher resolutions, $k = \{2, 3, 4\}$,

$$I_k^D = D_k(I_{k-1}^D \odot \sigma(H_{5-k}^E)) \tag{2}$$

where, I_k^D is the output from the *k-th* decoder block, I_k^E and H_k^E are the outputs from the *k-th* encoder blocks of image branch and pose branch respectively, σ denotes the *sigmoid* activation function, and $\odot$ denotes the Hadamard product. Finally, the resulting feature space goes through 4 consecutive basic residual blocks, followed by a convolution (1 × 1 kernel, stride=1, padding=0, bias=0) and *tanh* activation to project the feature maps into the final output image $\hat{I}_B$ of size 256 × 256.

The generator loss function $\mathcal{L}_\mathcal{G}$ is a combination of three objectives. It includes a pixel-wise l_1 loss $\mathcal{L}_1^\mathcal{G}$, an adversarial discrimination loss $\mathcal{L}_{GAN}^\mathcal{G}$ estimated using the discriminator $\mathcal{D}$, and a perceptual loss $\mathcal{L}_{VGG_\rho}^\mathcal{G}$ estimated using a VGG-19 network pretrained on ImageNet. Mathematically,

$$\mathcal{L}_1^\mathcal{G} = \left\| \hat{I}_B - I_B \right\|_1 \tag{3}$$

where $\|.\|_1$ denotes the l_1 norm or the mean absolute error.

$$\mathcal{L}_{GAN}^\mathcal{G} = \mathcal{L}_{BCE}\left(\mathcal{D}(I_A, \hat{I}_B), 1\right) \tag{4}$$

where $\mathcal{L}_{BCE}$ denotes the binary cross-entropy loss.

$$\mathcal{L}_{VGG_\rho}^\mathcal{G} = \frac{1}{h_\rho w_\rho c_\rho} \sum_{i=1}^{h_\rho} \sum_{j=1}^{w_\rho} \sum_{k=1}^{c_\rho} \left\| \phi_\rho(\hat{I}_B) - \phi_\rho(I_B) \right\|_1 \tag{5}$$

where ϕ_ρ denotes the output of dimension $c_\rho \times h_\rho \times w_\rho$ from the ρ*-th* layer of the VGG-19 network pretrained on ImageNet. We incorporate two perceptual loss terms for $\rho = 4$ and $\rho = 9$ into the cumulative generator objective. Therefore, the final generator objective is given by

$$\mathcal{L}_\mathcal{G} = \arg\min_G \max_D \quad \lambda_1 \mathcal{L}_1^\mathcal{G} + \lambda_2 \mathcal{L}_{GAN}^\mathcal{G} + \lambda_3 \left(\mathcal{L}_{VGG_4}^\mathcal{G} + \mathcal{L}_{VGG_9}^\mathcal{G}\right) \tag{6}$$

where λ_1, λ_2, and λ_3 are the tunable weights for the corresponding loss components.

The discriminator $\mathcal{D}$ is a generic PatchGAN [10] that operates on 70 × 70 receptive fields of the input. It takes the depth-wise concatenated image pairs, either (I_A, I_B) or $(I_A, \hat{I}_B)$, as a real (1) or fake (0) image transition, respectively.

The discriminator loss $\mathcal{L}_\mathcal{D}$ has only a single component $\mathcal{L}_{GAN}^\mathcal{D}$, calculated as the average BCE loss over real and fake transitions. Mathematically,

$$\mathcal{L}_{GAN}^\mathcal{D} = \frac{1}{2}\left[\mathcal{L}_{BCE}(\mathcal{D}(I_A, I_B), 1) + \mathcal{L}_{BCE}(\mathcal{D}(I_A, \hat{I}_B), 0)\right] \tag{7}$$

Therefore, the final discriminator objective is given by

$$\mathcal{L}_\mathcal{D} = \arg\min_D \max_G \quad \mathcal{L}_{GAN}^\mathcal{D} \tag{8}$$

4 Experimental Setup

Datasets: We use the multi-human parsing dataset LV-MHP-v1 [15] to train the coarse generation network in stage 1. The dataset contains 4980 high-quality images, each having at least two persons (average is three), and the respective semantic annotations for every individual in the scene. The annotation includes 19 label groups – background (0), hat (1), hair (2), sunglasses (3), upper clothes (4), skirt (5), pants (6), dress (7), belt (8), left shoe (9), right shoe (10), face (11), left leg (12), right leg (13), left arm (14), right arm (15), bag (16), scarf (17), and torso skin (18). As discussed in Sec. 3.1, we reduce the original label groups to 8 by merging as – background + bag (0), hair (1), face (2), both arms + torso skin (3), hat + sunglasses + upper clothes + dress + scarf (4), skirt + pants + belt (5), both legs (6), both shoes (7). While training the coarse generation network, we select one random instance of a scene as the target person and the remaining instances as the input context. We prepare 14854 training pairs from 4945 images and 115 test pairs from the remaining 35 images.

For data-driven refinement in stage 2 and rendering network in stage 3, we use the DeepFashion [17] dataset. The dataset contains high-quality single-person instances with wide pose and attire variations. A subset of the samples has color annotations for 16 semantic label groups. We reduce the number of label groups to 7 by merging multiple semantic regions as – background + bag (0), hair + headwear (1), face + eyeglass (2), neckwear + skin (3), top + dress + outer (4), skirt + belt + pants (5), leggings + footwear (6). We prepare 9866 images and corresponding semantic maps for creating our clustered database. We select 9278 image pairs for training and 786 image pairs for testing the rendering network.

Training details: We train the coarse generation network with batch size 16 and VGG feature-matching loss disabled. All other training parameters are kept to defaults as specified by the authors of Pix2PixHD [22].

The clustering follows Lloyd's K-means algorithm with 8 clusters, a relative tolerance of $1e^{-4}$, 1000 maximum iterations, and 10 random initializations for the centroids.

For the rendering network, we set $\lambda_1 = 5$, $\lambda_2 = 1$, and $\lambda_3 = 5$ in the generator objective. The parameters of both the generator and discriminator networks are initialized before optimization by sampling values from a normal distribution of mean = 0 and standard deviation = 0.02. We use the stochastic Adam optimizer [12] to update the parameters of both networks. We set the learning rate $\eta = 1e^{-3}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$, $\epsilon = 1e^{-8}$, and weight decay = 0 for both optimizers. The network is trained with batch size 4.

Evaluation metrics: Although quantifying visual quality is an open challenge in computer vision, researchers widely use a few quantifiable metrics to assess the perceptual quality of generated images. Following on from earlier published works [1,3,5,6,18,20,23–25,29–32,35,36], we calculate Structural Similarity Index (SSIM), Inception Score (IS), Detection Score (DS), Percentage of Correct Keypoints (PCKh), Average Keypoint Distance (AKD), Keypoint Visibility Retention Error (KVRE), and Learned Perceptual Image Patch Similarity (LPIPS) for quantitative benchmarks. SSIM considers image degradation as the perceived change in the structural information. IS estimates the KL divergence between the label and marginal distributions for many images using the Inception network [27] as an image classifier. DS measures the visual quality as an object detector's target class recognition confidence. PCKh quantifies the shape consistency based on the fraction of correctly aligned keypoints. AKD measures the average Euclidean distance between the target pose keypoints and the re-estimated keypoints from the rendered person instances to assess the impact of rendering on pose alignment. KVRE estimates the mismatch in keypoint visibility states as a measure of retaining pose consistency after rendering. LPIPS quantifies the perceptual similarity between the target and generated images by utilizing spatial feature maps retrieved from deep convolutional architectures such as VGG [26] or SqueezeNet [9].

5 Results

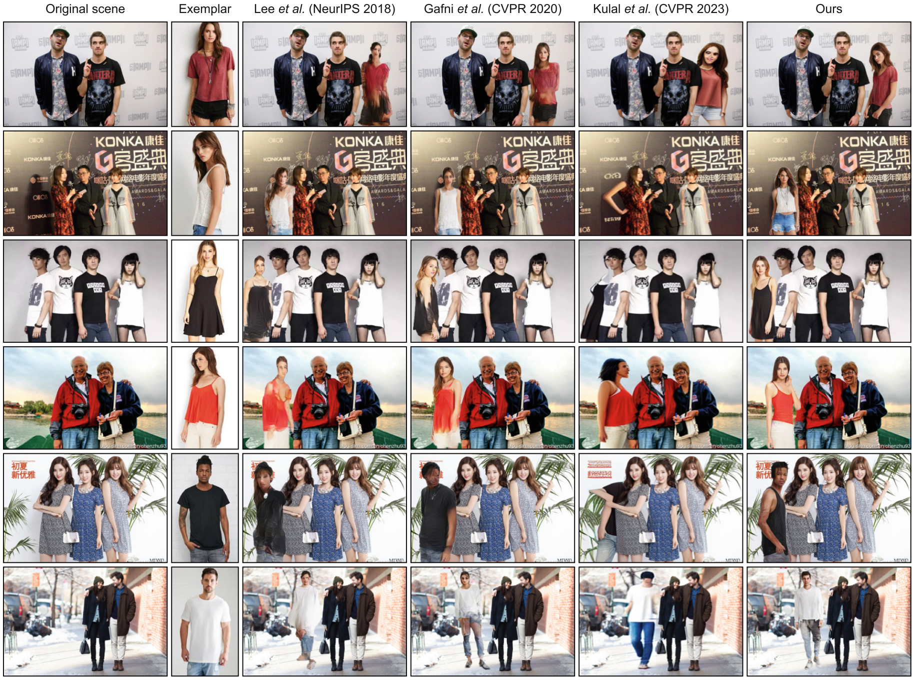

Original scene
Exemplar
Lee *et al.* (NeurIPS 2018)
Gafni *et al.* (CVPR 2020)
Kulal *et al.* (CVPR 2023)
Ours

Fig. 5. Qualitative comparison of the proposed method with existing person insertion techniques [6,13,14]. Additional results are included in the **supplementary material**.

Qualitative and quantitative comparisons: We have performed an extensive range of experiments to explore and analyze the efficacy of the proposed method. In Fig. 5, we compare our approach qualitatively with existing person image insertion techniques [6,13,14]. Additional results are included in the *supplementary material*. The visual analysis shows unrealistic persons for [14] and inadequate rendering for [6]. In [13], the authors have assumed the objective as a conditional inpainting problem, improving the overall visual quality of image blending over [6,14]. However, in our experiments, the technique [13] often fails to insert a new person into multi-person scenes, and the method lacks a faithful appearance attribute transfer to retain the exemplar's identity. In contrast, the proposed method can produce photorealistic, visually appealing results for person insertion into a complex scene with a semantically consistent pose while preserving the appearance and identity of the exemplar. To analyze the overall generation quality of the rendering network, we perform a quantitative comparison against recently proposed person image generation algorithms [1,3,5,6,18,20,23–25,29–32,35,36]. As shown in Table 1, the proposed rendering method outperforms existing algorithms in most evaluation metrics.

Subjective evaluation: Additionally, we have conducted an opinion-based user study with 72 volunteers to rate the final generated scenes as real or fake. Following the protocols in [6], we have kept the allowed observation time unrestricted during the study. The proposed method has received a mean opinion score of 64.4% against 59.2% by [13], 51.8% by [6], and 32.1% by [14].

Table 1. Quantitative comparison of the rendering network.

Method	**SSIM ↑**	**IS ↑**	**DS ↑**	**PCKh ↑**	**AKD ↓**	**KVRE ↓**	**LPIPS ↓ (VGG)**	**LPIPS ↓ (SqzNet)**
PG^2 [18]	0.773	3.163	0.951	0.89	-	-	0.523	0.416
Deform [25]	0.760	3.362	0.967	0.94	-	-	-	-
VUNet [5]	0.763	3.440	0.972	0.93	-	-	-	-
PATN [36]	0.773	3.209	0.976	0.96	-	-	0.299	0.170
XingGAN [30]	0.762	3.060	0.917	0.95	-	-	0.224	0.144
BiGraphGAN [29]	0.779	3.012	0.954	0.97	-	-	0.187	0.114
ADGAN [20]	0.677	3.116	0.938	0.96	4.582	0.026	0.256	0.144
GFLA [24]	0.709	3.291	0.946	0.96	4.119	0.023	0.269	0.145
PISE [31]	0.759	3.210	0.974	0.96	4.114	0.024	0.201	0.109
DPTN [32]	0.707	3.229	0.975	0.96	4.216	0.025	0.335	0.193
NTED [23]	0.725	3.438	**0.986**	0.97	3.655	0.021	0.229	0.131
CASD [35]	0.724	**3.446**	0.984	0.97	3.504	0.022	0.222	0.120
PIDM [1]	0.718	-	-	0.97	4.131	0.023	0.221	0.116
UPGPT [3]	0.679	-	-	0.94	5.306	0.030	0.285	0.167
WYWH (KP) [6]	0.788	3.189	-	-	-	-	0.271	0.156
WYWH (DP) [6]	0.793	3.346	-	-	-	-	0.264	0.149
Ours	**0.845**	3.351	0.968	**0.98**	**2.355**	**0.018**	**0.124**	**0.064**
Ground Truth	1.000	3.687	0.970	1.00	0.000	0.000	0.000	0.000

6 Ablation Study

Feature representation during clustering: As mentioned in Sec. 3.2, we use 512-dimensional VGG-encoded features to guide the refinement process. To evaluate the significance of feature representation in the proposed refinement strategy, we compare VGG-encoded features with raw pixel features in our ablation analysis by converting the input image into a feature vector. The conversion process downscales (nearest-neighbor interpolation) the original 176 × 256 images to 22 × 32, keeping the aspect ratio intact, followed by flattening to a 704-dimensional feature vector. We evaluate both the feature representation techniques for different numbers of clusters ($K = 8,\ 16,\ 32,\ 64$). As shown in Table 2, for a particular number of clusters K, VGG-encoded feature representation outperforms the raw pixel-based representation on the average similarity score of top retrievals. Fig. 6 illustrates the similarity score-based ranking of retrieved samples with each feature encoding type for both genders. The VGG feature-based clustering results in a better resemblance between the query and retrieved semantic maps. Our study shows that $K = 8$ works best in most cases.

Table 2. Quantitative ablation analysis on the feature representation for clustering.

Feature	Number of clusters	Average cosine similarity (top-1 match) ↑			Average cosine similarity (top-5 matches) ↑		
		Men	Women	Overall	Men	Women	Overall
Pixel	K = 8	0.7127	**0.7562**	**0.7608**	0.6912	**0.7366**	**0.7402**
	K = 16	**0.7146**	0.7539	0.7598	**0.6933**	0.7357	0.7402
	K = 32	0.7014	0.7449	0.7492	0.6768	0.7270	0.7302
	K = 64	0.5852	0.6767	0.6810	0.5580	0.6301	0.6346
VGG	K = 8	**0.8212**	**0.8319**	**0.8390**	0.7933	**0.8171**	**0.8245**
	K = 16	0.8184	0.8307	0.8371	**0.7941**	0.8146	0.8227
	K = 32	0.8073	0.8313	0.8379	0.7824	0.8140	0.8225
	K = 64	0.7995	0.8290	0.8368	0.7715	0.8109	0.8208

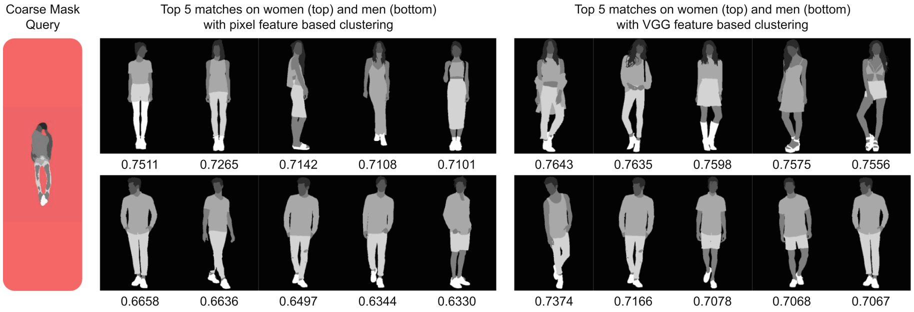

Coarse Mask Query
Top 5 matches on women (top) and men (bottom) with pixel feature based clustering
Top 5 matches on women (top) and men (bottom) with VGG feature based clustering
0.7511
0.7265
0.7142
0.7108
0.7101
0.7643
0.7635
0.7598
0.7575
0.7556
0.6658
0.6636
0.6497
0.6344
0.6330
0.7374
0.7166
0.7078
0.7068
0.7067

Fig. 6. Visual ablation analysis on the feature representation for clustering. The cosine similarity score for each retrieval is shown below the respective sample.

Attention mechanism: Attention in the rendering network plays a crucial role in the generated image quality. We explore four different settings to validate and select the optimal attention strategy. In the first setting (**Baseline**), we remove all attention operations and depth-wise concatenate I_4^E with H_4^E. The concatenated feature space is passed through the decoder block. As shown in Table 3, the Baseline model performs worst among all variants. We consider only one attention pathway in the rendering network in the second and third ablation settings. In the second variant (**HR only**), the attention operation is performed at the highest feature resolution only (just before the decoder block D_4). Similarly, in the third variant (**LR only**), the attention operation is performed at the lowest feature resolution only (just before the decoder block D_1). In the final settings (**Full**), we use the proposed attention mechanism as shown in Fig. 2 and described in Sec. 3.3. We train and evaluate all four variants on the same dataset splits while keeping all experimental conditions the same, as noted in Sec. 4. We show the evaluated metrics in Table 3 along with qualitative results in Fig. 7. We conclude from the analytical and visual results that the proposed attention mechanism provides the best generation performance.

Table 3. Quantitative ablation analysis of the rendering network.

Model	**SSIM ↑**	**IS ↑**	**DS ↑**	**PCKh ↑**	**AKD ↓**	**KVRE ↓**	**LPIPS ↓ (VGG)**	**LPIPS ↓ (SqzNet)**
Baseline	0.657	**3.667**	0.902	0.46	9.429	0.279	0.338	0.260
HR only	0.825	3.271	0.954	0.96	4.981	0.021	0.154	0.088
LR only	0.840	3.326	0.966	0.96	3.774	0.020	0.131	0.068
Full	**0.845**	3.351	**0.968**	**0.98**	**2.355**	**0.018**	**0.124**	**0.064**
Ground Truth	1.000	3.687	0.970	1.00	0.000	0.000	0.000	0.000

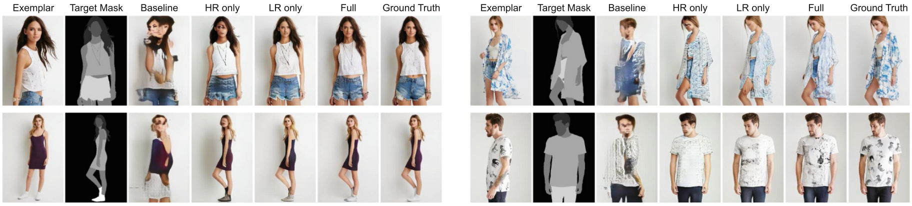


Fig. 7. Visual ablation analysis of the rendering network.

Refinement: We show the efficacy of the data-driven refinement on the final generation in Fig. 8 by comparing the rendered scene with and without applying the refinement technique on the initially estimated coarse semantic map.

7 Limitations

Although the proposed method can produce high-quality, visually appealing results for a wide range of complex scenes, there are a few occasions when the

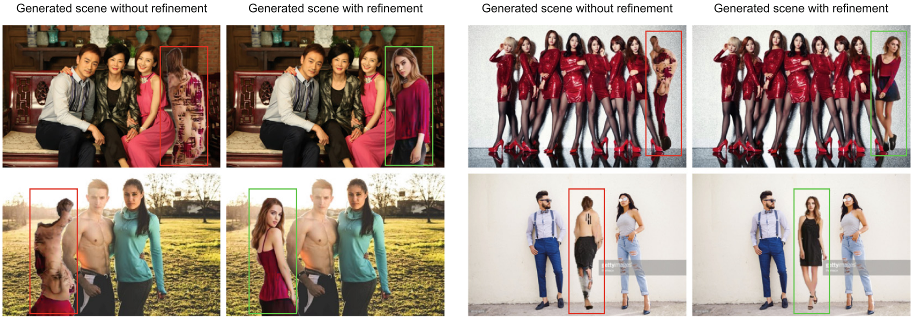

Generated scene without refinement
Generated scene with refinement
Generated scene without refinement
Generated scene with refinement

Fig. 8. Visual ablation analysis of the refinement strategy on rendering. Each pair of examples shows a rendered human in the modified scene *without* (**left**) or *with* (**right**) the refinement, marked with red and green bounding boxes, respectively.

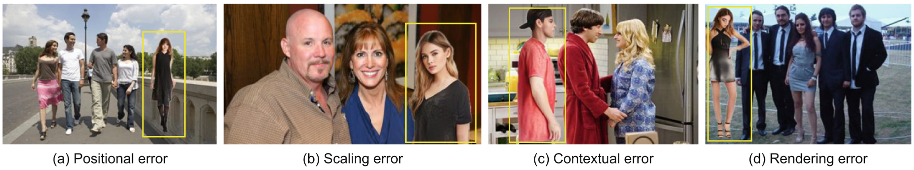

(a) Positional error
(b) Scaling error
(c) Contextual error
(d) Rendering error

Fig. 9. Limitations of the proposed method. The limiting cases may arise as a result of inconsistencies in **(a)** *position*, **(b)** *scale*, **(c)** *context*, or **(d)** *rendering*.

technique fails to generate a realistic outcome. Due to a disentangled multi-stage approach, these limiting cases may occur from different pipeline components. Coarse generation in stage 1 provides the spatial location and scale of the target person. Therefore, the wrong inference in this step leads to a misinterpretation of the position and scale in the final target. The refined semantic target map is retrieved from the pre-partitioned clusters based on encoded features of the coarse semantic map in stage 2. Consequently, an extremely rough generation in stage 1 or a misclassified outlier during clustering in stage 2 can lead to a generated person that does not blend well with the existing persons in the scene. Finally, due to a supervised approach of training the renderer in stage 3, the appearance attribute transfer may fail to generate high-quality outputs for imbalanced or unconventional target poses. We show a few such cases in Fig. 9.

8 Conclusions

In this work, we propose a novel technique for scene-aware person image synthesis by conditioning the generative process on the global context. The method is divided into three independent stages to focus on individual subtasks concisely. First, we use a coarse generation network based on a conditional image-to-image translation architecture to estimate the target person's spatial and pose attributes. While the spatial characteristics in the initial semantic map provide sufficient geometric information for the target, the semantic map does not preserve enough label group correctness, leading to improper attribute transfer in the rendering stage. We mitigate this issue through a data-driven distillation of the coarse semantic map by selecting candidate maps from a clustered knowledge base using a similarity score-based ranking. Finally, the appearance attributes from the exemplar are transferred to the selected candidate semantic map using a generative renderer. The rendered instance is then injected into the original scene using the geometric information obtained during coarse generation. In our experiments, we achieve highly detailed, realistic visual outcomes, which are further supported by relevant analytical evaluations.

References

1. Bhunia, A.K., Khan, S., Cholakkal, H., Anwer, R.M., Laaksonen, J., Shah, M., Khan, F.S.: Person image synthesis via denoising diffusion model. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
2. Chen, J., Zhang, Y., Zou, Z., Chen, K., Shi, Z.: Dense pixel-to-pixel harmonization via continuous image representation. arXiv preprint arXiv:2303.01681 (2023)
3. Cheong, S.Y., Mustafa, A., Gilbert, A.: UPGPT: Universal diffusion model for person image generation, editing and pose transfer. In: The IEEE/CVF International Conference on Computer Vision (ICCV) Workshops (2023)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
5. Esser, P., Sutter, E., Ommer, B.: A variational U-Net for conditional appearance and shape generation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
6. Gafni, O., Wolf, L.: Wish you were here: Context-aware human generation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
7. Güler, R.A., Neverova, N., Kokkinos, I.: Densepose: Dense human pose estimation in the wild. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
9. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size. arXiv preprint arXiv:1602.07360 (2016)
10. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-Image translation with conditional adversarial networks. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
11. Ke, Z., Sun, C., Zhu, L., Xu, K., Lau, R.W.: Harmonizer: Learning to perform white-box image and video harmonization. In: The European Conference on Computer Vision (ECCV) (2022)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: The International Conference on Learning Representations (ICLR) (2015)

13. Kulal, S., Brooks, T., Aiken, A., Wu, J., Yang, J., Lu, J., Efros, A.A., Singh, K.K.: Putting people in their place: Affordance-aware human insertion into scenes. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
14. Lee, D., Liu, S., Gu, J., Liu, M.Y., Yang, M.H., Kautz, J.: Context-aware synthesis and placement of object instances. In: The Conference on Neural Information Processing Systems (NeurIPS) (2018)
15. Li, J., Zhao, J., Wei, Y., Lang, C., Li, Y., Sim, T., Yan, S., Feng, J.: Multiple-human parsing in the wild. arXiv preprint arXiv:1705.07206 (2017)
16. Li, Y., Huang, C., Loy, C.C.: Dense intrinsic appearance flow for human pose transfer. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
17. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: DeepFashion: powering robust clothes recognition and retrieval with rich annotations. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
18. Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., Van Gool, L.: Pose guided person image generation. In: The Conference on Neural Information Processing Systems (NeurIPS) (2017)
19. Ma, L., Sun, Q., Georgoulis, S., Van Gool, L., Schiele, B., Fritz, M.: Disentangled person image generation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
20. Men, Y., Mao, Y., Jiang, Y., Ma, W.Y., Lian, Z.: Controllable person image synthesis with attribute-decomposed gan. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
21. Neverova, N., Guler, R.A., Kokkinos, I.: Dense pose transfer. In: The European Conference on Computer Vision (ECCV) (2018)
22. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
23. Ren, Y., Fan, X., Li, G., Liu, S., Li, T.H.: Neural texture extraction and distribution for controllable person image synthesis. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
24. Ren, Y., Yu, X., Chen, J., Li, T.H., Li, G.: Deep image spatial transformation for person image generation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
25. Siarohin, A., Sangineto, E., Lathuilière, S., Sebe, N.: Deformable GANs for pose-based human image generation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: The International Conference on Learning Representations (ICLR) (2015)
27. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
28. Tan, F., Bernier, C., Cohen, B., Ordonez, V., Barnes, C.: Where and who? automatic semantic-aware person composition. In: The IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2018)
29. Tang, H., Bai, S., Torr, P.H., Sebe, N.: Bipartite graph reasoning GANs for person image generation. In: The British Machine Vision Conference (BMVC) (2020)
30. Tang, H., Bai, S., Zhang, L., Torr, P.H., Sebe, N.: XingGAN for person image generation. In: The European Conference on Computer Vision (ECCV) (2020)

31. Zhang, J., Li, K., Lai, Y.K., Yang, J.: PISE: Person image synthesis and editing with decoupled gan. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
32. Zhang, P., Yang, L., Lai, J.H., Xie, X.: Exploring dual-task correlation for pose guided person image generation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
33. Zhao, H., Shen, X., Lin, Z., Sunkavalli, K., Price, B., Jia, J.: Compositing-aware image search. In: The European Conference on Computer Vision (ECCV) (2018)
34. Zhou, X., Huang, S., Li, B., Li, Y., Li, J., Zhang, Z.: Text guided person image synthesis. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
35. Zhou, X., Yin, M., Chen, X., Sun, L., Gao, C., Li, Q.: Cross attention based style distribution for controllable person image synthesis. In: The European Conference on Computer Vision (ECCV) (2022)
36. Zhu, Z., Huang, T., Shi, B., Yu, M., Wang, B., Bai, X.: Progressive pose attention transfer for person image generation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

GM-GAN: Geometric Generative Models Based on Morphological Equivariant PDEs and GANs

El Hadji S. Diop[1(✉)], Thierno Fall[1], Alioune Mbengue[1,2],
and Mohamed Daoudi[3,4]

[1] Department of Mathematics, NAGIP-Nonlinear Analysis and Geometric Information Processing Group, University Iba Der Thiam, BP 967, Thies, Senegal
ehsdiop@hotmail.com

[2] Department of Mathematics and Computer Science, University Cheikh Anta Diop, Dakar, Senegal

[3] IMT Nord Europe, Univ. Lille, Centre for Digital Systems, 59000 Lille, France
mohamed.daoudi@imt-nord-europe.fr

[4] Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, 59000 Lille, France

Abstract. This work deals with image generation, two main problems are addressed: (*i*) improvements of specific feature extraction while accounting at multiscale levels intrinsic geometric features, and (*ii*) equivariance of the network for reducing the complexity and providing a geometric interpretability. We propose a geometric generative model based on an equivariant partial differential equation (PDE) for group convolution neural networks (G-CNNs), so called PDE-G-CNNs, built on morphology operators and generative adversarial networks (GANs). The proposed geometric morphological GAN model, termed as GM-GAN, is obtained thanks to morphological equivariant convolutions in PDE-G-CNNs. GM-GAN is evaluated qualitatively and quantitatively using FID on MNIST and RotoMNIST, preliminary results show noticeable improvements compared classical GAN.

Keywords: PDEs · Equivariance · Morphological operators · Riemannian manifolds · Lie group · Symmetries · CNNs

1 Introduction

Significant advances in deep learning progress are attributed to CNNs [23]. Despite its successful applications in many real life problems, it is still not very clear why deep learning techniques work. Pursuing this goal, many works attempt to give an answer to this so challenging question by setting mathematical frameworks that underlie the process. A promising direction is to consider symmetries as a fundamental design principle for network architectures. Among noticeable properties in CNNs, the equivariance concerning translations played an important role. Equivariance means that the operation of performing a transformation of the input data then passing them through the network is the same

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 310–325, 2025.
https://doi.org/10.1007/978-3-031-78389-0_21

as passing the input data through the network and then performing a transformation of the output. CNNs are inherently translationally invariant; however, invariance does not extend straightforward to other types of transformations. G-CNNs [3,9,10] were introduced to tackle this issue by generalizing CNNs in a way such that symmetries are incorporated and fully exploited in the learning process. Very recently, PDE-G-CNNs [4,31] were proposed as PDEs-based framework based that generalized G-CNNs. The proposed PDEs were solved by providing analytical kernels approximations [31] and exact kernels sub-Riemannian approximations [4]. Intensive research on equivariant operators other than transformations is still conducted [20,29,33].

GANs [21,22] brought a new perspective to the deep learning community, deep learning with adversarial training is considered today as one of the most robust technique. With adversarial generative networks, there exists not only a good neural network-based classifier, referred to as the discriminator network, but also a generative network capable of producing realistic adversarial samples, all within a single architecture. This means that we now have a network that is aware of internal representations through its training to distinguish real inputs from artificial ones. Many extensions have been built for increasing its performances. Conditional GAN (CGAN) [19] was proposed as an extension of original GAN for generating facial images on the basis of facial attributes. Deep Convolutional GAN (DCGAN) [28] was proposed for image generation where both the generator and discriminator networks are convolutional. GRAN [24] is a GAN model based on a sequential process. Bidirectional GAN (BiGAN) and extensions [6,12] were proposed to map data into a latent code similar to an autoencoder. Generative Multi-Adversarial Network (GMAN) [16] was proposed for extending the minimax game to multiple players in GANs. In a different perspective, Wasserstein Generative Adversarial Network (WGAN) [1] was introduced to reduce the instability problems that occur during the training step, and also to eliminate the mode collapse effect. GANs and variants lack an inference mechanism.

In this work[1], we aim at providing noticeable improvements of former GAN models by using a geometric approach based on equivariant operators defined in a Lie group, and on mathematical morphology formulated in Riemannian manifolds. Main contributions can be summarized as follows: **1)** proposition of a new geometric generative model based on a new PDE-G-CNNs built on multiscale morphology operators and geometric image processing techniques, **2)** improvements of specific feature extraction while accounting intrinsic geometric features at multiple scales/levels, and **3)** equivariance of the network resulting in a complexity reduction and a geometric interpretability. Additional details and results that did not fit into the main paper can be found in supplementary material.

The paper is organized as follows. In Section 2, we define the notion of equivariance in Lie groups and present the group invariance property on Riemannian

[1] This work was partially supported by the ANR project Human4D ANR-19-CE23-0020 and the Centre d'Excellence Africain en Mathématique, Informatique et Technologie de l'Information et de la Communication (CEA-MITIC).

manifolds. In Section 3, we present the viscosity solutions for morphological dilations and erosions formulated as Lie group morphological convolutions in Riemannian manifolds. The proposed geometric generative (GM-GAN) model in presented in Section 4. Section 5 is dedicated to numerical experiments and comparisons with classical GAN models. The paper ends in Section 6 where concluding remarks and perspectives are discussed.

2 Equivariance and homogeneous spaces on Riemannian manifolds

Let M be a smooth manifold and $x \in M$. A linear mapping $v : C^\infty(M;\mathbb{R}) \to \mathbb{R}$ satisfying the Leibniz rule:

$$\forall\ f_1, f_2 \in C^\infty(M;\mathbb{R}) \quad v(f_1 f_2) = f_1(x)v(f_2) + v(f_1)f_2(x) \tag{1}$$

is called a derivation at x. For all $x \in M$, the set of derivations at x forms a real vector space of dimension d denoted T_xM so called the tangent space at x; its elements can be also called tangent vectors. In Euclidean space, an operator satisfying (1) is the derivative along a specific direction, and this definition is a generalization of derivatives on smooth manifolds in general.

Let G be a connected Lie group. We assume that the group G acts regularly on the spaces P and Q, meaning that there exists regular maps $\rho_P : G \times P \to P$ and $\rho_Q : G \times Q \to Q$ respectively defined for all $r, h \in G$, by:

$$\rho_P(rh, x) = \rho_P(r, \rho_P(h, x)) \text{ and } \rho_Q(rh, x) = \rho_Q(r, \rho_Q(h, x)), \tag{2}$$

making ρ_P and ρ_Q group actions on their respective spaces. In addition, we assume that the group G acts transitively on the spaces (smooth manifolds), meaning that for any two elements in these spaces, there exists a transformation in G that maps them to each other. This implies that P and Q can be viewed as homogeneous spaces.

Definition 1. *A Riemannian metric on a differentiable manifold M is given by a scalar product μ on each tangent space T_xM depending smoothly on the base point $x \in M$, that is, $\forall\ x \in M$, $\mu_x : T_xM \times T_xM \to \mathbb{R}$ is a symmetric, bilinear and positive definite map, and μ_x varies smoothly over M.*
A Riemannian manifold (M, μ) is a differentiable manifold M equipped with a Riemannian metric μ.

Definition 2. *Let G a connected Lie group with neutral element e and (M, μ) a connected Riemannian manifold. A left action of G on (M, μ) is an application $\varphi : G \times (M, \mu) \to (M, \mu)$ satisfying:*

1. *$\varphi(e, x) = x$, $\forall\ x \in (M, \mu)$.*
2. *$\varphi(g, \varphi(h, x) = \varphi(gh, x)$, $\forall\ g, h \in G$ and $\forall\ x \in (M, \mu)$.*

Let $\varphi : G \times (M, \mu) \to (M, \mu)$ be a left action of G on (M, μ). For a fixed $g \in G$, we define $\varphi_g : (M, \mu) \to (M, \mu)$; $x \mapsto \varphi_g(x) = \varphi(g, x)$.
The function $\varphi : G \times (M, \mu) \to (M, \mu)$ is a left action if $\forall\ g, h \in G$, one has: $\varphi_e = id_M$ and $\varphi_g \circ \varphi_h = \varphi_{gh}$.
Let $\varphi_h : (M, \mu) \longrightarrow (M, \mu)$ be the left group action (considered here as a multiplication) by an element $h \in G$ defined $\forall\ x \in (M, \mu)$ by:

$$\varphi_h(x) = h \cdot x. \tag{3}$$

Let $\mathcal{L}_h$ be the left regular representation of G on functions f defined on M by $(\mathcal{L}_h f)(x) = f(\varphi_{h^{-1}}(x))$, with h^{-1} as the inverse of $h \in G$.
We consider a layer in a neural network as an operator (from functions on M_1 to functions on M_2). To ensure the equivarianc of the network, we shall require the operator to be equivariant with respect to the actions on the function spaces.

Let x_0 be an arbitrary fixed point on the connected Riemannian manifold (M, μ). Let $\pi : G \to (M, \mu)$ be the projection defined by assigning to each element h of G an element of (M, μ) in the following:

$$\forall\ h \in G \quad \pi(h) = \varphi_h(x_0). \tag{4}$$

In other words, once a reference point $x_0 \in (M, \mu)$ is chosen, the projection $\pi(h)$ assigns to every element h in G the unique point in (M, μ) to which h sends the chosen reference point x_0 under the action of φ_h given by (3).

In this work, we consider a connected Lie group G that acts transitively on the connected Riemannian manifold (M, μ). This means that for any points x and $y \in (M, \mu)$, there exists an element $h \in G$ such that $\varphi_h(x) = y$, corresponding to the definition of an homogeneous space under the action of the group G.

Definition 3. *Let G be a connected Lie group with homogeneous spaces M and N. Let ϕ be an operator on functions from M to functions on N. We say that ϕ is equivariant with respect to G if for all functions f, one has:*

$$\forall\ h \in G,\ (\phi \circ \mathcal{L}_h) f = (\mathcal{L}_h \circ \phi) f, \tag{5}$$

Let $h \in G$, $x \in (M, \mu)$ and $T_x M$ be the tangent space of (M, μ) at the point x. The pushforward of the group action φ_h denoted $(\varphi_h)_*$ is defined by: $(\varphi_h)_* : T_x M \to T_{\varphi_h(x)} M$ such that for all smooth functions f on (M, μ) and all $v \in T_x M$, one has: $((\varphi_h)_* v) f := v(f \circ (\varphi_h)_*)$.

For all $x \in (M, \mu)$, we refer to G-invariance of vector fields $X : x \mapsto T_x M$ if $\forall\ h \in G$ and for all differentiable functions f, one has $X(x) f = X(\varphi_h(x))[\mathcal{L}_h f]$.

Definition 4. *A vector field X on (M, μ) is invariant with respect to G if $\forall\ h \in G$ and $\forall\ x \in (M, \mu)$, one has: $X(\varphi_h(x)) = (\varphi_h)_* X(x)$.*

Definition 5. *A $(0, 2)$-tensor field μ on M is G-invariant if $\forall\ h \in G$, $\forall\ x \in M$ and $\forall\ v, w \in T_x(M)$, one has: $\mu|_h(v, w) = \mu|_{\varphi_h(x)}((\varphi_h)_* v, (\varphi_x)_* w)$.*

It follows from Definition 5 that properties derived from metric tensor field G invariance and vector field G invariance are the same.

Definition 6. *Let* (M, μ) *a connected Riemannian manifold,* $x, y \in (M, \mu)$. *The distance between* x *and* y *is defined as:* $d_\mu(x, y) = \inf_{\gamma \, \in \, \Gamma_t(x,y)} \int_0^t \sqrt{\mu|_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}dt$, *with* $\Gamma_t(x, y) = \{\gamma : [0, t] \longrightarrow (M, \mu) \;$ *of class* $\; C^1, \gamma(0) = x \;$ *and* $\; \gamma(t) = y\}$.

Definition 7. *The cut locus is defined as the set of points* $x \in M$ *(or* $h \in G$*) from which the distance map is not smooth (except at* x *or* h*).*

Proposition 1. *Let* $x, y \in (M, \mu)$ *such that* $\varphi_h(y)$ *is away from the cut locus of* $\varphi_h(x)$. *Then,* $\forall\ h \in G$, *one has:* $d_\mu(x, y) = d_\mu\big(\varphi_h(x), \varphi_h(y)\big)$.

Remark 1. Staying away from the cut locus provides a unique distance in Definition 6. Also, thanks to Proposition 1, d_μ shares the same symmetries, since we derive it from a tensor field invariant under G.

3 Group morphological convolutions and PDEs

Link between morphological multiscale flat erosions and PDEs was established by running in $\mathbb{R}^n$ a first order Hamilton-Jacobi PDE type. Let (M, μ) be a compact and connected Riemannian manifold endowed with a metric μ, and $f, b : (M, \mu) \longrightarrow \mathbb{R}$.

Definition 8. *The group morphological convolution* $\diamondsuit$ *between* b *and* f *is defined* $\forall\ x \in (M, \mu)$ *by:* $b \diamondsuit f(x) = \inf_{p \in G} \{f(\varphi_p(x_0)) + b(\varphi_{p^{-1}}(x))\}$.

Denote TM the tangent bundle (M, μ) and $L : TM \to \mathbb{R}$ a Lagrangian function. Let $H : T^*M \to \mathbb{R}$ be the Hamiltonian associated to the Lagrangian L, H is defined on the cotangent bundle T^*M of (M, μ), $H(x, q) = \sup_{v \in T_x M} \{q(v) - L(x, v)\}$. The Hamilton-Jacobi PDE can be extended in Riemannian manifolds as follows: $\partial_t w + H\,(x, \nabla w) = 0$ in $(M, \mu) \times (0, +\infty)$; $w(\cdot, 0) = f$ on (M, μ). Riemannian multiscale operations can be performed by choosing a specific Hamiltonian, respectively, $H = \|\nabla_\mu w\|_\mu^k$ for the multiscale dilations and $H = -\|\nabla_\mu w\|_\mu^k$ for multiscale erosions, and taking $k > 1$ allows to deal with more general structuring functions than the quadratic ones.

Proposition 2. *Let* $f \in C^0((M, \mu), \mathbb{R})$ *a continuous function and let* $c_k = \frac{k-1}{k^{\frac{k}{k-1}}}$, $k > 1$. *Viscosity solutions of the Cauchy problem:*

$$\frac{\partial w}{\partial t} + \|\nabla_\mu w\|_\mu^k = 0 \; \textit{in} \; (M, \mu) \times (0;\ \infty);\; w(\cdot\,, 0) = f \; \textit{on} \; (M, \mu), \tag{6}$$

are given by: $f_t(x) = b_t^k \diamondsuit f(x) := \inf_{h \in G} \left\{ f\big(\varphi_h(x_0)\big) + c_k \dfrac{d_\mu\big(\varphi_{h^{-1}}(x), x_0\big)^{\frac{k}{k-1}}}{t^{\frac{1}{k-1}}} \right\}$,

where $b_t^k = c_k \dfrac{d_\mu(x_0, \cdot\,)^{\frac{k}{k-1}}}{t^{\frac{1}{k-1}}}$ *are the multiscale structuring functions.*

Proof. Viscosity solutions of the PDE (6) are given by HLO formulas [11]: $f_t(x) = \inf_{y\in M}\left\{ f(y) + c_k \frac{d_\mu(x,y)^{\frac{k}{k-1}}}{t^{\frac{1}{k-1}}} \right\}$. The projection π (4) is defined by associating any $h \in G$ to an element $x \in (M,\mu)$. Then, using the definition and accounting the invariance property in Proposition 1, one gets:

$$\begin{aligned} f_t(x) &= \inf_{h\in G}\left\{ f\big(\varphi_h(x_0)\big) + c_k \frac{d_\mu(x, \varphi_h(x_0))^{\frac{k}{k-1}}}{t^{\frac{1}{k-1}}} \right\} \\ &= \inf_{h\in G}\left\{ f\big(\varphi_h(x_0)\big) + c_k \frac{d_\mu\big(\varphi_{h^{-1}}(x), x_0\big)^{\frac{k}{k-1}}}{t^{\frac{1}{k-1}}} \right\} \\ &= \inf_{h\in G}\left\{ f\big(\varphi_h(x_0)\big) + b_t^k\big(\varphi_{h^{-1}}(x)\big) \right\} = b_t^k \diamondsuit f(x). \end{aligned}$$

□

By reversing the time, we can prove that the viscosity solutions of the Cauchy problem corresponding to multiscale dilations:

$$\frac{\partial w}{\partial t} - \|\nabla_\mu w\|_\mu^k = 0 \text{ in } (M,\mu)\times(0;\ \infty);\ w(\cdot\,,0) = f \text{ on } (M,\mu) \tag{7}$$

are given by [11]: $f^t(x) = \sup_{x\in(M,\mu)}\left\{ f(y) - C_k \frac{d_\mu(x,y)^{\frac{k}{k-1}}}{t^{\frac{1}{k-1}}} \right\}$, and thus, using the same arguments as in the preceding proof, one has: $f_t(x) = -(b_t^k \diamondsuit (-f))(x)$.

Proposition 3. *Let $k > 1$. For all $t, s \geq 0$, the family of structuring functions b_t^k satisfy the following semigroup property: $b_{t+s}^k = b_t^k \diamondsuit b_s^k$.*

4 Morphological equivariant PDEs for generative models

We aim at proposing generative models for images that are based on PDEs satisfying an equivariance property. Our approach is resumed in two major steps: **1)** design of morphological PDEs in Riemannian manifolds akin to Section 3 as alternatives for introducing non-linearities in traditional CNNs that preserve an equivariant processing in the composition of the feature maps in layers, and **2)** proposition of a generative model based on this structure and classical GANs.

4.1 Morphological PDE-based layers

Feature maps are carried out in traditional CNNs throughout the classical convolution, pooling and ReLU activation functions. Our goal is to propose PDEs that behave like traditional CNNs, in one hand, and preserve an equivariance property, on the other hand. For that purpose, PDEs will be formulated on group transformations to ensure equivariance and make PDEs consistent with G-CNNs

[3,9,10]. Equivariance is a robust way to incorporate desired and essential symmetries into the network so that there is no more need to learn such symmetries; consequently, the amount of data is reduced. Viewing layers as image processing operators allows us use well elaborated image analysis and processing techniques to design the network. Thin image analysis is needed to achieve our objective. Due to its nonlinearity aspects, good shape and geometry description capabilities, mathematical morphology appeared as an efficient and powerful tool for multiscale image and data analysis [30]. For a better analysis of geometrical image structures, it is also interesting to consider works from geometric image analysis [5,13,15,17,34]. Image and data analysis and processing methods based on non-Euclidean metrics; for instance, Riemannian metrics, are well known to improve a lot Euclidean based approaches. Riemannian manifolds are proved to behave very well for capturing thin data structures, providing then better representations and analysis of geometrical structures present in the data. This fact is shown in many image processing studies with real life applications; for instance, in video surveillance, shape and surface analysis, human body and face analysis, image segmentation [2,7,26,27,32,35]. For these reasons, we choose homogeneous spaces to avoid Euclidean metrics so that the network is provided with image processing capabilities for a better handling of geometric thin structures [8,11,14,18,25]. Doing so should make feature maps richer, and combined with the equivariance property of the morphological PDEs will provide neat improvements of classical GANs in terms of quality of the content generation. Morphological PDEs are thus used to replace the pooling operations and ReLU activation functions in the proposed generative model.

4.2 PDE model design

PDE-G-CNNs were formally introduced in homogeneous spaces with G-invariance metric tensor fields on quotient spaces [31]. Built on the primary approach, the proposed model is based on a combination of traditional CNNs and morphological PDE layers of Hamilton-Jacobi type in Riemannian manifolds, and is composed of the following PDEs:

- Convection: $\dfrac{\partial w}{\partial t} + \alpha w = 0$ in $(\mathcal{M}, \mu) \times (0,\ \infty)$; $w(\cdot, 0) = f$ on $(\mathcal{M}, \mu)$.
- Diffusion: $\dfrac{\partial w}{\partial t} + (-\Delta_\mu)^{k/2} w = 0$ in $(\mathcal{M}, \mu) \times (0,\ \infty)$; $w(\cdot, 0) = f$ on $(\mathcal{M}, \mu)$.
- Morphological multiscale erosions and dilations for $(+)$ and $(-)$ sign:

$$\frac{\partial w}{\partial t} \pm \|\nabla_\mu w\|_\mu^k = 0 \text{ in } (\mathcal{M}, \mu) \times (0,\ \infty);\ w(\cdot, 0) = f \text{ on } (\mathcal{M}, \mu), \tag{8}$$

where α a is vector field invariant under G on $(\mathcal{M}, \mu)$, Δ_μ represents the Laplace-Beltrami operator, $\|\cdot\|_\mu$ the norm induced by the Riemannian metric μ and $k > 1$. The above system of PDEs consitutes the PDE model solved in a step basis using the operator splitting method, where each step corresponds to one of the PDEs. In this work, we only use the morphological multiscale operations steps (8), the convection and diffusion terms are left for future work. PDEs (8) introduce nonlinearities into the generator network of the GM-GAN using morphological convolutions, which are obtained a viscosity sense and given respectively for multiscale dilations and erosions thanks to Proposition 2.

Proposition 4. *Let $f \in C^{\infty}((\mathcal{M},\mu))$ and $B \subset (\mathcal{M},\mu)$ an non-empty set. Consider the flat structuring function $b : (\mathcal{M},\mu) \to \mathbb{R} \cup \{\infty\}$. Then, one has:* $-\left(b \diamondsuit (-f)\right)(x) = \sup_{\substack{h \in G \\ \varphi_{h^{-1}}(x) \in B}} f\left(\varphi_h(x_0)\right)$.

The max pooling of function f with motif B can in fact be seen as a flat morphological dilation with a structurant element B. It is truly the case for example for $\mathbb{R}^n$. Indeed, for $f \in C^0(\mathbb{R}^n)$ and $B \subset \mathbb{R}^n$ a compact set, for every $x \in \mathbb{R}^n$, one has:$-\left(b \diamondsuit_{\mathbb{R}^n} (-f)\right)(x) = \sup_{y \in B} f(x-y)$, where the right hand side is in fact a flat dilation with a structurant element B.

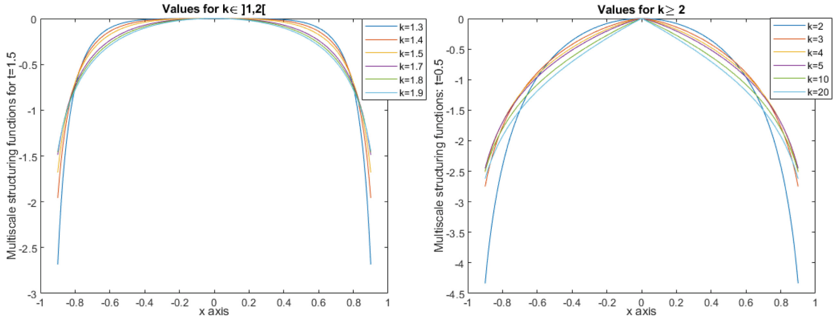
Proposition 5. *Let $f \in C^0_c((\mathcal{M},\mu))$. Morphological dilation with the following structuring function: $b(x) = 0$, if $x = x_0$; and $b(x) = \sup_{x \in \mathcal{M}} f(x)$, otherwise, is exactly the same as applying a ReLU to f:* $-\left(b \diamondsuit (-f)\right)(x) = \max\{0, f(x)\}$.

4.3 Architecture of morphological equivariant PDEs based on GAN

Similarly to GAN, the proposed geometric morphological GAN (GM-GAN) is composed of two networks: a generator (G) and a discriminator (D) which are both multi-layer perceptrons. As detailed in the preceding section, we introduce into the network G morphological PDE-based layers through the resolution in a step basis of Hamilton-Jacobi PDEs (8), whose viscosity solutions are given for multiscale erosions and dilations thanks to Proposition 2. To deal with computation issues and practical implementation of the proposed framework, we take advantage of the geometric properties of hyperbolic spaces and generate various and rich content on data with multiple transformations. For doing so, we provide the distance d_μ in the geodesic ball by considering the hyperbolic ball $B = \{(x_1, x_2) \in \mathbb{R}^2$ such that $x_1^2 + x_2^2 < 1\}$, which is endowed with the metric $\mu = \dfrac{4(\mathrm{d}x_1^2 + \mathrm{d}x_2^2)}{(1 - \|x\|^2)^2}$, where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^2$. The distance is obtained as follows: $d_\mu(x,y) = \operatorname{Argcosh}\left(1 + \dfrac{2\|x-y\|^2}{(1-\|x\|^2)(1-\|y\|^2)}\right)$.

Concave structuring functions $b_t^k = c_k \dfrac{d_\mu(x_0,\cdot)^{\frac{k}{k-1}}}{t^{\frac{1}{k-1}}}$ are represented in Fig. 1 for different values of t and k in $]-1;\ 1[$.

GM-GAN training procedure remains the same as in traditional GANs. Specifically, the training procedure is carried out separately but simultaneously. The model takes as input some noise z defined with a prior probability p_z, and then, attempts to learn the distribution of the generator p_g, by representing a function $G(z;\theta_g)$ from z to the data space. The discriminator network D takes



(a) (b)

Fig. 1. $b_t^k(x)$, $x \in]-1;\ 1[$: (a) for $t = 1.5$ and $k \in]1;\ 2[$. (b) for $t = 0.5$ and $k \geq 2$

an input image x and finds a function $D(x; \theta_d)$ from x to a single scalar, which is the probability that the image x comes from p_{data} which defines the origin of the sampled images. The output of the D network returns a value close to 1 if x is a real image from p_{data}, and a value very close to 0 if x comes from p_g; otherwise. The main objective of network D is to maximize $D(x)$ for an image coming from the true data distribution p_{data}, while minimizing $D(x) = D(G(z; \theta_g))$ for images generated from p_z and not from p_{data}. The objective of the generator G is to deceive the D network, meaning to maximize $D(G(z; \theta_g))$. This is equivalent to minimize $1 - D(G(z; \theta_g))$ as D is a binary classifier. This conflict between these objectives is called the minimax game and formulated as follows: $\min \max E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z; \theta_g)))]$. The case $p_g = p_{data}$ corresponds to the global optimum of the minimax game. Main contributions of the proposed GM-GAN rely on the equivariance property and non linearity characteristics brought out by group morphological convolutions and their ability to extract thin geometrical features, which lead to richer feature maps and a reduction of the amount training data.

For the GM-GAN generator, let x be the input data into the morphological layer called *Morphoblock*. Then, x goes first through a multiscale morphological erosion operation, followed by a multiscale morphological dilation. Afterwards, both erosion and dilation are followed by a linear convolution. The output of the PDE layer is obtained by a linear combination of the two outputs. The overall architecture of the GM-GAN generator is illustrated in Fig. 2.

5 Numerical experiments

GM-GAN and GAN are applied to MNIST dataset. MNIST database consists of 70,000 black-and-white 28x28 images that represent handwritten digits from 0 to 9. It is divided into a training set of 60,000 images and a test set of 10,000 images. Same training parameters are set for GM-GAN and GAN: number of epochs to 200, the batch size to 64, the latent space dimensionality to 100, and

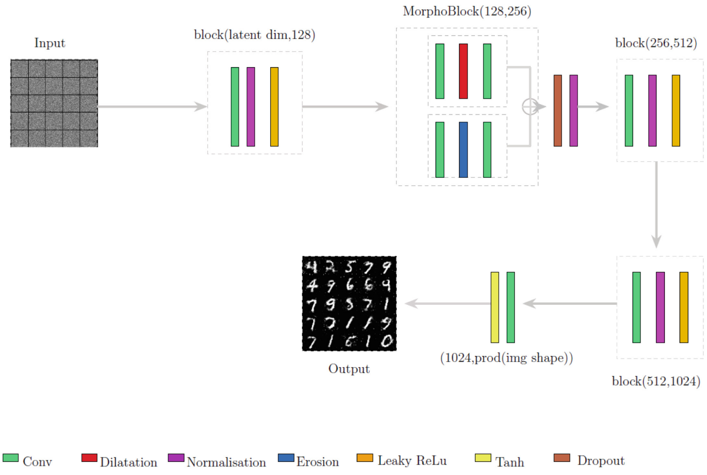
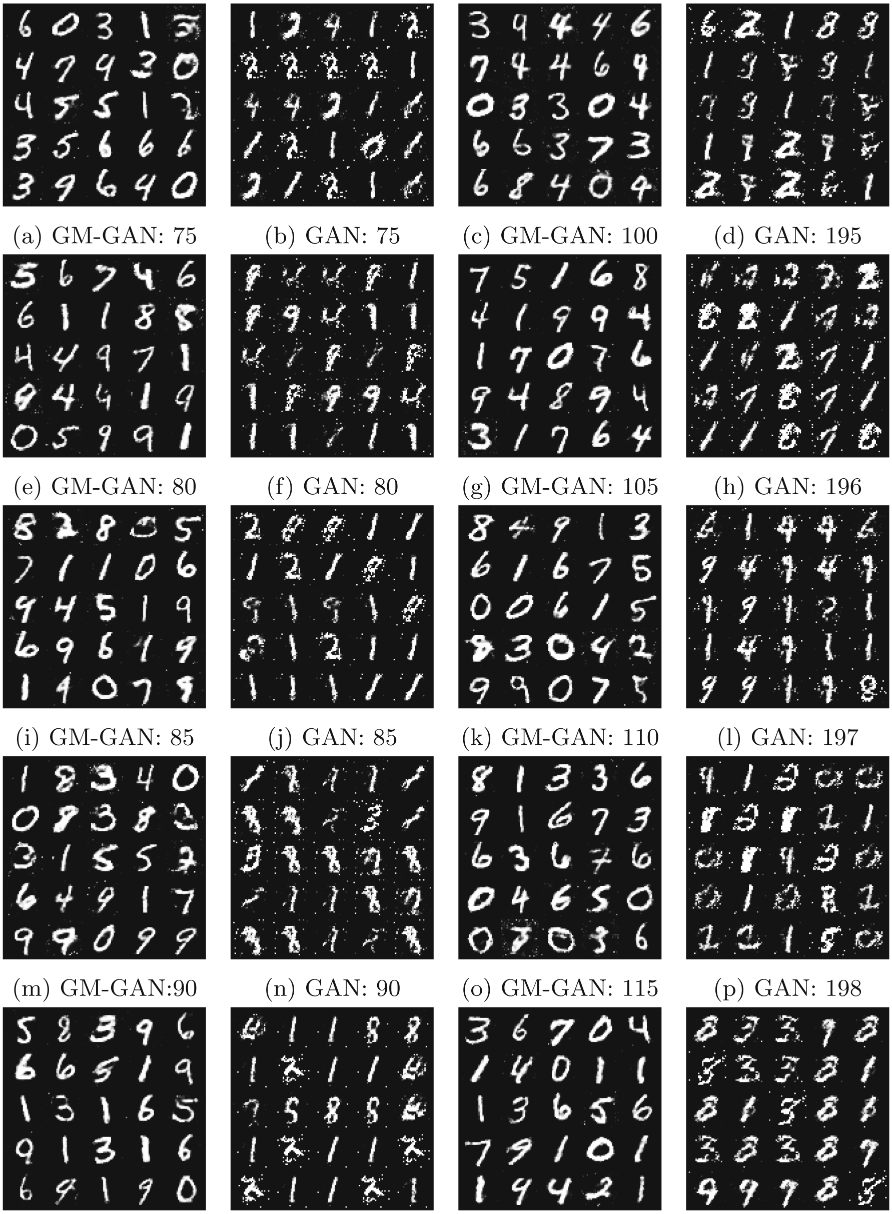


Fig. 2. Architecture of GM-GAN generator

the interval between image samples to 400. Generated images with GM-GAN and GAN are displayed in Fig. 3 showing higher generation quality with GM-GAN in comparison to traditional GAN.

This can be seen by comparing images produced at epochs 70 to 95 with GM-GAN (Figs. 3a, 3e, 3i, 3m and 3q) and those generated with GAN at same epochs (Figs. 3b, 3f, 3j, 3n and 3r). For instance, some digits are clearly identifiable with GM-GAN based generation, whereas it is almost impossible to recognize the digits with GAN based ones. We also observe that the images generated with GM-GAN at epochs going from 100 to 120 (Figs. 3c, 3g, 3k, 3o and 3s) are of better quality than generated ones with GAN for the last five epochs going from epoch 195 to 199 (Figs. 3d, 3h, 3l, 3p and 3t). To better discriminate that fact, we zoom in on some areas in images generated at epochs 85, 92 and 96 (Figs. 4-(a)-(b), (c)-(d) and (e)-(f); respectively), and highlight the realistic variations between the generated images of the same digit. This indicates that GM-GAN has a deeper understanding of the sample characteristics and is capable of generalizing them beyond the specific examples they are trained on. This can be observed in Fig. 4-(b) with digits 3 and 6, in Fig. 4-(d) with digits 2 and 8, and in Fig. 4-(f) with digits 9 and 7.

GM-GAN complexity is also reduced throughout the equivariance property by eliminating the need to learn symmetries. This is illustrated by reducing MNIST training dataset by a half and comparing generated images at epoch 42. GM-GAN results (Fig. 5a show again better image quality and high variations of generated digits in comparison to GAN (Fig. 5b. Results highlight the importance of equivariance in morphological operators, turning out to dataset



(a) GM-GAN: 75 (b) GAN: 75 (c) GM-GAN: 100 (d) GAN: 195

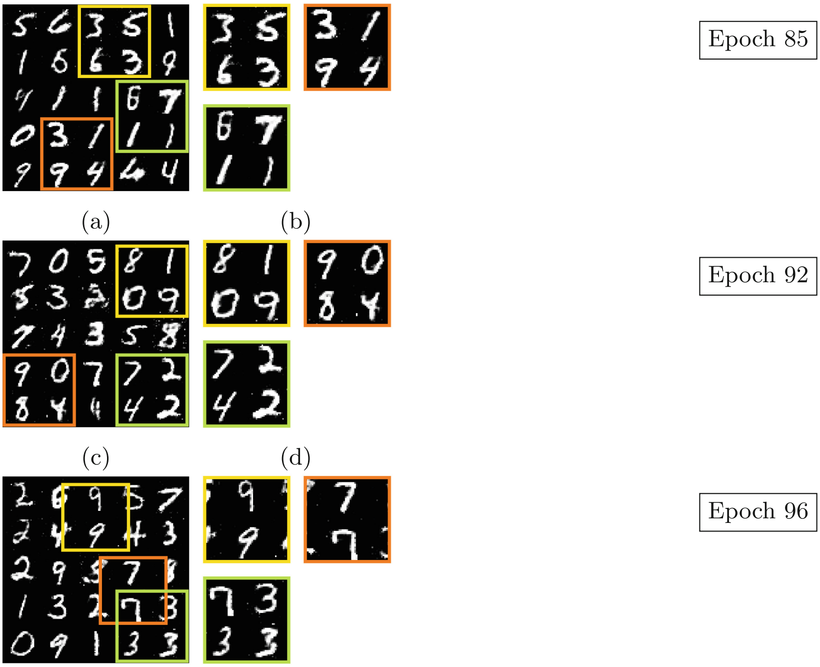
(e) GM-GAN: 80 (f) GAN: 80 (g) GM-GAN: 105 (h) GAN: 196

(i) GM-GAN: 85 (j) GAN: 85 (k) GM-GAN: 110 (l) GAN: 197

(m) GM-GAN:90 (n) GAN: 90 (o) GM-GAN: 115 (p) GAN: 198

(q) GM-GAN: 95 (r) GAN: 95 (s) GM-GAN: 120 (t) GAN: 199

Fig. 3. Image generation using MNIST: GM-GAN vs. GAN

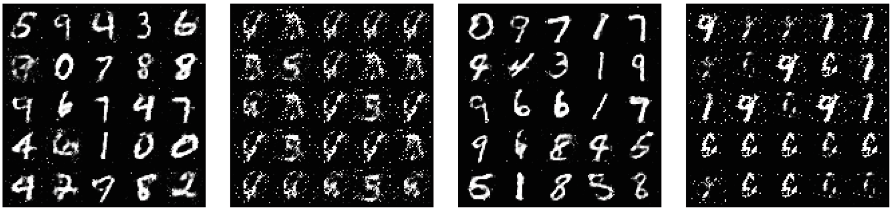


(a) (b)

(c) (d)

(e) (f)

Fig. 4. Zoom in on images generated with GM-GAN at different epochs

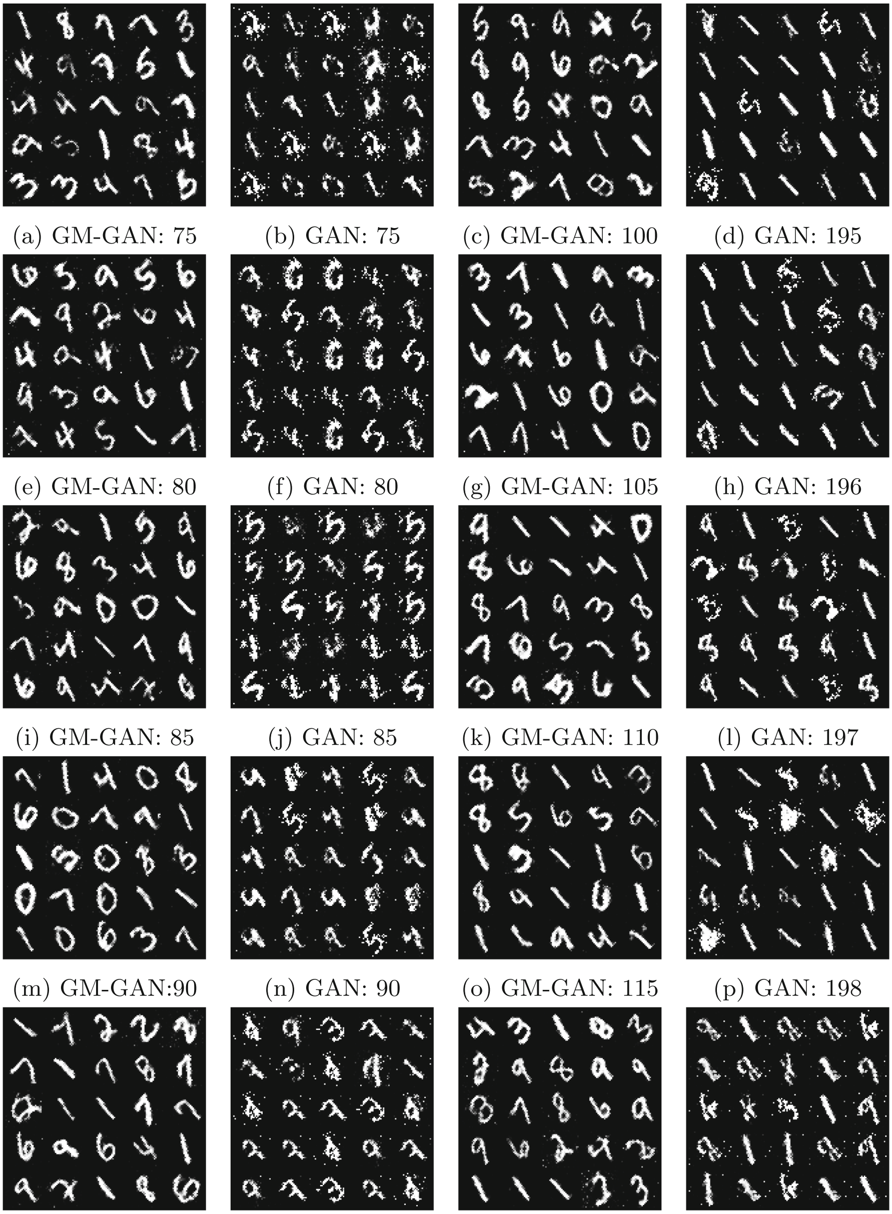


(a) GM-GAN(1/2) (b) GAN(1/2) (c) GM-GAN (d) GAN

Fig. 5. GM-GAN vs. GAN at epoch 42 with half (1/2) and whole MNIST dataset

reduction without significantly impacting generation results (see Fig. 5c or GM-GAN and Fig. 5d for images generated at the same epoch using the hole dataset).

To highlight again the usefulness of morphological equivariant operators, we apply both GM-GAN and GAN models on RotoMNIST; generated images are displayed in Fig. 6. It can be seen in results obtained with GM-GAN from epoch 70 to 95 (Figs. 6a, 6e, 6i, 6m, and 6q) that digits are clearly identifiable and far better than those generated with GAN at the same epochs (Figs. 6b, 6f, 6j, 6n, and 6r) where digits are barely formed. The same is noticed with GM-GAN



(a) GM-GAN: 75 (b) GAN: 75 (c) GM-GAN: 100 (d) GAN: 195

(e) GM-GAN: 80 (f) GAN: 80 (g) GM-GAN: 105 (h) GAN: 196

(i) GM-GAN: 85 (j) GAN: 85 (k) GM-GAN: 110 (l) GAN: 197

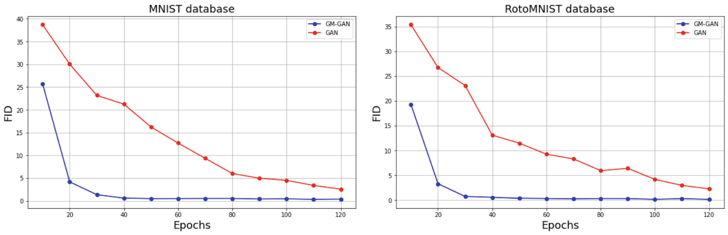
(m) GM-GAN:90 (n) GAN: 90 (o) GM-GAN: 115 (p) GAN: 198

(q) GM-GAN: 95 (r) GAN: 95 (s) GM-GAN: 120 (t) GAN: 199

Fig. 6. Image generation using RotoMNIST: GM-GAN vs. GAN

from epoch 100 to 120 (Figs. 6c, 6j, 6k, 6o, and 6s), in comparison with GAN for the last 5 epochs (Figs. 6d, 6h, 6l, 6p, and 6t). This demonstrates that GM-GAN is more suitable for data under rotation transformations, and highlights one more time the importance of equivariance for generating satisfactory results under various transformations.

Quantitative evaluations are provided using the Frëchet Inception Distance (FID). A low FID indicates a high similarity between generated and real data, corresponding to good generation quality. In Fig. 7, we present the FID curves of both models over epochs (taking FID of generated images at intervals of 10 epochs) on both MNIST and RotoMNIST datasets. It can be seen that starting from epoch 40, FIDs of GM-GAN generated results are significantly lower than ones generated using GAN, which confirms the qualitative results discussed just above.



(a) (b)

Fig. 7. FID using GM-GAN vs. GAN with: (a) MNIST. (b) RotoMNIST

6 Conclusion and perspectives

We have proposed here a geometric generative GM-GAN model based on PDE-G-CNNs and built from derived equivariant morphological operators and geometric image processing techniques. The proposed equivariant morphological PDE layers are composed of multiscale dilations and erosions without any need to approximate convolutions kernels, and meanwhile, group symmetries are defined on Lie groups allowing a geometrical interpretability of GM-GAN with left invariance properties. As shown by preliminary results on MNIST and RotoMNIST datasets, preliminary qualitative and quantitative results show noticeable improvements compared classical GAN. Indeed, thin image features are better extracted by accounting intrinsic geometric features at multiscale levels, and the network complexity is reduced. The proposed approach can be extended to various generative models. Future works include applying GM-GAN on other datasets, designing fully equivariant generative models entirely based on PDE-G-CNNs, and studying GM-GAN complexity to demonstrate the computational advantages of the proposed model over classical GAN.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223. PMLR (2017)
2. Balan, V., Stojanov, J.: Finslerian-type GAF extensions of the riemannian framework in digital image processing. Filomat **29**(3), 535–543 (2015)
3. Bekkers, E.J., Lafarge, M.W., Veta, M., Eppenhof, K.A., Pluim, J.P., Duits, R.: Roto-translation covariant convolutional networks for medical image analysis. In: Medical Image Computing and Computer Assisted Intervention - MICCAI 2018: 21st International Conference. Proceedings, Part I, pp. 440–448. Granada, Spain (Sep (2018)
4. Bellaard, G., Bon, D.L., Pai, G., Smets, B.M., Duits, R.: Analysis of (sub-)Riemannian PDE-G-CNNs. Journal of Mathematical Imaging and Vision pp. 1–25 (2023)
5. Burger, M., Sawatzky, A., Steidl, G.: First order algorithms in variational image processing. Springer (2016)
6. Chen, M., Denoyer, L.: Multi-view generative adversarial networks. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 10. pp. 175–188. Springer (2017)
7. Citti, G., Franceschiello, B., Sanguinetti, G., Sarti, A.: Sub-riemannian mean curvature flow for image processing. SIAM Journal on Imaging Sciences **9**(1), 212–237 (jan 2016)
8. Citti, G., Sarti, A.: A cortical based model of perceptual completion in the roto-translation space. Journal of Mathematical Imaging and Vision **24**, 307–326 (2006)
9. Cohen, T., Welling, M.: Group Equivariant Convolutional Networks. In: International conference on machine learning. pp. 2990–2999. PMLR (2016)
10. Cohen, T.S., Geiger, M., Weiler, M.: A general theory of equivariant cnns on homogeneous spaces. Advances in neural information processing systems **32** (2019)
11. Diop, E.H.S., Mbengue, A., Manga, B., Seck, D.: Extension of Mathematical Morphology in Riemannian Spaces. In: Lecture Notes in Computer Science, pp. 100–111. Springer International Publishing (2021)
12. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. arXiv preprint arXiv:1605.09782 (2016)
13. Dubrovina-Karni, A., Rosman, G., Kimmel, R.: Multi-region active contours with a single level set function. IEEE Trans. Pattern Anal. Mach. Intell. **37**(8), 1585–1601 (2014)
14. Duits, R., Bekkers, E.J., Mashtakov, A.: Fourier transform on the homogeneous space of 3D positions and orientations for exact solutions to linear PDEs. Entropy **21**(1), 38 (2019)
15. Duits, R., Burgeth, B.: Scale spaces on Lie groups. In: International Conference on Scale Space and Variational Methods in Computer Vision. pp. 300–312 (2007)
16. Durugkar, I., Gemp, I., Mahadevan, S.: Generative multi-adversarial networks. arXiv preprint arXiv:1611.01673 (2016)
17. Fadili, J., Kutyniok, G., Peyré, G., Plonka-Hoch, G., Steidl, G.: Guest editorial: mathematics and image analysis. Journal of Mathematical Imaging and Vision **52**, 315–316 (2015)
18. Franceschiello, B., Mashtakov, A., Citti, G., Sarti, A.: Geometrical optical illusion via sub-Riemannian geodesics in the roto-translation group. Differential Geom. Appl. **65**, 55–77 (2019)

19. Gauthier, J.: Conditional generative adversarial nets for convolutional face generation. Class project for Stanford CS231N: convolutional neural networks for visual recognition, Winter semester **2014**(5), 2 (2014)
20. Gerken, J.E., Aronsson, J., Carlsson, O., Linander, H., Ohlsson, F., Petersson, C., Persson, D.: Geometric deep learning and equivariant neural networks. Artif. Intell. Rev. **56**(12), 14605–14662 (2023)
21. Goodfellow, I.: Generative Adversarial Networks. In: NIPS. p. 57 (2017)
22. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
23. Recent Advances in Convolutional Neural Networks: Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., TingLiu, Wang, X., Wang, L., Wang, G., Cai, J., Chen, T. Pattern Recogn. **77**, 354–377 (2018)
24. Im, D.J., Kim, C.D., Jiang, H., Memisevic, R.: Generating images with recurrent adversarial networks. arXiv preprint arXiv:1602.05110 (2016)
25. Janssen, M.H., Janssen, A.J., Bekkers, E.J., Bescós, J.O., Duits, R.: Design and processing of invertible orientation scores of 3D images. Journal of mathematical imaging and vision **60**, 1427–1458 (2018)
26. Kurtek, S., Jermyn, I.H., Xie, Q., Klassen, E., Laga, H.: Elastic shape analysis of surfaces and images. In: Riemannian Computing in Computer Vision, pp. 257–277. Springer International Publishing (2016)
27. Pierson, E., Daoudi, M., Tumpach, A.B.: A Riemannian Framework for Analysis of Human Body Surface. In: IEEE/CVF Winter Conference on Applications of Computer Vision. WACV, pp. 2763–2772. Waikoloa, HI, USA (Jan (2022)
28. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
29. Romero, D., Bekkers, E., Tomczak, J., Hoogendoorn, M.: Attentive Group Equivariant Convolutional Networks. In: Proceedings of Machine Learning Research. pp. 8188–8199 (2020)
30. Shih, F.: Image processing and mathematical morphology: fundamentals and applications. CRC Press, Boca Raton (2009)
31. Smets, B.M.N., Portegies, J., Bekkers, E.J., Duits, R.: PDE-Based Group Equivariant Convolutional Neural Networks. Journal of Mathematical Imaging and Vision **65**(1), 209–239 (2022)
32. Su, J., Kurtek, S., Klassen, E., Srivastava, A.: Statistical analysis of trajectories on Riemannian manifolds: Bird migration, hurricane tracking and video surveillance. The Annals of Applied Statistics **8**(1) (Mar 2014)
33. Tian, C., Zhang, Y., Zuo, W., Lin, C.W., Zhang, D., Yuan, Y.: A Heterogeneous Group CNN for Image Super-Resolution. IEEE Transactions on Neural Networks and Learning Systems pp. 1–13 (2024)
34. Welk, M., Weickert, J.: Pde evolutions for m-smoothers: from common myths to robust numerics. In: International Conference on Scale Space and Variational Methods in Computer Vision. pp. 236–248. Springer (2019)
35. Younes, L.: Shapes and Diffeomorphisms. Springer, Berlin Heidelberg (2019)

NR-CION: Non-rigid Consistent Image Composition Via Diffusion Model

Wei Liu(✉), Liuan Wang, and Jun Sun

Fujitsu R&D Center, Co., LTD., Beijing, China
liuwei@fujitsu.com

Abstract. Text guided image diffusion model has demonstrated remarkable ability in consistent image generation. In this paper, we introduce a training free image composition framework that realizes the non-rigid objects composition based on a pair of source and target prompts. Specifically, we aim at blending the user provided object reference image into the background image in a non-rigid manner and keep the balance of fidelity and editability. For example, we can make a standing dog jumping while preserving its shape and appearance under the guidance of target prompt. Our proposed method has three key components: firstly, the reference image and background are inverted into latent noises with different image inversion methods. Secondly, we guarantee the consistent image attribute generation of the reference object by injecting the self-attention key and value features from original pipeline in sampling steps. Thirdly, we iteratively optimize the object mask in the target pipeline, and progressively compose image in different regions. Experiments shows that our proposed method can achieve the non-rigid object image editing and seamless composition, the results are impressive in consistent and editable image composition.

Keywords: Image composition · diffusion model · non-rigid image generation · self-attention · cross-attention

1 Introduction

Recently, text-based image generation by diffusion model[25,26] has achieve great advance, which has the capability to generate promising and fantastic image conforming with user provided prompt. Meanwhile, the text based image editing[3,5,11,13] is a promising direction. Some works focus on the consistent image generation[7,11,22] task, which aim to edit the local attributes of the object, such as the image styles, the local object color, etc. Some works intend to generate different views or more complex non-rigid images such as raising hand. Meanwhile, they maintain the context and shape information of the object. The balance of editability and fidelity is very important and practical. These capabilities are useful for comics, advertisement design or video generation applications.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 326–338, 2025.
https://doi.org/10.1007/978-3-031-78389-0_22

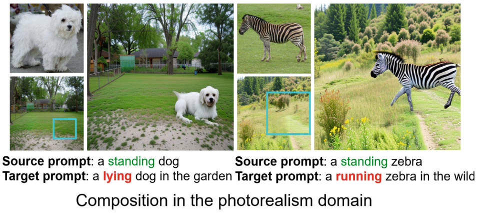

Source prompt: a standing dog
Target prompt: a lying dog in the garden
Source prompt: a standing zebra
Target prompt: a running zebra in the wild
Composition in the photorealism domain

Fig. 1. Our Non-rigid consistent image composition method aims to blend the reference foreground object to the background image seamlessly in a non-rigid manner

Although these works have achieved good performance in text-based image editing tasks, their edited image only contains the foreground object which is not practical for real application and scenario (Fig. 1).

Image composition[3,4,17,19] task refers to blending a specific object into the main background image seamlessly and harmonically. For instance, a standing dog image is desired to incorporate into the park image full of grasses, how to naturally blend the dog to the park image is a challenge task. Traditional works focus on blending the static standing dog into the park which may not satisfy the real various applications' needs. In comics and advertisement industry, the of customers' demands are various, they may want the dog looks like jumping. Editing the reference image by customers' requirements is more practical. Here we raise a question: How to generate a running dog that has the same appearance and blend it into the garden background image harmonically?

To answer the above question, in this paper, we propose the Non-Rigid Consistent Image compositiON framework (NR-CION), which has the capability to realize non-rigid object image generation by the reference image and user provided prompt, and accomplish the image composition with background. The framework is performed in a training free manner and does not need any further fine tuning or optimization process. The framework accomplishes the non-rigid image editing and object image composition simultaneously, and achieves promising performance. The generated image maintains context and shape information while achieving the non-rigid image editing. Our proposed method is based on the theoretical DDIM inversion[28] and image reconstruction, the reference and background image are inverted into latent noises with different image inversion methods. In sampling steps, we design two pipelines for denoising reference latent noises. Denoising pipeline D1 denoises reference latent noises with null text inversion. Denoising pipeline D2 denoises reference latent noises with target prompt. We iteratively optimize the foreground mask with cross attention in the D2 pipeline under the guidance of target prompt, and progressively compose image in different regions. To guarantee the consistent attribute generation of the reference object, the self-attention key and value features of D2 are injected from source pipeline D1. To balance the consistency and editability, we control the starting time step of feature injection.

The key question is to guarantee the accurate mask extraction of the foreground in the reference latent map. To this end, we optimize the mask obtained from cross-attention by Ground-SAM [15,18] in the latent map. The optimized mask can not only enhance the consistent foreground image reconstruction, but also accurately be blended with the background image.

Our contribution can be summarized as follows:

- We propose a training free non-rigid consistent image composition framework short as NR-CION to perform the non-rigid image composition task.
- We iteratively optimize the object mask accurately with cross attention and a pretrained segmentation model which can guarantee the consistent image editing and seamlessly image composition.
- Our experiments demonstrate the effectiveness of our proposed NR-CION in non-rigid consistent image editing and seamless image composition.

2 Related work

2.1 Text-based image editing

Text-based image editing is a fundamental and challenge task that employs extra textual prompt to manipulate the source images. GAN-based image editing[1,9,16] has been carried out extensively. Recently, text-based image manipulation with diffusion model has drawn more attention, lots of works have achieved the state-of-the-art performance. SD-Edit[20] is the first attempt to utilize diffusion model for image editing task which demonstrate the powerful capability, Prompt to prompt[11] is the foundation work which use the cross attention map to preserve the structure or spatial layout, thus accomplishes the image editing based on text prompt only. Limited to the image inversion methods, prompt to prompt is limited to image synthesis. DiffEdit[8] employ a caption and a query to compute the mask during diffusion process and perform object replacement with the mask guidance. Imagic[13] shows impressive image editing performance which is based on Imagen[27]. Null text inversion[22] and Prompt Tuning Inversion[10] employ an embedding to optimize the difference between sampling and image inversion process. Masa-Ctrl[7] further develops a framework that utilizes a combination of self-attention and cross-attention for wide image editing applications, Direct Inversion[12] decouples the preservation and editing branches to realize the balance of fidelity and editability.

2.2 Image composition

Image composition blends a foreground region from one image to another background image to generate the realistic composition image. Image composition has a wide range of applications such as data augmentation, entertainment, E-commercial, advertising, artist creation etc. Image composition can be decomposed into multiple sub-tasks such as object placement, image blending, image harmonization, shadow generation and so on. In image blending task, traditional methods[6,23] target to smooth the transition from foreground to background. Another group of methods[29,30] aim to obtain smooth transition by enhancing gradient domain consistency. Recent deep learning based works[14,31] introduce learnable image blending to generate a seamless composition image. With the advent of the diffusion model, text-guided image composition[2,3] has became a new research direction. TF-ICON[19] leverages pre-trained diffusion models to conduct training free cross domain text and image guided image composition.

2.3 Image inversion

DDIM inversion[28] works well for unconditional diffusion model. It is found that it lacks in text guided diffusion model when classifier free guidance. Classifier free guidance will enlarge the reconstruction error. Adding noises to the input image with DDIM inversion[28] to obtain a latent noise, and denoising in the sampling process to reconstruct the input image, lots of works struggle to preserve the context, shape and layout information. Null text inversion[22] introduces a null text embedding to optimize the reconstruction step by step with MSE loss. Prompt tuning inversion[10] employs a learnable embedding and interpolates with the target embedding to achieve minimized reconstruction error, Negative prompt inversion[21] achieves the equivalent reconstruction effect without optimization, Direct inversion[12] designs a decoupled preservation and editing branches to realize image editing while preserving image consistency.

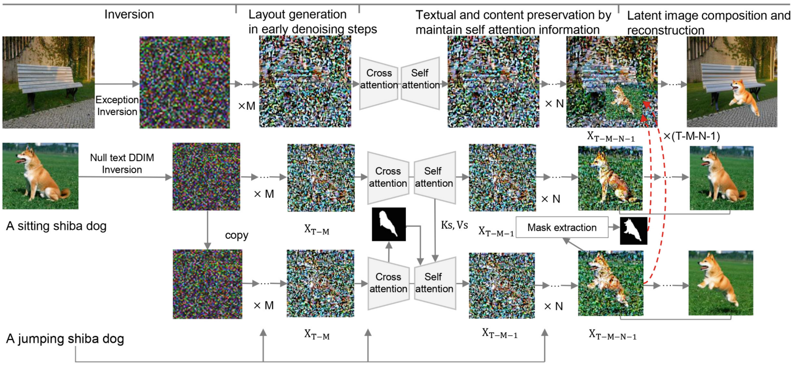


Fig. 2. An overview of our proposed NR-CION, the target is to generate non-rigid reference image editing and perform image composition with the background image. In the reference latent sampling process, the layout and structure information are generated under the guidance of target prompt in early denoising steps, then the target pipeline query from the source pipeline by self-attention features to maintain the texture information, finally the foreground latent map is blended with the main latent map guided by the foreground mask extracted. After the third stage denoising steps, the Synthesized composition image is generated

3 Preliminary

3.1 Latent diffusion model

Diffusion model has achieved great success in image generation, latent diffusion model[26] performs diffusion process in the latent space which reduces the diffusion complexity. In the latent space, the latent noise z_t can be obtained by adding a series of gaussian noises ϵ to the input image step by step like the following equation:

$$z_t = \sqrt{\alpha_t} z_0 + \sqrt{1 - \alpha_t} \epsilon \tag{1}$$

To reconstruct the image from z_t, the deterministic DDIM sampling[28] is employed:

$$x_{t-1} = \sqrt{\alpha_{t-1}} f_\theta(x_t, t) + \sqrt{1 - \alpha_{t-1}} \epsilon_\theta(x_t, t) \tag{2}$$

During the reconstruction process, the optimization objective is as follows:

$$L_{diffusion} = E_{x, \epsilon \sim N(0,1), t}[||\epsilon - \epsilon_\theta(x_t, t)||_2^2] \tag{3}$$

3.2 Classifier free guidance

In classifier free guidance, the prediction is performed firstly in an unconditional prediction manner, then is extrapolated with the conditional prediction results. This process will amplify the effects of text guidance. The classifier free guidance will magnify the accumulated error during the DDIM sampling process in text guided image editing.

3.3 Attention mechanism

In U-Net module of stable diffusion[26], attention mechanism plays an important role in generating diverse and contextual images by the user's input prompt. As for image editing task, the target prompt embedding is injected into the cross-attention layers, which controls the image generation. Cross-Attention can also help to obtain a mask related to the prompt in the latent space. Self-attention layers help the model focus more on texture information, controlling the self-attention layers allows the model to preserve the appearance and texture information in image editing generation. The attention mechanism can be depicted as follows:

$$Atten(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})V \tag{4}$$

4 Method

Given a background image, a reference foreground image, a source and target prompt, and a bounding box mask defining the location to place the foreground image in the background image, our objective is to blend the foreground object into the background seamlessly in a non-rigid manner. We adopt a training free text-image diffusion model to realize the image composition task. In the output image I^*, the background should maintain the original background appearance, and the foreground object should preserve the textual and shape information. The target prompt describes the change of the object such as postures, which controls the object image generation in a non-rigid way. The whole process involves three modules: background and foreground image inversion in section 4.1, non-rigid foreground object generation and mask generation in section 4.2 and image composition in section 4.3.

4.1 Image inversion

Diffusion based image editing methods invert edited image into latent space, but deviation exists when reconstructing latent noises back to target image by target prompts. To overcome the reconstruction error, we adopt different inversion strategies. As shown in Fig. 2, We adopt DDIM with exceptional prompt in[19] for background image inversion, which will alleviate the reconstruction error in classifier free guidance. In classifier free guidance such as DDIM, the exceptional prompt is effect to reduce reconstruction error under text guidance. The exception prompt is only performed in image inversion process but not in sampling process. As for foreground image inversion process, we utilize the deterministic DDIM with null prompt to invert the whole foreground image into latent noises. For the target prompt image generation pipeline, the latent noises is initialized by copying from the original inverted noises.

4.2 Non-rigid foreground object generation and mask generation

Our original idea is that we found in early sampling stage, the layout and structure information is reconstructed under the prompt guidance as shown in Fig. 3. Based on this observation, we adopt the three stage reconstruction procedures. As depicted in Fig. 2, we reconstruct the layout and structure of the foreground object in the first stage which occupies M denoising time steps, In each denoising step, the target prompt embedding is injected to the cross attention layers to control the image generation. Then the texture and appearance information are featured in the second stage. Finally, the latent image composition and composed image reconstruction will be performed in the third stage.

Based on the founding of existing work[11], the tokens in the target prompt can be reflected in the attention map, the desired foreground mask can be roughly obtained by the attention map. Specifically, the attention mask can be obtained by calculating the average attention map in the early denoising time steps. At the end of the second stage, the rough average attention map can be refined by a

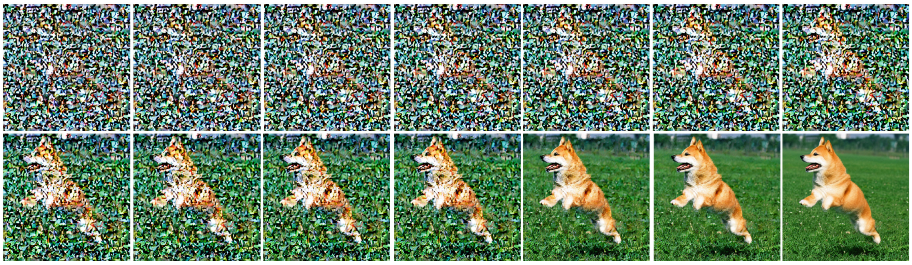


Fig. 3. Target guided intermediate results in Sampling process

SOTA segmenter Ground-SAM together with the target prompt to acquire the accurate foreground mask. The obtained accurate mask can be utilized for the final image composition in the third denoising stage.

```
Algorithm 1: Non-Rigid Consistent Image composition
Input: The background image $I_m$, the reference image $I_r$
source prompt $P_s$, target prompt $P_t$, foreground mask $M_{bbx}$
Output: The edited latent composition map $I_{m_*}$
for t=1,2,...T do
    $x_t^m \leftarrow DDIM(x_{t-1}^m, t-1, P_{excep})$
    $x_t^m \leftarrow DDIM(x_{t-1}^r, t-1, P_{null})$
end for
Sampling process
for t=T,T-1,...1 do
    $x_{t-1}^r, \{Q_s, K_s, V_s\} \leftarrow DDIM(x_t^r, t, P_{null})$
    if t$\geq$ T-M do
        $x_{t-1}^m \leftarrow DDIM(x_t^m, t, P_{excep})$
        $x_{t-1}^{r^*}, \{Q_t, K_t, V_t\} \leftarrow DDIM(x_t^{r^*}, t, P_{target})$
    end if
    if t<T-M and t $\geq$ T-M-N do
        $x_{t-1}^m \leftarrow DDIM(x_t^m, t, P_{excep})$
        $A_t^* = Q_t, K_s, V_s$
        $x_{t-1}^{r^*} \leftarrow DDIM(x_t^{r^*}, t, P_{target}, A_t^*)$
    end if
    if t==T-M-N-1 do
        $M^{seg} = MaskExtract(x_t^{r^*})$
        $H_t^* = H_t^m \odot (1 - M^{seg}) + H_t^r \odot M^{seg}$
    end if
    if t<T-M-N-1 do
        $x_{t-1}^{m^*} \leftarrow DDIM(x_t^{m^*}, t, P_{target}, A_t^*)$
    end if
end for
$I_0{}^{m^*} = D(x_0{}^{m^*})$
Return $I_{m^*}$
```

Self-attention plays a crucially important role in feature preservation of image editing task [7,10]. So we adopt the two correlated pipelines to reconstruct the foreground image, After the early S denoising steps, the layout and structure information is reconstructed. The foreground mask is obtained by the cross attention followed by a mask refinement process. In self-attention layer of source and target pipelines, the query, key and value features (short as Q,K,V) are projected from the spacial feature, the source and target features can be denoted as (Q_s,K_s,V_s) and (Q_t,K_t,V_t) respectively. From the second stage of the target prompt pipeline, we query the projected feature Q_t with the source pipeline feature K_s, V_s) in each corresponding denoising step, and output the features for later denoising steps. This manipulation realizes the texture and appearance information preservation. To accurately maintain the features from the source pipeline, we restrict to query from the corresponding source texture information only on the target object mask regions.

4.3 Image composition

We design a training free image composition method based on diffusion model in the sampling process. Before the noises injection, positioning,resizing and padding zeros on the reference image according to user defined mask is performed. As shown in Fig. 2, the foreground mask has been obtained by the mask extraction module in T-M-N-1 time step. The original latent map is overrode by the foreground latent map considering the foreground mask. The obtained foreground mask can be represented as M^{seg} in the main latent map. In incorporating the noises, we segment the whole latent noises into two parts: the foreground in the filling object and the background regions. The latent map of main image(background image) and the reference image can be denoted as H^m and H^r individually. The composition noise can be calculated as follows:

$$H_t^* = H_t^m \odot (1 - M^{seg}) + H_t^r \odot M^{seg} \tag{5}$$

The whole non-rigid image composition process can be depicted in algorithm 1.

5 Experiments

In this section, we firstly introduce the implementation details of the experiments and organize the benchmark for the experiments. Then we compare our proposed method with the baseline methods. Finally, we finish the ablation study to verify the effectiveness of our proposed method.

5.1 Implementation details and benchmark

Our experiment is based on the pretrained model Stable Diffusion v1.4. The background image is inverted into latent noises with exceptional inversion in[19], while the foreground image is inverted with DDIM null text inversion. In the desired target prompt pipeline of foreground image, we set the starting latent noise the same as the source pipeline. The sampling steps include 50 denoising steps. After M steps (M=5), the self-attention layer's feature the source pipeline will be injected to the target corresponding layers. After N steps (N=25), the mask extraction module will extract the foreground mask based on the target prompt, then the latent noise in this step will be blended with the background latent noise considering the desired bounding box and the obtained foreground mask. To evaluate the effectiveness of our proposed method, we collect samples from the COCO, ImageNet datasets and benchmark in[19] and develop 127 samples benchmark, each sample includes a background image, foreground image, a source prompt, a target prompt and the bounding box mask in main image. All the images are from photorealism domain.

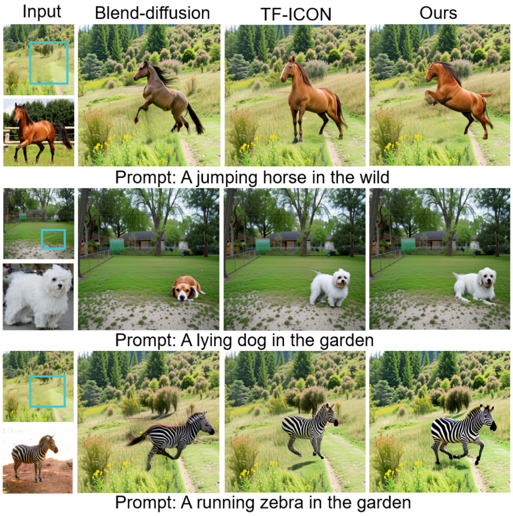

Input
Blend-diffusion
TF-ICON
Ours
Prompt: A jumping horse in the wild
Prompt: A lying dog in the garden
Prompt: A running zebra in the garden

Fig. 4. Qualitative comparison with previous SOTA method in text guided diffusion based image composition

5.2 Compared with previous methods

To evaluate the effectiveness of our proposed methods, we use four evaluation metrics in two aspects: background preservation(PSNR, LPIPS[32]) outside of the foreground mask, target prompt and image consistency(CLIP Similarity[24]) in both whole image and edited regions. The evaluation results are depicted in Table 1.We also establish some qualitative evaluations. We compare our proposed method with the previous diffusion based state-of-the-art methods including blended diffusion[3], TF-ICON[19] based on the target prompt. The synthesis results are presented in Fig. 4. Our proposed method can perform non-rigid

Table 1. Model based quantitative evaluation results of image composition

Method	PSNR(bg)↑	LPIPS(bg)↓	CLIP(whole)↑	CLIP(fg)↑
Blended[3]	23.9	0.064	22.7	22.3
TF-ICON[19]	20.5	0.072	23.2	22.5
Ours	**24.3**	**0.061**	**25.5**	**24.2**

image composition conforming with the target prompt, while previous work can only generate static object composed image.

5.3 Ablation study

The ablation study is performed in the following components: self-attention injection to the whole images, mask based self-attention by cross attention, mask refinement module, background preservation. We conduct the experiments to verify the effectiveness of each components. In the baseline method, the image composition is performed by sampling the reference latent noises from T to 0 using DDIM target prompt without any injection. We add each component Sequentially and the experiment results is presented in Table 2. The results of the experiments show that our proposed method outperform the other combination of components.

Table 2. Ablation study: quantitative comparison of each components

Method	PSNR(bg)↑	LPIPS(bg)↓	CLIP(whole)↑	CLIP(fg)↑
Baseline	19.7	0.069	21.5	21.3
+SA injection	20.1	0.068	22.7	22.5
+CA SA injection	20.5	0.066	23.1	23.3
+Mask refine	21.1	0.065	25.3	23.9
+Background	**24.3**	**0.061**	**25.5**	**24.2**

6 Limitation and future work

There are some limitations of our work. The consistent image generation relies on the self-attention injection from the reference latent features. Due to limited size of feature map and limited pre-trained generation model capability, the model can not learn the texture and layout information effectively from only one reference image, especially for non-rigid image generation. Our next step work focuses on employing the external knowledge to refine the texture information and improve consistent image composition from different views.

7 Summary

We introduce NR-CION, a training free non-rigid consistent image composition framework that leverages inversion and diffusion methods guided by target prompt to achieve non-rigid object image composition. We iteratively optimize the object mask accurately with cross attention and a state-of-the-art segmenter which can guarantee the consistent image editing and seamlessly image composition. Our experiments demonstrate the effectiveness of our proposed NR-CION in non-rigid consistent image editing and seamless image composition. We hope our work can contribute to the community in the image composition task.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4432–4441 (2019)
2. Avrahami, O., Fried, O., Lischinski, D.: Blended latent diffusion. ACM Transactions on Graphics (TOG) **42**(4), 1–11 (2023)
3. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18208–18218 (2022)
4. Azadi, S., Pathak, D., Ebrahimi, S., Darrell, T.: Compositional gan: Learning image-conditional binary composition. Int. J. Comput. Vision **128**, 2570–2585 (2020)
5. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023)
6. Burt, P.J., Adelson, E.H.: A multiresolution spline with application to image mosaics. ACM Transactions on Graphics (TOG) **2**(4), 217–236 (1983)
7. Cao, M., Wang, X., Qi, Z., Shan, Y., Qie, X., Zheng, Y.: Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22560–22570 (October 2023)
8. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. arXiv preprint arXiv:2210.11427 (2022)
9. Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., Raff, E.: Vqgan-clip: Open domain image generation and editing with natural language guidance. In: European Conference on Computer Vision. pp. 88–105. Springer (2022)
10. Dong, W., Xue, S., Duan, X., Han, S.: Prompt tuning inversion for text-driven image editing using diffusion models. arXiv preprint arXiv:2305.04441 (2023)
11. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
12. Ju, X., Zeng, A., Bian, Y., Liu, S., Xu, Q.: Direct inversion: Boosting diffusion-based editing with 3 lines of code. International Conference on Learning Representations (2023)

13. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6007–6017 (2023)
14. Ke, Z., Sun, C., Zhu, L., Xu, K., Lau, R.W.: Harmonizer: Learning to perform white-box image and video harmonization. In: European Conference on Computer Vision. pp. 690–706. Springer (2022)
15. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
16. Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., Gao, J.: Object-driven text-to-image synthesis via adversarial training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12174–12182 (2019)
17. Liu, N., Li, S., Du, Y., Torralba, A., Tenenbaum, J.B.: Compositional visual generation with composable diffusion models. In: European Conference on Computer Vision. pp. 423–439. Springer (2022)
18. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
19. Lu, S., Liu, Y., Kong, A.W.K.: Tf-icon: Diffusion-based training-free cross-domain image composition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2294–2305 (2023)
20. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2022)
21. Miyake, D., Iohara, A., Saito, Y., Tanaka, T.: Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. arXiv preprint arXiv:2305.16807 (2023)
22. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6038–6047 (2023)
23. Porter, T., Duff, T.: Compositing digital images. In: Proceedings of the 11th annual conference on Computer graphics and interactive techniques. pp. 253–259 (1984)
24. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
25. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
26. 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)
27. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Adv. Neural. Inf. Process. Syst. **35**, 36479–36494 (2022)
28. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. International Conference on Learning Representations (2021)

29. Tao, M.W., Johnson, M.K., Paris, S.: Error-tolerant image compositing. Int. J. Comput. Vision **103**, 178–189 (2013)
30. Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., Shum, H.Y.: Mesh editing with poisson-based gradient field manipulation. In: ACM SIGGRAPH 2004 Papers, pp. 644–651 (2004)
31. Zhang, H., Zhang, J., Perazzi, F., Lin, Z., Patel, V.M.: Deep image compositing. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 365–374 (2021)
32. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)

Neighborhood Feature Enhancement Flow Diffusion Model for Point Cloud Generation

Hongcheng Wang, Dongdong Zhang(✉), Taotao Liu, and Xumai Qi

Department of Computer Science and Technology, Tongji University, Shanghai, China
{wanghc,ddzhang,ttliu,2111137}@tongji.edu.cn

Abstract. Collecting 3D point cloud data is cumbersome, so generating high-quality point clouds from existing data can save time and resources while providing more data to support tasks in various fields. In this paper, we propose a neighborhood feature enhancement flow diffusion model for point cloud generation. First, we constructed a multi-scale neighborhood feature aggregation module, which utilizes k-nearest neighbors sampling at different scales to obtain the neighborhood coordinates of each point, thereby aggregating them into coarse global features. Second, we develop a neighborhood attention-based feature enhancement module that uses geometric information in the neighborhood coordinate space to enhance coarse features in the feature space. Then, we used a point-voxel convolutional neural network to reduce redundant features in the enhancement features and output the latent vector of the point cloud. Finally, we transform the latent vectors into data-consistent prior flow features using our designed feature-to-flow data transformation module, seamlessly integrating them into the denoising diffusion model for accurate generation from noisy point clouds. This prior flow approach improves the consistency and coherence of point cloud density distribution. Extensive experiments on the ShapeNet dataset validate the effectiveness of the model in generating 3D point clouds.

Keywords: Diffusion · Attention mechanism · Point cloud generation

1 Introduction

Point cloud data enhances multimedia applications by providing detailed visual and spatial information, improving user immersion and interactivity. Its use in VR, AR, and MR enables the creation of realistic 3D environments [1,2]. However, obtaining point cloud data is time-consuming and sensor-dependent. Thus, it is essential to employ artificial intelligence to advance point cloud generation technologies for delivering high-fidelity data support in the multimedia domain.

In the past few years, deep learning-based methods have become mainstream in 3D point cloud generation. This process typically involves several key steps.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 339–354, 2025.
https://doi.org/10.1007/978-3-031-78389-0_23

The first step is data preprocessing, including denoising, normalization, and data augmentation to ensure consistency and diversity in model inputs. Next is feature extraction, where deep neural networks, such as convolutional neural networks (CNNs), extract high-dimensional features from point clouds, capturing complex relationships and geometric structures between points. The third step involves the design and training of generative models, such as variational autoencoders (VAEs) [3–6], generative adversarial networks (GANs) [7–11], normalizing flow models [12–14], or autoregressive models [15–18], each of which employs different methods to generate new points cloud. These methods typically use hierarchical CNNs or shared MLPs for feature extraction, which may overlook some neighborhood features when aggregating spatial features, potentially leading to a decline in the quality of some generated point clouds.

More recently, denoising diffusion models (DDMs) have achieved remarkable success in image generation has inspired extensive research into point cloud generation algorithms based on DDMs [19,20]. DDMs simulate the transition of data from an ordered state to a noisy state, training neural networks to learn denoising functions that gradually restore noisy data to its original form. However, the step-by-step denoising approach introduces uncertainty and error, particularly with complex geometric structures, which can hinder complete detail recovery and reduce the quality of generated point clouds [21–25]. Additionally, the noise removal process may cause uneven density in the generated point cloud due to fluctuations in the probability density of point cloud data.

To address these issues, this paper proposes a neighborhood feature enhancement point cloud flow diffusion model. Firstly, we introduce a stackable multiscale neighborhood feature aggregation module (MNFA), which uses farthest point sampling (FPS) to select centroids and circular k-nearest neighbors (k-NN) to find adjacent points, obtaining the spatial coordinates around each sampled point. Then, feature aggregation on each sampled region connects local features to form global coarse features. Second, we propose a neighborhood attention-based feature enhancement module (NAFE) that utilizes an attention mechanism to enhance neighborhood features and improve the quality of feature representation. We use k-NN to find the neighbors for each point and perform feature grouping and relative position encoding based on these neighbors. Then, a multi-layer perceptron (MLP) calculates attention weights, which are applied to the value features through weighted summation to generate new feature representations. The enhanced features are then transformed back to the input dimensions through a linear layer and combined with the original features to produce the final enhanced features. This module effectively captures the relationships between points. The enhanced features are finally processed through a point-voxel convolutional neural network (PVCNN) [26] to reduce redundant features in the point cloud, output the mapping vector of the point cloud, and perform reparameterization. Finally, the feature-to-flow Data transformation module (FFDT) applies affine coupling layers to invert the reparameterized features, constructing a normalized flow that ensures data consistency. The stacking of multiple affine coupling layers allows the overall transformation to capture complex distributions. Based on the principles of diffusion models, we use this as the prior distribution for generating shapes. In this way, we improve the problem of incoherence in some of the generated point clouds caused by the sampling randomness of the DDMs.

Our main contributions can be summarized as follows:

- We propose a neighborhood feature-enhanced flow diffusion model for point cloud generation. Our model designs the multi-scale neighborhood feature aggregation module and the neighborhood attention-based feature enhancement module to extract and enhance point cloud features, providing reliable priors for diffusion.
- We developed a feature-to-flow data transformation module that uses normalized flow mapping to stabilize noise sampling during the denoising process of DDMs, thereby improving the quality of the generated point clouds.

2 Related Work

This section explores deep learning techniques for point cloud generation, categorizing various methods and discussing their benefits and drawbacks.

2.1 GAN and VAE-based methods

GANs and VAEs are classical algorithms for image generation and have also been applied to 3D point cloud generation. Achlioptas et al. [7] introduced the r-GAN approach, a significant advancement in deep learning for point clouds. However, MLP structures struggle with capturing local geometric features, limiting fine-grained structure generation. Shu et al. [9] proposed TreeGAN, using a tree-structured GCN to enhance point cloud quality by preserving parent node information. Despite its effectiveness, GCN is challenged by high computational complexity and long training times. Wen et al. [10] introduced a dual-generator approach using two GAN generators for point cloud generation: one for upsampling and the other for refinement. This model requires intricate training strategies and larger datasets. Kim et al. presented SetVAE [3], which uses probabilistic graph models and variational inference to generate diverse, complex point clouds. Although SetVAE produces realistic results, its stochastic nature can cause instability.

2.2 Flow and Autoregression-based methods

Normalizing flows and autoregressive methods generate new data through reversible mappings and probability density estimation. Yang et al. [14] introduced PointFlow, a 3D point cloud model using continuous normalizing flows and variational inference. PointFlow's advantage is its stability compared to GANs. Kim et al. developed SoftFlow [12], which trains normalizing flows on manifolds by estimating conditional distributions of perturbed input data, avoiding dimension mismatch issues. Sun et al. [18] proposed PointGrow, enhancing point cloud correlation through cycles and self-attention for robust generation. However, autoregressive models, such as RPG [17], face scalability issues due to their iterative nature and may struggle with capturing fine details.

2.3 Diffusion based methods

With the success of diffusion models in image processing, researchers have expanded their use to 3D generation.Luo et al. [22] introduced a point cloud probability generation model that simplifies the training objective from the variational bound of point cloud shape likelihood, accelerating the diffusion model training speed. Zhou et al. [25] proposed PVD, which combines denoising diffusion models with a hybrid point-voxel representation for point cloud generation. This method involves a sequence of denoising operations to recover point cloud data from Gaussian noise. It efficiently manages large-scale point cloud datasets and generates intricate topological structures. However, the point-voxel decomposition and subsequent merging could add to computational costs. Both transformer-based [27,28] and diffusion-based methods have achieved remarkable success in the field of image processing. However, challenges remain in point cloud generation as these methods can introduce cumulative errors during denoising that reduce the quality of some point clouds.

3 Methods

This section elaborates on the intricacies of our proposed framework with a meticulous explanation, summarized in Fig.1, which shows all the modules of the model and their workflows. The entire model consists of three modules: the multi-scale neighborhood feature aggregation module (MNFA), the neighborhood attention-based feature enhancement module (NAFE), and the feature-to-flow data transformation module (FFDT). The first two modules are mainly used to extract and enhance neighborhood features, while the last module is used to transform latent features into flow data and input them into the diffusion model for point cloud generation. The four blocks at the bottom of Fig.1 represent some detailed components of the pipeline.

3.1 Multi-scale Neighborhood Feature Aggregation Module

Most diffusion-based point cloud generation models use point coordinates as input, extracting features with shared MLPs and representing the whole with a global feature descriptor. This method may overlook some neighborhood information of the point cloud, leading to a decline in the quality of subsequent generation. Therefore, we adopt a multi-scale k-NN approach, performing two layers of sampling and aggregation for each center point extracted using FPS. This method ensures the compactness of the local neighborhood by utilizing the Euclidean distance between centroid points and other points in the point

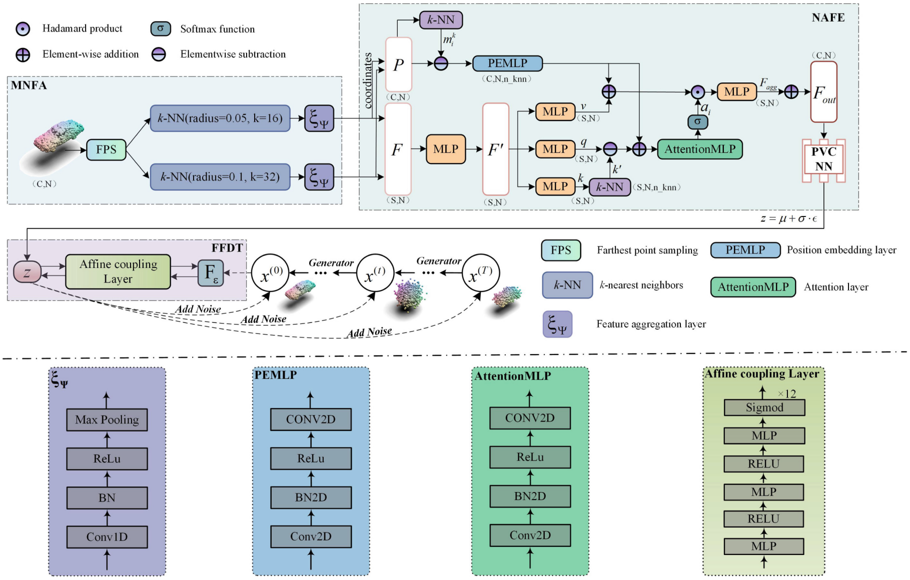


Fig. 1. The pipeline of our generation model. First, we aggregate neighborhood point cloud coordinates and features using the multi-scale neighborhood feature aggregation module (MNFA). Then, we enhance the neighborhood features and obtain the latent vector z using the neighborhood attention-based feature enhancement module (NAFE). Finally, we transform z into flow data using the feature-to-flow data transformation module (FFDT) and input it into the diffusion model for point cloud generation.

cloud. In this way, the centroid points pay more attention to local geometric and topological information.

This module first selects m points from the input point cloud set $X = \{x_1, x_2, x_3, \cdots, x_N\}$ as the initial point set $S = \{x_1, x_2, x_3, \cdots, x_m\}$. For each point x in the set X, it calculates the minimum distance to the points in the set S. Then select the next point x_{m+1} so that it maximizes the minimum distance from the center point of the set S. Repeat this step until the desired number of center points of s is selected to obtain a new set $S = \{x_1, x_2, x_3 \cdots x_s\}$. This process is called FPS can be expressed by the following equation:

$$\begin{aligned} d(x,S) &= \min_{x_j \in S} \|x - x_j\| \\ x_{i+1} &= \arg\max_{x \in X \setminus S} d(x,S) \end{aligned} \tag{1}$$

After obtaining the set of center points S, perform k-NN search to establish the neighborhood relationships for each point. For each point $p_i \in S$, calculate its distance to other points and find the k nearest points and former the set of neighbors $\mathcal{N}_k(p_i) = \{p_{i1}, p_{i2}, \cdots, p_{ik}\}$. The p_{ij} represents the j-th nearest point to p_i. The k-NN process can be expressed as:

$$\{p_i\}_{i=1}^{k} = k\text{-NN}(X, \{S_i\}_{i=1}^{s}) \tag{2}$$

Finally, we aggregate the k-NN of each center point to obtain a new set of point coordinates P. Finally, the point coordinates are aggregated through the module $\xi_\Psi(\cdot)$ with parameter Ψ, which employs 1×1 convolution, batch normalization (BN), ReLU, and max pooling to aggregate the neighborhood set $\{P_i\}_{i=1}^{s}$ into coarse global features:

$$F = \text{Contact}\xi_\Psi\left(\{p_i\}_{i=1}^{k}\right) \tag{3}$$

3.2 Neighborhood Attention-based Feature Enhancement Module

Point cloud data consists of an unordered set of three-dimensional points, lacking explicit topological structure. Therefore, after aggregating the point cloud P, we utilize an attention mechanism to aggregate the features of each point's k nearest neighbors, capturing neighborhood feature correlations from the geometric structure of the point cloud to enhance the discriminability and robustness of the point cloud feature representation.

We first use a linear layer to map F to the high-dimensional initial feature map of the point cloud F'. Simultaneously, for each point in the input P, we use Equation (2) to find the 16 nearest points based on their spatial positions and obtain the indices of these 16 nearest points $N_k(p_i)$:

$$N_k(p_i) = \text{argsort}_{16}\left(\sum_{d=1}^{3}\left(\parallel p_a - p_b \parallel^2\right)\right) \tag{4}$$

where p_a and p_b are any two points in P_i. Next, we use three MLPs on F' to map the attention keys (k), values (v), and queries (q), respectively.

To enhance neighborhood features, we combine the coordinates of each point p_i in P with the coordinates of its k-NN indices $N_k(p_i)$ through a grouping operation, thereby generating a new feature set $\{\mathbf{m}_i^k | l = 1, 2, \ldots, 16\}$. Then, perform elementwise subtraction between $\mathbf{m}_k^{\boldsymbol{l}}$ and the coordinates of P. Subsequently, apply positional encoding through the PEMLP layer. This ensures that the features of each point not only include its own information but also reflect its relative position within the entire point cloud and its local geometric relationships.

$$PE = \text{PEMLP}(P \ominus \mathrm{m}_k^1) \tag{5}$$

In order to capture the differences and similarities between the features of a point and its neighboring points, we reshape the query features k' by applying the same grouping operation used in the aforementioned positional encoding to the key (k) and indices $N_k(p_i)$. Next, we add positional encoding to q, k', and v, and compute the attention weights using the query q and the key k'. These weights are then used to perform a weighted sum of the value features.

$$a_i = \text{Softmax}(\text{AttnMLP}(q \oplus k' + PE))\ ,\ F_{agg} = \sum a_i \cdot (\nu + PE) \tag{6}$$

Then, the aggregated features F_{agg} are mapped back to the original feature space using an MLP and added to the input features through residual concatenation, resulting in an enhanced feature representation F_{out}.

$$F_{\text{out}} = \text{MLP}(F_{\text{ agg}}) + F' \tag{7}$$

Finally, the PVCNN is used to reduce redundant features in the point cloud, outputting the latent feature vectors of the point cloud.

3.3 Feature-to-Flow Data Transformation Module

In point cloud diffusion models, generating samples by denoising randomly sampled noise from a Gaussian distribution can lead to accumulated noise errors, resulting in discontinuous point clouds. To address this issue, we combine flow models to map latent vectors from the Gaussian distribution to the data distribution, achieving a reversible and continuous transformation. As shown in Fig. 2, the probability density can be precisely controlled when sampling noise, thus generating more accurate samples.

Inspired by PointFlow [14], we transform the discrete flow into a continuous flow model, thus conforming to the denoising process, and transform it from the form of the data x to the form of the latent vector z:

$$\begin{aligned} z &= F_{\varepsilon}(w) = w + \int_{t_0}^{t_1} f_{\varepsilon}(w(t), t)dt \\ \log P_{\varepsilon}(z) &= \log P\Big(F_{\varepsilon}^{-1}(z)\Big) - \int_{t_0}^{t_1} Tr\left(\frac{\partial f_{\varepsilon}}{\partial w(t)}\right)dt \end{aligned} \tag{8}$$

where f is a neural network, w represents the prior distribution parameters and $w(t_1) = z$. To find the value corresponding to the prior distribution at time t_0, using the inverse operation of the flow: $w(t_0) = z + \int_{t_1}^{t_0} f(w(t), t)\text{d}t$.

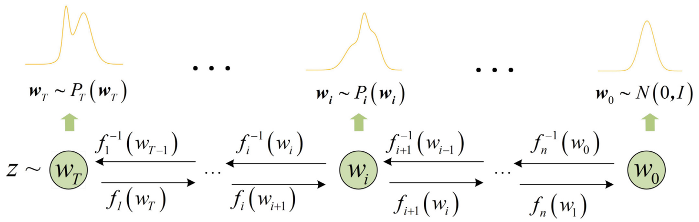


Fig. 2. The normalization flow generates samples consistent with the latent vector distribution from a continuous, easy-to-sample distribution.

We use affine coupling layers to perform the inverse transformation on the input. Each layer transforms a part of the current input while keeping the other part unchanged, ensuring a one-to-one correspondence between z and w. This method allows the precise calculation of the probability for the target distribution through the application of the change of variables formula:

$$p(z) = p_w(w)\left| \det \frac{\partial F_\varepsilon}{\partial w} \right|^{-1},\ w = F_\varepsilon^{-1}(z) \tag{9}$$

3.4 Generation process

Our research utilizes a DDM to generate point cloud data. The N point clouds $X^{(0)} = \{x_i^{(0)}\}_{i=1}^N$ are treated as diffusing particles, with each point sampled from the distribution $q(x_i^{(0)}|z)$ of the latent shape variable z. The process involves two stages: forward noise addition and reverse denoising.

Traditionally, the noise addition process in DDMs does not involve a learning process. Given a point cloud distribution $x^{(0)} \sim q(x^{(0)})$, a fixed single-step and multi-step forward diffusion process is conducted, defined as follows:

$$q(x_i^{(1:T)}|x_i^{(0)}) = \prod_{t=1}^{T} q(x_i^{(t)}|x_i^{(t-1)}), q(x^{(t)}|x^{(t-1)}) = \mathcal{N}(x^{(t)}; \sqrt{1-\beta_t} x^{(t-1)}, \beta_t I) \tag{10}$$

Eventually, the point cloud distribution approaches a standard normal distribution $\mathcal{N}(0, I)$. The parameterized neural network and a latent variable are then used to gradually recover a 3D point cloud with a specific shape from the noise $p(x_i^{(T)})$ through a reverse diffusion process called $p_\theta(x^{0:T}|z)$:

$$q(X^{(1:T)}|X^0) = \prod_{i=1}^{N} q(x_i^{(1:T)}|x_i^{(0)}) \quad , \quad p_{\boldsymbol{\theta}}(X^{(0:T)}|z) = \prod_{i=1}^{N} p_{\boldsymbol{\theta}}(x_i^{(0:T)}|z) \tag{11}$$

The objective of the denoising diffusion model is to maximize the log-likelihood of the target point cloud $X^{(0)}$, denoted as $\mathbb{E}[\log p_\theta(X^{(0)})]$. Due to the difficulty in directly optimizing this objective, we introduce the evidence lower bound (ELBO):

$$\mathbb{E}[\log p_\theta(X^{(0)})] \geq \mathbb{E}_q\Big[\log \frac{p_\theta(X^{(0:T)}, z)}{q(X^{(1:T)}, z|X^{(0)})}\Big] \tag{12}$$

Unlike the method mentioned above that samples random noise from a Gaussian distribution for denoising, we introduce a continuous data stream in Section 3.3. Therefore, we can replace z with w. Expand equation (12) and substitute equation (9) into it, we obtain the final objective function to be optimized:

$$\begin{aligned} L_G(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varepsilon}) =& \mathbb{E}_q \Bigg[\sum_{t=2}^{T} \sum_{i=1}^{N} D_{\mathrm{KL}} \left(q \left(\boldsymbol{x}_i^{(t-1)} \mid \boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)} \right) \| \right. \\ & \left. p_{\boldsymbol{\theta}} \left(\boldsymbol{x}_i^{(t-1)} \mid \boldsymbol{x}_i^{(t)}, \boldsymbol{z} \right) \right) - \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}} \left(\boldsymbol{x}_i^{(0)} \mid \boldsymbol{x}_i^{(1)}, \boldsymbol{z} \right) \qquad (13) \\ & + D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}} \left(\boldsymbol{z} \mid \boldsymbol{X}^{(0)} \right) \| p_{\boldsymbol{w}} (\boldsymbol{w}) \cdot \left| \det \frac{\partial F_{\varepsilon}}{\partial \boldsymbol{w}} \right|^{-1} \right) \Bigg] \end{aligned}$$

4 Experiments

4.1 Experiment Settings

Datasets. For the point cloud generation task, we conducted experiments on the ShapeNet dataset, which contains 51,127 shapes across 55 categories. The dataset is split into training, testing, and validation sets with proportions of 80%, 15%, and 5%, respectively. We quantitatively compared our method with several state-of-the-art generative models in three categories of ShapeNet: airplanes, chairs, and cars.

Table 1. Comparison results (%) on shape metrics of our model and baseline models

Method	Chair				Airplane				Car			
	1-NNA (↓)		COV (↑)		1-NNA (↓)		COV (↑)		1-NNA (↓)		COV (↑)	
	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD
r-GAN	83.69	99.7	24.27	15.13	98.40	96.79	30.12	14.32	94.46	99.01	19.03	6.539
l-GAN (CD)	68.58	83.84	41.99	29.31	87.30	93.95	38.52	21.23	66.49	88.78	38.92	23.58
l-GAN (EMD)	71.90	64.65	38.07	44.86	89.49	76.91	38.27	38.52	71.16	66.19	37.78	45.17
PointFlow	62.84	60.57	42.90	**50.00**	75.68	70.74	47.90	46.41	58.1	56.25	46.88	50
SoftFlow	59.21	60.05	41.39	47.43	76.05	65.80	46.91	47.90	64.77	60.09	42.9	44.6
SetVAE	58.84	60.57	46.83	44.26	76.54	67.65	43.70	48.40	59.94	59.94	**49.15**	46.59
DPF-Net	62.00	58.53	44.71	48.79	75.18	65.55	46.17	48.89	62.35	54.48	45.74	49.43
DPM	60.05	74.77	44.86	35.50	76.42	86.91	48.64	33.83	68.89	79.97	44.03	34.94
MeshDiffusion	53.69	**57.63**	46.00	46.71	76.44	76.26	47.34	42.15	81.43	87.84	34.07	25.85
PVD	57.09	60.87	36.68	49.24	73.82	64.81	**48.88**	**52.09**	54.55	53.83	41.19	50.56
Ours	**56.92**	61.47	**47.82**	44.47	**72.73**	**61.51**	47.04	51.81	**51.28**	**50.34**	48.56	**51.44**

Implementation details and Evaluation metric. In our experiment, the voxel size is fixed at 16, and the k values for the two layers are [16, 32] and [64, 128]. To evaluate the quality of point clouds, we use a series of evaluation algorithms: coverage score (COV), minimum matching distance (MMD), 1-nearest neighbor classifier accuracy (1-NNA), and Jensen-Shannon divergence (JSD).

4.2 Comparison to State-of-the-art Works

Quantitative evaluation. In this section, we conduct two sets of experiments for comparison based on the ShapeNet dataset. The first set of experiments compares the 1-NNA and COV metrics on the chair and airplane subsets of ShapeNet. We selected a range of state-of-the-art algorithms as comparison benchmarks. These algorithms represent the diversity and latest advancements in the field of 3D shape generation, including non-diffusion algorithms such as r-GAN [7], 1-GAN [7], PointFlow [14], SoftFlow [12], SetVAE [3], and DPF-Net [13], as well as diffusion-based algorithms such as DPM [22], MeshDiffusion[29], PVD [25]. The specific data is presented in Table 1.

The second set of experiments compares the MMD and JSD metrics on the airplane and chair datasets. We compared our method with PC-GAN [7], GCN-GAN [30], TreeGAN [9], PointFlow [14], ShapeGF [31], PDGN [32] and DPM[22]. The detailed results are shown in Table 2.

Table 2. Comparison of point cloud generation performance. CD is multiplied by 10^3. EMD is multiplied by 10^1, and JSD is multiplied by 10^3.

Method	Airplane			Chair		
	MMD (↓)		JSD (↓)	MMD (↓)		JSD (↓)
	CD	EMD	-	CD	EMD	-
PC-GAN	3.819	1.810	6.188	13.436	3.104	6.649
GCN-GAN	4.713	1.650	6.669	15.354	2.213	21.708
TreeGAN	4.323	1.953	15.646	14.936	3.613	13.282
DualGAN	3.321	1.082	1.304	12.687	1.879	7.154
PointFlow	3.688	1.090	1.536	13.631	1.856	12.474
ShapeGF	3.306	1.027	**1.059**	13.175	1.785	**5.996**
PDGN	3.287	1.121	1.891	12.852	2.082	6.764
DPM	3.276	1.061	1.067	12.276	**1.784**	7.797
Ours	**3.254**	**0.418**	1.396	**12.223**	1.857	8.808

Analyzing the data in Tables 1 and 2 shows that our model performs comparably to state-of-the-art methods across various metrics. This success is due to our model's comprehensive consideration of both local and global information and its effective integration of spatial coordinates. Additionally, the sequential flow model stabilizes the diffusion generation process, significantly enhancing the quality and high-fidelity generation of 3D shapes in the generated point cloud data.

Visual results. We compared our results with the diffusion-based PVD and flow-based PointFlow point cloud generation algorithms. Fig. 3 shows the visualization results of the generated point clouds on the ShapeNet dataset, displaying

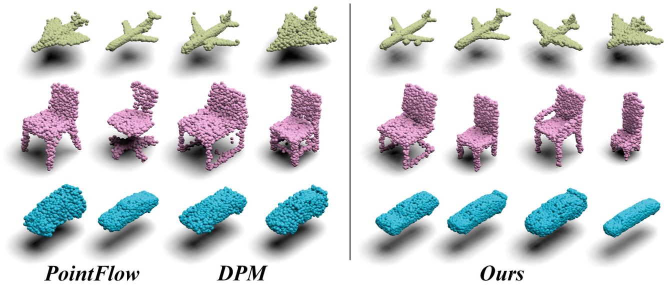


Fig. 3. Visualization of the generation quailty. From top to bottom: “Airplane”, “Chair” and “Car”.

airplanes, chairs, and cars from top to bottom. As shown in the figure, the generated point clouds exhibit clear structural features, strongly confirming the effectiveness of our model in capturing geometric information and significantly improving the issues of holes and scattered points in the generated outputs.

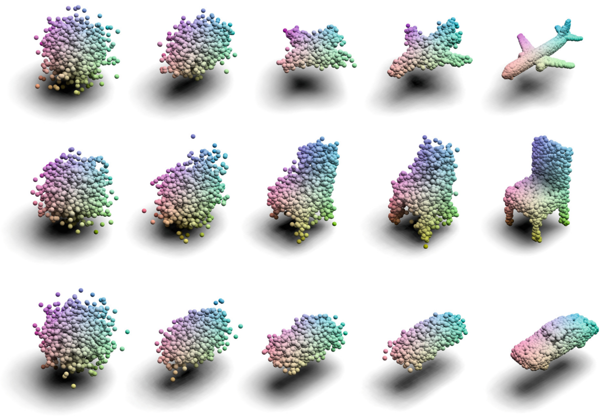


Fig. 4. Visualization of the denoising process. From top to bottom: “Airplane”, “Chair” and “Car”.

To further demonstrate the generative capability of our model, we also provide the generation process from noisy to denoised for the above three categories. As shown in Fig. 4, the progression unfolds from random noise to the final 3D shapes in left-to-right order.

The visualization of features in joint training. We train a model jointly without any class conditioning on 6 different categories (airplane, chair, car, tower, bag, and basket) from ShapeNet. We employ t-SNE to project the latent vectors generated by the NAFE into a 2D plane and present it in Fig. 5. The figure clearly delineates distinct separations among the majority of the categories, signifying the proficiency of our model in capturing meaningful distinctions within the data.

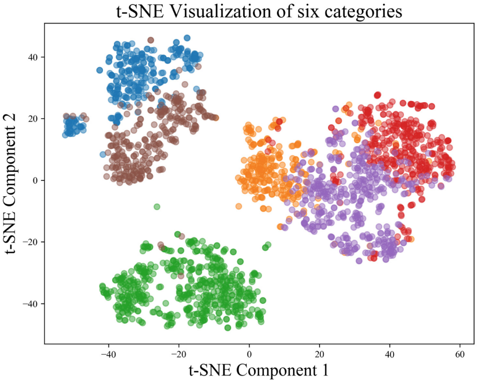


Fig. 5. The t-SNE visualization of the joint distribution generated by unconditional joint training of 6 categories.

We randomly selected some small sample data from the generated point clouds for demonstration (tower, bag, and basket). Each type of point cloud contains between 100 and 200 points, with test samples numbering approximately 11 to 24. The visualization results are shown in Fig. 6. Our network is capable of forming novel high-fidelity point clouds, ensuring surface coverage without significant gaps. The experimental results validate the feasibility of multi-object training, and our model demonstrates strong learning ability for few-shot point cloud data. This allows us to create a single model that can handle different training categories without the need to train a unique model for each category.

Point Cloud Completion. In addition to the aforementioned research analysis, we are considering whether our model can perform upsampling with fewer input point clouds. During inference, the proposed model can generate a series of point-to-point distance samples for each point cloud, thereby achieving point cloud completion. Specifically, we use 150 and 200 points as input for each point set of airplanes and cars, respectively. By utilizing the pre-trained model, we combine global shape variables and partial point clouds to synthesize a complete point cloud. The qualitative results of the output point cloud visualization are shown in Fig. 7. As the output point cloud shows, when our input points are relatively sparse, the trained model can accurately perform point cloud upsampling.

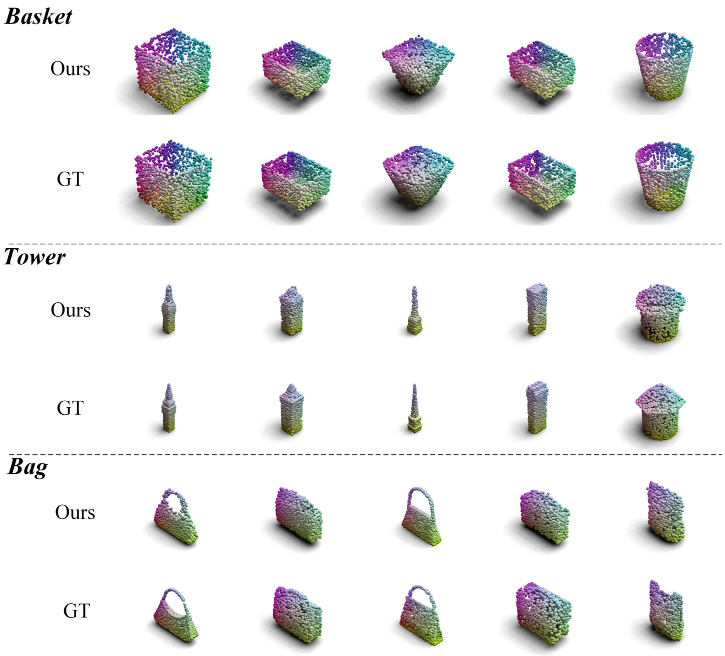


Fig. 6. Examples of few-shot point cloud generation from joint training. From top to bottom: “Basket,” “wer,”nd “g”.

4.3 Ablation Study

In this section, we conduct an ablation study comparing MNFA, NAFE, and PVCNN within NAFE. To show the effectiveness of our model on multi-class training, we trained separately on three subsets (chairs, cars, and airplanes) and tested on the airplane class. Each set of experiments was iterated 200,000 steps. Table 3 shows the necessity of each of the three modules by evaluating their removal individually and summarizing the quantitative results. Optimal performance is achieved with all modules integrated. The removal and reintroduction of the PVCNN revealed similar numerical results, indicating that our proposed two modules collectively impact the point cloud data and focus on geometric complexity.

5 Conclusion

In this paper, we propose the neighborhood feature enhancement flow diffusion model for point cloud generation. The constructed point cloud feature extrac-

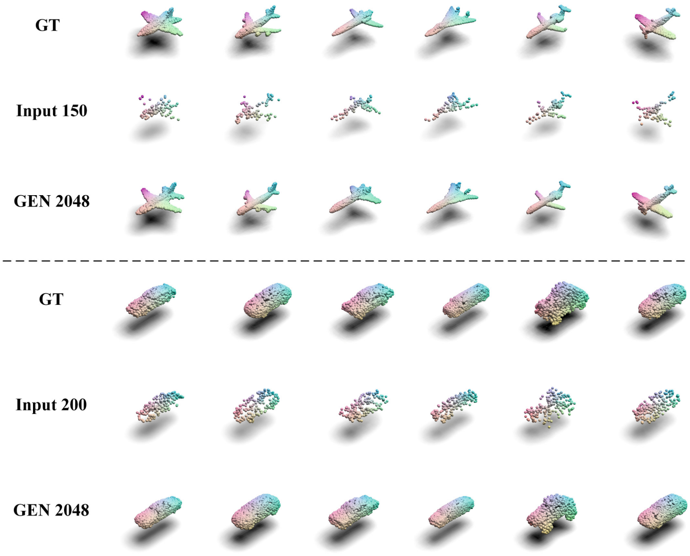


Fig. 7. Visualised experimental results of point cloud completion.

Table 3. Our model component ablation study results (%)

MNFA	NAFE	PVCNN	1-NNA(↓)		COV(↑)		JSD(↓)
			CD	EMD	CD	EMD	-
×	×	×	78.66	79.83	23.72	24.02	14.32
✓	×	×	75.17	79.13	23.86	24.36	12.50
✓	✓	×	73.42	84.58	24.67	26.79	11.58
✓	✓	✓	**70.28**	**78.32**	**26.36**	**27.27**	**11.24**

tion module utilizes neighborhood features of the point cloud and enhances both global and local point cloud features, providing robust prior knowledge for subsequent diffusion. By integrating these enhanced features into normalized streaming data and denoising diffusion models, we generate high-quality point clouds. Experimental results demonstrate the effectiveness of this method in both single-class and multi-class generation tasks on general point cloud datasets.

Acknowledgements. This research is funded by National Key R&D Program of China (No. 2021YFB3301900).

References

1. Liu, J., Wu, Y., Gong, M., Miao, Q., Ma, W., Xu, C.: Exploring Dual Representations in Large-Scale Point Clouds: A Simple Weakly Supervised Semantic Segmentation Framework. In: ACM MM, pp. 2371-2380 (2023)
2. Yuan, Y., Kong, R., Xie, S., Li, Y., Liu, Y.: Patchbackdoor: Backdoor attack against deep neural networks without model modification. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 9134-9142 (2023)
3. Kim, J., Yoo, J., Lee, J., Hong, S.: Setvae: Learning hierarchical composition for generative modeling of set-structured data. In:CVPR, pp. 15059-15068 (2021)
4. Litany, O., Bronstein, A., Bronstein, M., Makadia, A.: Deformable shape completion with graph convolutional autoencoders. In:CVPR, pp. 1886-1895 (2018)
5. Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L. J.: Structurenet: Hierarchical graph networks for 3d shape generation. arXiv preprint arXiv:1908.00575 (2019)
6. Pang, Y., Wang, W., Tay, F. E., Liu, W., Tian, Y., Yuan, L.: Masked autoencoders for point cloud self-supervised learning. In: ECCV, pp. 604-621 (2022)
7. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: PMLR, pp. 40-49 (2018)
8. Ibing, M., Lim, I.,Kobbelt, L.: 3d shape generation with grid-based implicit functions. In: CVPR, pp. 13559-13568 (2021)
9. Shu, D. W., Park, S. W., Kwon, J.: 3d point cloud generative adversarial network based on tree structured graph convolutions. In: ICCV, pp. 3859-3868 (2019)
10. Wen, C., Yu, B., Tao, D.: Learning progressive point embeddings for 3d point cloud generation. In: CVPR, pp. 10266-10275 (2021)
11. Yang, Z., Chen, Y., Zheng, X., Chang, Y., Li, X.: Conditional GAN for Point Cloud Generation. In:ACCV, Computer Vision, pp. 3189-3205 (2022)
12. Kim, H., Lee, H., Kang, W. H., Lee, J. Y., Kim, N. S.: Softflow: Probabilistic framework for normalizing flow on manifolds. In: NeurIPS, vol. 33, pp. 16388-16397 (2020)
13. Klokov, R., Boyer, E., Verbeek, J.: Discrete point flow networks for efficient point cloud generation. In: ECCV, pp. 694-710 (2020)
14. Yang, G., Huang, X., Hao, Z., Liu, M. Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: ICCV, pp. 4541-4550 (2019)
15. Chen, C., Han, Z., Liu, Y. S., Zwicker, M.: Unsupervised learning of fine structure generation for 3D point clouds by 2D projections matching. In: ICCV, pp. 12466-12477 (2021)
16. Ibing, M., Kobsik, G., Kobbelt, L.: Octree transformer: Autoregressive 3d shape generation on hierarchically structured sequences. In: CVPR, pp. 2697-2706 (2023)
17. Kol, W. J., Chiu, C. Y., Kuo, Y. L., Chiu, W. C.: Rpg: Learning recursive point cloud generation. In: IROS, pp. 544-551 (2022)
18. Sun, Y., Wang, Y., Liu, Z., Siegel, J., Sarma, S.: Pointgrow: Autoregressively learned point cloud generation with self-attention. In: WACV, pp. 61-70 (2020)
19. Nichol, A. Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: PMLR, pp. 8162-8171 (2021)
20. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR, pp. 10684-10695 (2022)
21. Li, Y., Dou, Y., Chen, X., Ni, B., Sun, Y., Liu, Y., Wang, F.: Generalized deep 3d shape prior via part-discretized diffusion process. In: CVPR, pp. 16784-16794 (2023)

22. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: CVPR, pp. 2837-2845 (2021)
23. Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K.: Lion: Latent point diffusion models for 3d shape generation. In: NeurIPS, vol. 35, pp. 10021-10039 (2022)
24. Wu, L., Wang, D., Gong, C., Liu, X., Xiong, Y., Ranjan, R., Liu, Q.: Fast point cloud generation with straight flows. In: CVPR, pp. 9445-9454 (2023)
25. Zhou, L., Du, Y., Wu, J.: 3d shape generation and completion through point-voxel diffusion. In: ICCV, pp. 5826-5835 (2021)
26. Liu, Z., Tang, H., Lin, Y., Han, S.: Point-voxel cnn for efficient 3d deep learning. In: NeurIPS, vol. 32 (2019)
27. Guo, M. H., Cai, J. X., Liu, Z. N., Mu, T. J., Martin, R. R., Hu, S. M.: Pct: Point cloud transformer. In: Computational Visual Media, vol. 7, pp. 187-199 (2021)
28. Xu, R., Hui, L., Han, Y., Qian, J., Xie, J.: Transformer-based Point Cloud Generation Network. In: ACM MM, pp. 4169-4177 (2023)
29. Liu, Z., Feng, Y., Black, M. J., Nowrouzezahrai, D., Paull, L., Liu, W.: Meshdiffusion: Score-based generative 3d mesh modeling.In: ICLR, (2023)
30. Valsesia, D., Fracastoro, G., Magli, E.: Learning localized representations of point clouds with graph-convolutional generative adversarial networks. IEEE Trans. Multimedia **23**, 402–414 (2020)
31. Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., Hariharan, B.: Learning gradient fields for shape generation. In: ECCV, vol. 16, pp. 364-381 (2020)
32. Hui, L., Xu, R., Xie, J., Qian, J., Yang, J.: Progressive point cloud deconvolution generation network. In: ECCV, vol. 16, pp. 397-413 (2020)

Beta-Sigma VAE: Separating Beta and Decoder Variance in Gaussian Variational Autoencoder

Seunghwan Kim and Seungkyu Lee(✉)

Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea
{overnap,seungkyu}@khu.ac.kr

Abstract. Variational autoencoder (VAE) is an established generative model but is notorious for its blurriness. In this work, we investigate the blurry output problem of VAE and resolve it, exploiting the variance of Gaussian decoder and β of beta-VAE [14]. Specifically, we reveal that the indistinguishability of decoder variance and β hinders appropriate analysis of the model by random likelihood value, and limits performance improvement by omitting the gain from β. To address the problem, we propose Beta-Sigma VAE (BS-VAE) that explicitly separates β and decoder variance σ_x^2 in the model. Our method demonstrates not only superior performance in natural image synthesis but also controllable parameters and predictable analysis compared to conventional VAE. In our experimental evaluation, we employ the analysis of rate-distortion curve and proxy metrics on computer vision datasets. The code is available on https://github.com/overnap/BS-VAE.

Keywords: variational autoencoder · generative modeling · image synthesis · representation learning · rate-distortion theory

1 Introduction

Generative modeling has been a headliner of deep learning research over the last decade. It approximates the distribution of observed samples such as natural images or natural language sentences. Variational autoencoder (VAE) [17,30], one of the most popular generative deep neural networks with well-developed mathematical background, has demonstrated competitive performance in realistic sample synthesis [3,29], image segmentation [19], data augmentation [27], image compression [10], and reinforcement learning [26,28].

However, VAE has a notorious blurry output problem that hinders achieving cutting-edge generation quality. As a consequence, VAE has been adopted in various downstream tasks, but left off in major generative network applications. The technical source of the blurry output problem is difficult to pinpoint. Prior methods have been proposed to improve either the reconstruction quality or generation quality of VAEs with the variance of decoder distribution [34] and β

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 355–369, 2025.
https://doi.org/10.1007/978-3-031-78389-0_24

of beta-VAE [14]. The lower the variance of decoder is, the sharper the output images are, since the variance represents the noise of decoder distribution. In return, the risk of bad local minimizers increases, as the loss smoothing effect of high variance is reduced [8]. On the other hand, β extends VAE outside the likelihood, which allows beta-VAE to obtain useful properties such as latent disentanglement [5,6,11,14] and rate-distortion tradeoff [1,2,4]. One can achieve sharp output by carefully tuning β.

These two parameters appear to have similar effects. Moreover, in special cases, e.g. Gaussian VAE with constant decoder variance, they are mathematically equivalent. Nevertheless, as they have separate design motivations, it is clear that their purposes and impacts are different. Confusion with the two parameters in prior approaches hinder performance improvement and model analysis of VAEs. For example, a method considering the two parameters are the same and optimizing a single integrated parameter cannot achieve the optimality of two parameters properly. The integrated parameter also leads to an indeterminate variance, so the likelihood value becomes arbitrary. In this case, likelihood values can vary for the same model and weights making the comparison virtually meaningless, which is very damaging to the research of VAEs.

In this work, we analyze the confusion about the influence of decoder variance and β, and propose a simple solution that derives optimal performance of VAEs.

Our contributions are as follows:

- **Investigation of blurry output problem in VAEs.** The blurry output is a complex problem that is difficult to explain with any single factor. We classify it into poor reconstruction and poor generation followed by respective problem definitions and analysis.
- **Identification of the problems occurring in Gaussian VAE in which the variance of decoder σ_x^2 and β of beta-VAE [14] are considered as a single integrated parameter.** Both parameters show similar effects and have been used to address the blurry output problem of VAEs. On the other hand, based on their different design motivations, σ_x^2 and β affect the quality of reconstruction and generation respectively, which introduces non-optimality in the performance of VAEs.
- **Proposing a simple and explicit method to separate β and σ_x^2.** Our method, Beta-Sigma VAE (BS-VAE), improves the performance of Gaussian VAE, as it takes advantage of both parameters. It also makes VAE more controllable, since it obtains a model of the rate-distortion curve with optimal decoder variance. Furthermore, it ensures that the same model and weights always have the same likelihood value, which enables predictable and meaningful analysis.

Our claims are validated on computer vision datasets. Our method, BS-VAE, is independent of architecture and scale, so it is applicable to most VAE-variants. We hope that our efforts encourage following research on VAEs to extend constructive analysis and accomplish competitive performance in many generative network applications.

2 Background

Variational autoencoder (VAE). VAE [17,30] models a parameterized distribution $p_\theta(x) = \int p_\theta(x|z)p(z)dz$ for the observable variable x and latent variable z. It is fundamentally a maximum likelihood estimation. The log-likelihood $\log p_\theta(x)$ is generally intractable. Hence, VAE performs variational inference employing variational distribution $q_\phi(z|x)$. It learns evidence lower bound (ELBO) of the log-likelihood that consists of reconstruction error, Equation (1), and KL divergence, Equation (2). Note that the objectives are about a single sample x_i for convenience.

$$\begin{aligned} & - \log p_\theta(x_i) \leq -\mathrm{ELBO}(\theta, \phi, x_i) \\ = & - E_{z\sim q_\phi(z|x_i)}[\log p_\theta(x_i|z)] \qquad (1) \\ & + D_{KL}(q_\phi(z|x_i)||p(z)) \qquad (2) \end{aligned}$$

Gaussian VAE. The architecture of VAE, the encoder $q_\phi(z|x)$ followed by the decoder $p_\theta(x|z)$, is similar to an autoencoder. Different from autoencoder, VAE establishes probability distributions which are usually set to Gaussian in computer vision applications [9,17,37]. For the observable variable x and latent variable z, Gaussian VAE is the variational autoencoder consisting of the following encoder $q_\phi(z|x)$ and decoder $p_\theta(x|z)$.

$$\begin{aligned} q_\phi(z|x) &\sim \mathcal{N}(\mu_z(x), \Sigma_z(x)) \\ p_\theta(x|z) &\sim \mathcal{N}(\mu_x(z), \Sigma_x(z)) \end{aligned}$$

where Σ_z is the diagonal covariance matrix and Σ_x is the scalar matrix in conventional setting.

$$\begin{aligned} \Sigma_z(x) &= \mathrm{diag}(\sigma_z^2(x)) \\ \Sigma_x(z) &= \sigma_x^2(z)I \end{aligned}$$

Restricting the Σ_z to diagonal matrix induces orthogonality between latent channels [20,24,32], which helps latent disentanglement and constrains the computation to be linear in $\dim z$. However, it is argued that this unduly limits the expressive power of encoder [35,40].

The Σ_x is usually assumed to be scalar and constant. The typical VAE that outputs only the mean μ_x is correspond to the case as it implies $\sigma_x^2 = 1/2$. This makes computation easier and avoids the optimization problem [25,31] that occurs when Σ_x is a trainable parameter. The learnable Σ_x tends to approach 0 as training progresses, causing the objective to diverge to infinite. However, the constant scalar variance does not allow VAE to reach the optimal latent structure, whereas the learnable scalar variance does [8,9]. This theoretical achievement is extended to the empirical nonlinear case [18,25], which reports its superior performance despite being unstable and prone to overfitting. We will adopt scalar $\Sigma_x = \sigma_x^2(z)I$ but discuss constant σ_x^2.

Learnable decoder variance. The learnable variance of decoder σ_x^2 outperforms constant scalar variance [25,34], but introduces a nontrivial optimization problem [25,31]. In many conventional studies and implementations, the variance of decoder is often left constant. This empirically leads to degraded results [8,34], as the trainable variance has been discussed as essential for the optimization of Gaussian VAE [8,9,18]. Although few works successfully employed learnable variance stably [34,38], constant variance has been used in most prior research because learnable variance makes training process unstable and the effect is considered trivial [34].

Beta-VAE. Beta-VAE [14] on which applied works rather focus, demonstrates a simple yet effective enhancement on VAE. It introduces hyperparameter β into the ELBO that balances the reconstruction error and KL divergence as shown in Equation (3). β influences the regularization by the KL divergence and latent disentanglement [5,6,11,14], which results in the efforts of fine-tuning β in practice [19,28]. These effects are attributed by estimating how well the variational distribution $q_\phi(z|x)$ follows the prior $p(z)$ in many cases [15].

$$\begin{aligned} L_\beta(\theta, \phi, x_i) = & - E_{z \sim q_\phi(z|x_i)}[\log p_\theta(x|z)] \\ & + \beta D_{KL}(q_\phi(z|x_i)||p(z)) \end{aligned} \tag{3}$$

Rate-distortion theory on β. The balance of β is explained by rate-distortion theory [16] in which VAE is analogous to lossy compression [1,2,4]. The function of VAE is viewed as compressing a given x into a usually lower-dimensional z and restoring it, resembling a lossy compression system. In this context, reconstruction error corresponds to distortion and KL divergence term corresponds to rate in information theory. Therefore beta-VAEs are depicted by rate-distortion curve where each β value determines a specific point. This indicates that beta-VAE changes the generation performance with β, unlike vanilla VAE, as the location of a point on the curve characterizes the model's performance.

3 Beta-Sigma VAE

3.1 Categorizing Blurriness

VAE is notorious for producing undesirable blurry output, which is a drawback given that its competitors, such as GAN [12] or diffusion model [36], produce very sharp output. Here, blurry means losing fine details that are usually present in high frequencies. This is a complex mix of phenomena, making it difficult to pinpoint a technical source. To ease further analysis, we categorize the blurry output problem into two types: poor reconstruction and poor generation.

Poor reconstruction refers to a model failing to reconstruct the training data regardless of generation. It corresponds to underfitting in general terms, which means that the VAE is not trained well, i.e., its likelihood for training or test data is low. The main cause is inadequate distribution modeling that does not

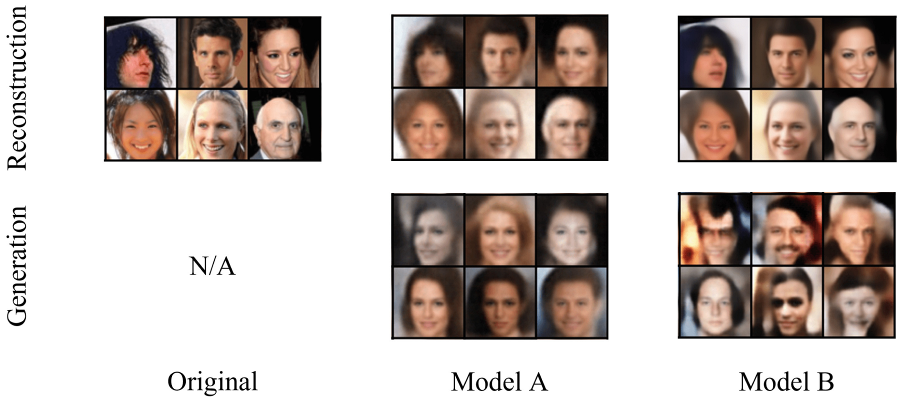


Fig. 1. The toy example of poor reconstruction and poor generation on CelebA dataset [22]. Model A displays a blurry reconstruction, but the quality of reconstruction and generation is consistent. Model B shows a relatively clear reconstruction, but the generation is blurry and unrealistic. Their setup is identical to the one in the experiment, and the samples are selected without any intention, i.e., no cherry picking.

fit the given data. In Gaussian VAE, the value of variance σ_x and whether σ_x is constant or learnable are important for good reconstruction. The impact of variance modeling has been reported extensively [8,9,34]. For example, the low variance provides a high likelihood and thus improves reconstruction practically. The other cause is the limitation of neural network architecture, which is not the focus of this work, so many architectures and techniques have been proposed to address it [7,29,37].

Poor generation refers to a model failing to generate while being good at reconstruction relatively. In general terms, this corresponds to overfitting, but note that it is an evaluation of output generated from the prior $p(z)$, not the reconstruction of test data. It thus has little to do with likelihood. This is mainly due to the mismatch between the prior $p(z)$ and the aggregated posterior $q_\phi(z) = \int q_\phi(z|x)p(x)dx$, i.e., the gap between sampling in evaluation and reconstruction in training. To solve this, different choices of the distribution of the prior [39] or hierarchical VAE [9] have been introduced, but the simplest is beta-VAE [14]. Beta-VAE increases the influence of KL divergence as in Equation (3), so that $q(z|x)$ matches $p(z)$ even if the parameter deviates from the optimal likelihood. This is a good way to resolve poor sampling because it helps to approach $q_\phi(z) = p(z)$ practically [5].

We provide the example in Fig. 1. Model A is an example of poor reconstruction, trained with constant σ_x^2 and high β $(= 10)$. This model shows low likelihood, but the quality of reconstruction and generation is consistent. Model B is an example of poor generation, adopting learnable σ_x^2 without β $(= 1)$. This model demonstrates high likelihood, but the generation is relatively blurry and unrealistic. Their setup is identical to the one in the experiment. Since a model can only do one side well, we must distinguish between the two when approaching the blurry output problem.

3.2 Problem Investigation

Prior works and implementations practically assume constant variance building the decoder to output mean μ_x [34,41]. This is problematic due to degraded performance and is further complicated by the introduction of β. We first explore the situation in which the variance and β are equal. Specifying the distribution as Gaussian allows us to expand ELBO further. The reconstruction error, shown in Equation (1), is expanded as Equation (4).

$$-\log p_\theta(x_i|z) = \frac{(x_i - \mu_x(z))^2}{2\sigma_x^2(z)} + \frac{1}{2}\log 2\pi\sigma_x^2(z) \tag{4}$$

The log-sigma term on the right can be ignored in optimization if the variance is constant. Considering the beta-VAE with $\sigma_x^2 = 1/2$, then the β of it mathematically equal to the $2\sigma_x^2$ in conventional VAE up to a constant multiplier [9,34], i.e., with a learning rate adaptation. This stems from the fact that the two objectives are identical in their form. Here we present a slightly more general relationship between β and the variance in the same fashion, indicated in Equation (5) in which previously claimed equality is a special case of $C = 1/2$.

β as constant decoder variance. For the Gaussian beta-VAE with variance $\sigma_x^2 = C$ and conventional Gaussian VAE with variance $\sigma_x^2 = \beta{\cdot}C$ where C is a constant scalar, the gradients of their objectives are identical up to a constant multiplier β, as indicated in Equation (5). Hence, they are the same model in terms of neural network training, and the last $\equiv$ symbol in Equation (5) implies this. Note the subtlety that C on the left is the variance of beta-VAE, and σ_x^2 on the right is of a general VAE.

$$\begin{aligned}
& L_\beta(\theta,\phi,x_i,\sigma_x^2) \\
= \quad & E_{z\sim q_\phi(z|x_i)}\left[-\log p_\theta(x|z)\right] + \beta D_{KL}(q_\phi(z|x_i)||p(z)) \\
= \quad & E_{z\sim q_\phi(z|x_i)}\left[\frac{(x_i-\mu_x(z))^2}{2\sigma_x^2(z)} + \frac{1}{2}\log 2\pi\sigma_x^2(z)\right] + \beta D_{KL}(q_\phi(z|x_i)||p(z)) \\
= \quad & E_{z\sim q_\phi(z|x_i)}\left[(x_i-\mu_x(z))^2\right]/2\sigma_x^2 + \beta D_{KL}(q_\phi(z|x_i)||p(z)) + O(\log\sigma_x^2) \\
& -\mathrm{ELBO}(\theta,\phi,x_i,\sigma_x^2) \\
= \quad & E_{z\sim q_\phi(z|x_i)}\left[-\log p_\theta(x|z)\right] + D_{KL}(q_\phi(z|x_i)||p(z)) \\
= \quad & E_{z\sim q_\phi(z|x_i)}\left[\frac{(x_i-\mu_x(z))^2}{2\sigma_x^2(z)} + \frac{1}{2}\log 2\pi\sigma_x^2(z)\right] + D_{KL}(q_\phi(z|x_i)||p(z)) \\
= \quad & E_{z\sim q_\phi(z|x_i)}\left[(x_i-\mu_x(z))^2\right]/2\sigma_x^2 + D_{KL}(q_\phi(z|x_i)||p(z)) + O(\log\sigma_x^2) \\
\Rightarrow \quad & \nabla L_\beta(\theta,\phi,x_i,C) = -\beta\nabla\mathrm{ELBO}(\theta,\phi,x_i,\beta\cdot C) \\
\Rightarrow \quad & \beta\cdot C \equiv \sigma_x^2
\end{aligned} \tag{5}$$

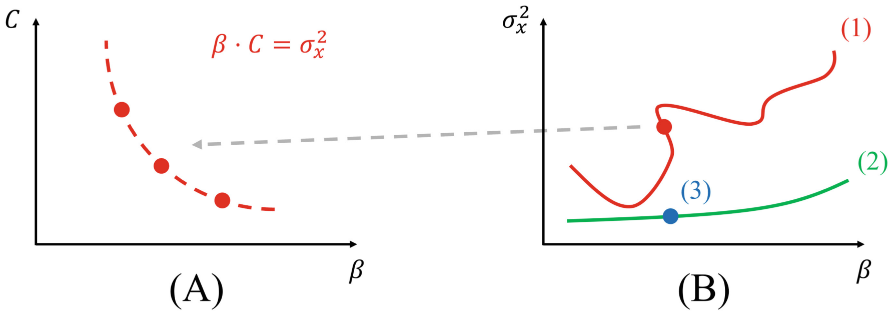


Fig. 2. The conceptual figure of optimizing σ_x^2 and β. **(A)** The dashed line indicates a constant σ_x^2 beta-VAE with same weights. Since the single integrated parameter $\beta \cdot C \equiv \sigma_x^2$ is set, researchers can arbitrarily choose β and C values for a σ_x^2. This harms VAE research by the inconsistency. **(B)** (1) A typical VAE cannot control each parameter. β has almost no function beyond tuning σ_x^2 here. (2) Our method can tune the β value while maintaining a reasonably low σ_x^2 value for the best likelihood. (3) The existing model with learnable decoder variance cannot adjust β, so it only represents a single point .

The only value we can set in beta-VAE is the integrated parameter $\beta \cdot C$, not separate β or σ_x^2, as they compensate each other. It means that introducing β has almost no effect beyond tuning σ_x^2 as long as we use constant decoder variance, since it is completely absorbed in the variance. This not only negates the performance gain of β but also makes the likelihood inconsistent, blocking meaningful model analysis.

First, in the setting, decoder variance can be an arbitrary value. As given in Equation (5) and discussed in some works [9,34], if we consider the beta-VAE variance as $C = 1/2$, then $\sigma_x^2 = \beta/2$, leading to the consistent likelihood. However, most researchers treat β as an isolated hyperparameter and calculate the likelihood from the beta-VAE variance C. This leaves the variance value to the researcher's discretion, as indicated in Fig. 2A. Consequently, studies that describe β without specifying C or code are not reproducible.

Worse still, the arbitrary variance introduces uncertainty in likelihood, since the reconstruction error is determined by σ_x^2 as in Equation (4). This causes critical confusion in model analysis because the likelihood, which is a key value in the maximum likelihood estimation model, becomes inconsistent. For instance, constant variance beta-VAE has been usually considered as either $C = 1/2$ or $C = \beta/2$ for the model with the same objective, or even parameters. The (lower bound of) log-likelihoods in each setting can be drastically different, so VAE studies that exhibit similar human-perceptual performance often show likelihood from -10^6 to 10^6, making comparison virtually impossible.

Also, it is important to note that the goals of beta-VAE are different from those of conventional VAE. The beta-VAE is not the technique for obtaining the highest likelihood, but rather securing disentanglement or quality generation [5, 6,11,14]. It is evident from the very introduction of β, which makes the objective no longer the likelihood as in Equation (3). However, the gradient of the constant σ_x^2 model is still within the likelihood, as demonstrated in Equation (5). It does not lead to the benefits that only β can achieve. Only the integrated parameter $\beta{\cdot}C \equiv \sigma_x^2$ is set, preventing control of each parameter. In this context, the role of β is limited to adjusting σ_x^2, and the optimality of σ_x^2 and β cannot be achieved. We depict it in Fig. 2A and Fig. 2B-1.

This inseparability of the variance and β have confounded their respective effect. For example, researchers pursuing sharp generation ought to reduce the variance to increase likelihood [41]. Many implementations, in fact, have chosen small βs (indeed, $\beta{\cdot}C \equiv \sigma_x^2$) to diminish the blurriness of generation. The optimal σ_x^2 and the optimal β are different. The optimal σ_x^2 is arguably the maximizer of likelihood, but the optimal β depends on the purpose. In [34] dealing with similar confusion, they have pointed out the pervasive imprecise implementation of σ_x^2, but their claim that the optimal σ_x^2 is also the optimal β is incorrect. Such confusion not only harms the practical performance of VAE but also the theoretical analysis of VAE.

A natural approach to address the limitation of integrated parameter $\beta \cdot C \equiv \sigma_x^2$ is to separate the two parameters. Since the constant variance beta-VAE cannot achieve the aim, we employ the learnable variance beta-VAE. Still, implementing the learnable decoder variance poses an optimization problem [31]. We first analyze how the objective behaves in the setting.

When the variance of decoder is considered as the trainable parameter, σ_x^2 and β are distinct to each other, as the gradient of the objective changes. The key to the distinction is the log-sigma term in Equation (4). In this setting, Equation (5) does not hold since the log-sigma term is not constant. The log-sigma term is derived from the normalizing factor of Gaussian probability density function, allowing the decoder function to remain as a probability distribution. Letting the variance change rather than constant enhances the expressiveness of model, but the distribution becomes uncontrollable if the variance converges to 0 or ∞.

In optimization, the log-sigma term prevents the infinitely large σ_x^2 to reduce the objective [24]. A large variance compensates for the error arising from prediction failure, as illustrated in Equation (4), hence σ_x^2 may diverge to infinity without the log-sigma term. Namely, the log-sigma term encourages the model to learn a large σ_x^2 for challenging samples and a small σ_x^2 for easier ones. Consequently, the variance represents an uncertainty, making it reasonable that its value decreases as training progresses, even if it approaches 0. This leads to the unstable optimization caused by the zero variance. Indeed, it has been claimed that this infinite gradient helps in achieving the optimal latent structure [8].

Although it intuitively or theoretically makes sense, unstable optimization is undesirable for practical uses. A few works [34,38] have provided implementations for the stable decoder with learnable variance exploiting the property of Gaussian, which we employ in our method.

3.3 Method

We propose a method to separate the variance of decoder and β, simply introducing β with learnable variance. To maintain stable optimization, we first adopt the optimal variance.

Optimal decoder variance σ_x^2. For the reconstruction error of a Gaussian VAE (Equation (4)), a single sample x_i, and its sampled latent z_i, we can find a analytical optimal $\sigma_x^{2^*}(z_i)$ for a given $(x_i - \mu(z_i))^2$.

$$\begin{aligned}
& \frac{\partial}{\partial \sigma_x}[-\mathrm{ELBO}(\theta, \phi, x_i, z_i)] \\
= \quad & \frac{\partial}{\partial \sigma_x}[-\log p_\theta(x_i|z_i) + D_{KL}(q_\phi(z|x_i)||p(z))] \\
= \quad & \frac{\partial}{\partial \sigma_x}[\frac{(x_i - \mu_x(z_i))^2}{2\sigma_x^2(z_i)} + \frac{1}{2}\log 2\pi\sigma_x^2(z_i) + O(1)] \\
= \quad & -\frac{(x_i - \mu_x(z_i))^2}{\sigma_x^3(z_i)} + \frac{1}{\sigma_x(z_i)} \quad = \quad 0 \\
\Rightarrow \quad & \sigma_x^{2^*}(z_i) = (x_i - \mu_x(z_i))^2
\end{aligned}$$

This is an alternative to directly implementing trainable variance [34,41]. We employ this because it is mathematically clear and easy to implement.

Albeit it has been argued as the method to find the optimal β [34], according to our claim, the optimal σ_x^2 is not identical to the optimal β. Rather, the Gaussian VAE with optimal decoder variance is not associated with β, i.e., $\beta = 1$, as demonstrated in Fig. 2B-3. σ_x^2 and β should be taken as different parameters.

$$\begin{aligned}
L_{\beta\sigma}(\theta, \phi) = \frac{1}{2}E_{z\sim q_\phi(z|x)}&[\log 2\pi(x - \mu_x(x))^2 + 1] \\
&+ \beta D_{KL}(q_\phi(z|x)||p(z))
\end{aligned} \tag{6}$$

Then β can be reintroduced into the optimal σ_x^2 model. As a result, we build a new objective named Beta-Sigma VAE (BS-VAE) as shown in Equation (6). Although it looks like a straightforward and simple extension, BS-VAE achieves the control of each parameter, as illustrated in Fig. 2B-2. It takes advantage of both parameters and ensures that the same model and weights always provide the same likelihood value. It also shows significant performance improvement over prior works in our experimental evaluation.

4 Experimental Evaluation

4.1 Evaluation Setup

We train and compare BS-VAEs and typical beta-VAEs with constant σ_x^2, which provide empirical evidence of our proposition. First, to reveal the ambiguity of reconstruction error, we visualize the rate-distortion curve, which exhibits the performance of each model as a point on the curve. The proposed BS-VAE draws a single curve. On the other hand, conventional beta-VAE with constant decoder variance has multiple interpretations along the fixed variance values and corresponding distortion of the curve. We test in three different ways: $\sigma_x^2 = 1/2$, $\sigma_x^2 = \beta/2$, which are the views often adopted in previous research, and the case with optimal σ_x^2, which is the upper bound of beta-VAE performance interpretation. Secondly, we evaluate the VAEs based on proxy metrics, i.e. Fréchet inception distance [13] (FID) and log-likelihood on unseen data. Although likelihood is a good indicator of generative model and it directly measures the optimization of VAE, generation is difficult to be evaluated in a single figure. For example, a fully memorized model, i.e., a lossless compression system, achieves an infinite log-likelihood on training set, ignoring important values such as diversity. Thus the proxy metrics are convincing indicators by preventing the model from simply remembering the training data. To improve FID, β of beta-VAE has been adjusted by practitioners at the cost of likelihood frequently. log-likelihood on unseen data has been used as an indicator for generalization capability in previous works [37,39]. Additionally, to evaluate generative neural networks, we conduct a qualitative evaluation of generated samples.

All models consist of a Gaussian encoder with diagonal covariance matrix and a Gaussian decoder. We employ common shallow convolutional neural network architecture with a residual connection to implement VAEs for our experiments. They are evaluated on popular computer vision datasets, CelebA [22] and MNIST [21]. They consist of 4-layer residual block encoder and 4-layer convolutional decoder with 64 latent channels to train CelebA dataset. MNIST test networks are simplified to have 3 layers for each encoder and decoder with 32 latent channels. We train each model for 50 epochs using AdamW optimizer [23] and evaluate them on the fully trained model. For more specific settings, see https://github.com/overnap/BS-VAE.

As the evaluation is for proof-of-concept, it is conducted on relatively shallow neural networks and light datasets. We emphasize that BS-VAE is applicable to most VAE-variants, because our argument is about the parameterization of Gaussian VAE, independent of architecture and scale. However, it is difficult to ensure that it applies to the larger architecture using VAE as a part, such as latent diffusion model [33]. The discussion about it is an interesting future work.

4.2 Experimental Results

We train VAEs on CelebA with β scaled from 0.0001 to 1000, which is wide enough for common use. We evaluate ELBO of the models and plot their rate-distortion curves as summarized in Fig. 3. BS-VAEs (red crosses) outperform two types of constant variance beta-VAEs (blue circles and orange x). beta-VAEs as $\sigma_x^2 = 1/2$ appear to fall short in drawing the desired rate-distortion trade-off. In the $\sigma_x^2 = \beta/2$ case, distortions are significantly high compared to our model in low rate cases. As the rate decreases, the performance gap between $\sigma_x^2 = \beta/2$ case and ours becomes larger. This is a critical drawback of beta-VAEs since VAE naturally pursues to reduce the rate (i.e., KL divergence) in training to

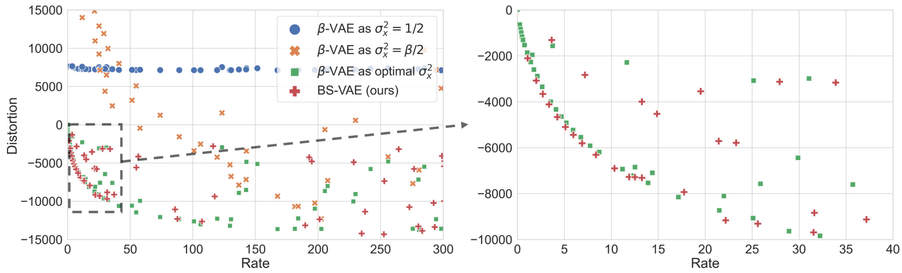


Fig. 3. The rate-distortion curve plotting BS-VAEs and conventional beta-VAEs with constant σ_x^2. The constant variance can be interpreted in various ways, so the optimal σ_x^2 that leads distortion to the lower bound and two common σ_x^2s are indicated. BS-VAEs outperform the conventional models by any interpretation of σ_x^2 .

satisfy given tasks. This can be explained by Equation (5). Assuming $C = 1/2$, $\beta = 2\sigma_x^2$ holds through learning rate adaptation. Extended to the trainable σ_x^2 VAE, the equation no longer holds, and the more delicate relationship between β and σ_x^2 is disclosed by the same development.

$$\beta = 2\sigma_x^2(z) + \frac{\sigma_x^2(z) \log 2\pi\sigma_x^2(z)}{D_{KL}(q_\phi(z|x)||p(z))}$$

Notably, the influence of the log-sigma term, governed by the KL divergence term in its denominator, increases as the KL divergence diminishes, explaining the performance gap clearly.

Proposed BS-VAEs outperform compared to the constant variance beta-VAEs as optimal σ_x^2 (green squares) in Fig. 3. The constant variance models evaluated with optimal σ_x^2 represent the upper bound for their likelihood. Therefore, BS-VAEs generally achieves better performance than typical beta-VAEs regardless of the interpretation of σ_x^2, by leveraging both parameters. Previous studies have shown similar results only at certain β, especially near the optimal σ_x^2 value [8,34].

We train VAEs with β from 0.01 to 100 on CelebA and MNIST and present their proxy metrics in Table 1. The models are trained five times each, and the results are shown with their means. Note that ELBO is calculated instead of the direct log-likelihood. For constant σ_x^2 models, the lower bound of ELBO is shown for meaningful comparison, i.e., assuming optimal σ_x^2. Otherwise, there is much of a gap like the left of Fig. 3, e.g., $\log p_\theta(x) = -8000$.

In both datasets, BS-VAEs demonstrate better performance than constant σ_x^2 models where $\beta = 1$. Note that lower FID and higher likelihood indicate better performance in the tasks. Furthermore, BS-VAEs with $\beta = 1$ show better performance compared to the constant variance models over the entire β range. These results concur with those reported in previous studies: BS-VAE with $\beta = 1$ is conceptually identical to [8] and implementationally identical to [34]. We thus claim that the improvement comes from the benefit of learnable decoder variance rather than any implementation-specific gain.

As illustrated in BS-VAEs with $\beta \neq 1$ in Table 1, we obtain learnable variance models with various βs by the reintroduction of β into the optimal variance model. They all attain better FID scores compared to the constant models for the same β. As the good proxy metric is a goal of tuning β, the empirical best β for our model is 10, exhibiting significant performance gain. This naturally disproves the previous claim that the optimal σ_x^2 means the optimal β [34]. Even in the optimal variance model, β can be adjusted to achieve better proxy metrics or latent disentanglement. Moreover, BS-VAEs attain the best likelihood at $\beta = 1$ where the objective remains as likelihood. This is not the case in constant models where the likelihood increases as β decreases despite the objective drifting away from the log-likelihood. These results align with our arguments in Section 3 and Fig. 2.

We display reconstructed and generated samples of these models in Fig. 4. Arguably, BS-VAEs excel in reconstruction quality regardless of the β value, meeting the basic purpose of VAE, i.e., lossy compression. A possible explanation for this is that moderate β values do not hinder the achievement of optimal latent structure [8]. In BS-VAE, varying β only changes generation quality, while the conventional VAE does not. This is because the β we adjust in the constant model, as shown in Equation (5) and Fig. 2A, is actually the integrated parameter $\beta \cdot C \equiv \sigma_x^2$. BS-VAE at $\beta = 10$ exploits both σ_x^2 and β, resulting in both good reconstruction and good generation.

Table 1. Proxy metric evaluations of BS-VAEs and constant decoder variance beta-VAEs with various βs. The FID [13] and the log-likelihood on test set are shown with the common log-likelihood for reference. The models are trained five times each, showing their means. BS-VAE obtains the best likelihood at $\beta = 1$ and the best FID at $\beta = 10$, demonstrating that optimal σ_x^2 does not mean optimal β

Model		CelebA			MNIST		
Name	β	FID (↓)	Test $\log p_\theta(x)$	$\log p_\theta(x)$	FID (↓)	Test $\log p_\theta(x)$	$\log p_\theta(x)$
Beta-VAE with constant σ_x^2	0.01	194.7	$>$ **10684**	$>$ **10762**	190.3	$>$ **667**	$>$ **659**
	0.1	151.7	$>$ 10384	$>$ 10412	**163.6**	$>$ 626	$>$ 618
	1	**126.4**	$>$ 10616	$>$ 10626	225.8	$>$ 291	$>$ 286
	10	149.4	$>$ 6923	$>$ 6898	351.7	$>$ -19	$>$ -20
	100	235.8	$>$ 2233	$>$ 2190	352.5	$>$ -19	$>$ -20
BS-VAE (Ours)	0.01	188.5	$>$ 10772	$>$ 10848	67.4	$>$ 796	$>$ 788
	0.1	130.2	$>$ 12996	$>$ 13037	75.5	$>$ 850	$>$ 840
	1	90.8	$>$ **14384**	$>$ **14434**	59.2	$>$ **887**	$>$ **877**
	10	**73.7**	$>$ 13205	$>$ 13256	**38.4**	$>$ 662	$>$ 656
	100	106.2	$>$ 7668	$>$ 7630	332.8	$>$ -15	$>$ -17

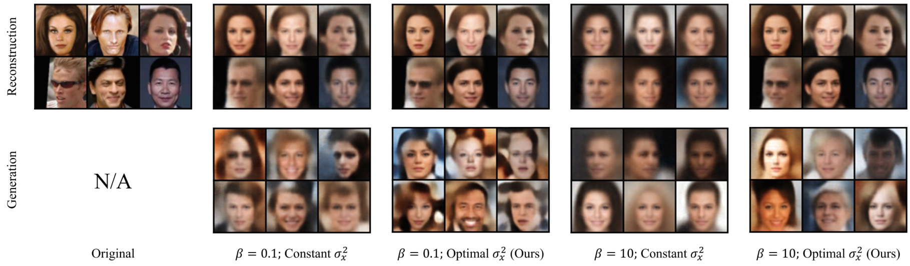

Reconstruction
Generation
N/A
Original
β = 0.1; Constant σ_x^2
β = 0.1; Optimal σ_x^2 (Ours)
β = 10; Constant σ_x^2
β = 10; Optimal σ_x^2 (Ours)

Fig. 4. Reconstructed or generated samples of common beta-VAEs with constant decoder variance and our BS-VAEs. Our models maintain good reconstruction quality within tested βs. The samples are selected without any intention, i.e., no cherry picking .

5 Conclusion

We investigated and addressed the blurry output problem of VAE. In particular, we elucidated the confusion between the variance of Gaussian decoder σ_x^2 and β of beta-VAE [14]. We also proposed BS-VAE to handle the indistinguishability problem of beta-VAE with constant decoder variance. Our BS-VAE is simple but explicitly separates the σ_x^2 and β, demonstrating competitive performance over prior work with predictable and meaningful analysis. We expect that the following research avoids ambiguity and obtains optimal VAE performance in applications.

Acknowledgements. This work was supported by the National Research Foundation of Korea (NRF) grant (No.NRF-2020R1A2C1015146) and the IITP (Institute of Information & Communications Technology Planning & Evaluation) grant (National Program for Excellence in SW, 2023-0-00042 in 2024) funded by the Korea government (Ministry of Science and ICT).

References

1. Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R.A., Murphy, K.: Fixing a broken elbo. In: ICML. pp. 159–168. PMLR (2018)
2. Bae, J., Zhang, M.R., Ruan, M., Wang, E., Hasegawa, S., Ba, J., Grosse, R.B.: Multi-rate vae: Train once, get the full rate-distortion curve. In: ICLR (2023)
3. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. In: Proceedings of The 20th SIGNLL CoNLL. p. 10. Association for Computational Linguistics (2016)
4. Bozkurt, A., Esmaeili, B., Tristan, J.B., Brooks, D., Dy, J., van de Meent, J.W.: Rate-regularization and generalization in variational autoencoders. In: AISTATS. pp. 3880–3888. PMLR (2021)
5. Burgess, C.P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in β-vae. arXiv preprint arXiv:1804.03599 (2018)

6. Chen, R.T., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. NeurIPS **31** (2018)
7. Child, R.: Very deep vaes generalize autoregressive models and can outperform them on images. arXiv preprint arXiv:2011.10650 (2020)
8. Dai, B., Wenliang, L., Wipf, D.: On the value of infinite gradients in variational autoencoder models. NeurIPS **34**, 7180–7192 (2021)
9. Dai, B., Wipf, D.: Diagnosing and enhancing vae models. In: ICLR (2018)
10. Duan, Z., Lu, M., Ma, Z., Zhu, F.: Lossy image compression with quantized hierarchical vaes. In: Proceedings of the IEEE/CVF WACV. pp. 198–207 (2023)
11. Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D.H., Dy, J., Meent, J.W.: Structured disentangled representations. In: AISTATS. pp. 2525–2534. PMLR (2019)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS **27** (2014)
13. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS **30** (2017)
14. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: ICLR (2016)
15. Hoffman, M.D., Riquelme, C., Johnson, M.J.: The β-vae's implicit prior. In: Workshop on Bayesian Deep Learning, NIPS. pp. 1–5 (2017)
16. Huang, S., Makhzani, A., Cao, Y., Grosse, R.: Evaluating lossy compression rates of deep generative models. In: ICML. pp. 4444–4454. PMLR (2020)
17. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
18. Koehler, F., Mehta, V., Zhou, C., Risteski, A.: Variational autoencoders in the presence of low-dimensional data: landscape and implicit bias. arXiv preprint arXiv:2112.06868 (2021)
19. Kohl, S., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J.R., Maier-Hein, K., Eslami, S., Jimenez Rezende, D., Ronneberger, O.: A probabilistic u-net for segmentation of ambiguous images. NeurIPS **31** (2018)
20. Kunin, D., Bloom, J., Goeva, A., Seed, C.: Loss landscapes of regularized linear autoencoders. In: ICML. pp. 3560–3569. PMLR (2019)
21. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
22. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE ICCV. pp. 3730–3738 (2015)
23. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2018)
24. Lucas, J., Tucker, G., Grosse, R.B., Norouzi, M.: Don't blame the elbo! a linear vae perspective on posterior collapse. NeurIPS **32** (2019)
25. Mattei, P.A., Frellsen, J.: Leveraging the exact likelihood of deep latent variable models. NeurIPS **31** (2018)
26. Nair, A.V., Pong, V., Dalal, M., Bahl, S., Lin, S., Levine, S.: Visual reinforcement learning with imagined goals. NeurIPS **31** (2018)
27. Norouzi, S., Fleet, D.J., Norouzi, M.: Exemplar vae: Linking generative models, nearest neighbor retrieval, and data augmentation. NeurIPS **33**, 8753–8764 (2020)
28. Pong, V.H., Dalal, M., Lin, S., Nair, A., Bahl, S., Levine, S.: Skew-fit: state-covering self-supervised reinforcement learning. In: ICML. pp. 7783–7792 (2020)
29. Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. NeurIPS **32** (2019)

30. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: ICML. pp. 1278–1286. PMLR (2014)
31. Rezende, D.J., Viola, F.: Taming vaes. arXiv preprint arXiv:1810.00597 (2018)
32. Rolinek, M., Zietlow, D., Martius, G.: Variational autoencoders pursue pca directions (by accident). In: Proceedings of the IEEE/CVF CVPR. pp. 12406–12415 (2019)
33. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF CVPR. pp. 10684–10695 (2022)
34. Rybkin, O., Daniilidis, K., Levine, S.: Simple and effective vae training with calibrated decoders. In: ICML. pp. 9179–9189. PMLR (2021)
35. Shekhovtsov, A., Schlesinger, D., Flach, B.: Vae approximation error: Elbo and exponential families. In: ICLR (2021)
36. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML. pp. 2256–2265. PMLR (2015)
37. Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. NeurIPS **29** (2016)
38. Takahashi, H., Iwata, T., Yamanaka, Y., Yamada, M., Yagi, S.: Student-t variational autoencoder for robust density estimation. In: IJCAI. pp. 2696–2702 (2018)
39. Tomczak, J., Welling, M.: Vae with a vampprior. In: AISTATS. pp. 1214–1223. PMLR (2018)
40. Wipf, D.: Marginalization is not marginal: no bad vae local minima when learning optimal sparse representations. In: ICML. pp. 37108–37132. PMLR (2023)
41. Yu, R.: A tutorial on vaes: From bayes' rule to lossless compression. arXiv preprint arXiv:2006.10273 (2020)

HingeRLC-GAN: Combatting Mode Collapse with Hinge Loss and RLC Regularization

Osman Goni[1](✉), Himadri Saha Arka[1], Mithun Halder[1],
Mir Moynuddin Ahmed Shibly[1], and Swakkhar Shatabda[2]

[1] Department of Computer Science and Engineering, United International University, United City, Madani Avenue, Dhaka 1212, Bangladesh
{ogoni202046,harka202008,mhalder201041}@bscse.uiu.ac.bd,
moynuddin@cse.uiu.ac.bd

[2] Department of Computer Science and Engineering, BRAC University, Kha 224, Bir Uttam Rafiqul Islam Avenue. Merul Badda, Dhaka 1212, Bangladesh
swakkhar.shatabda@bracu.ac.bd

Abstract. Recent advances in Generative Adversarial Networks (GANs) have demonstrated their capability for producing high-quality images. However, a significant challenge remains mode collapse, which occurs when the generator produces a limited number of data patterns that do not reflect the diversity of the training dataset. This study addresses this issue by proposing a number of architectural changes aimed at increasing the diversity and stability of GAN models. We start by improving the loss function with Wasserstein loss and Gradient Penalty to better capture the full range of data variations. We also investigate various network architectures and conclude that ResNet significantly contributes to increased diversity. Building on these findings, we introduce HingeRLC-GAN, a novel approach that combines RLC Regularization and the Hinge loss function. With a FID Score of 18 and a KID Score of 0.001, our approach outperforms existing methods by effectively balancing training stability and increased diversity.

Keywords: GAN · Diversity · Mode Collapse · Hinge Loss · Regularization · ResNet · Fréchet inception distance (FID) · Kernel Inception Distance (KID)

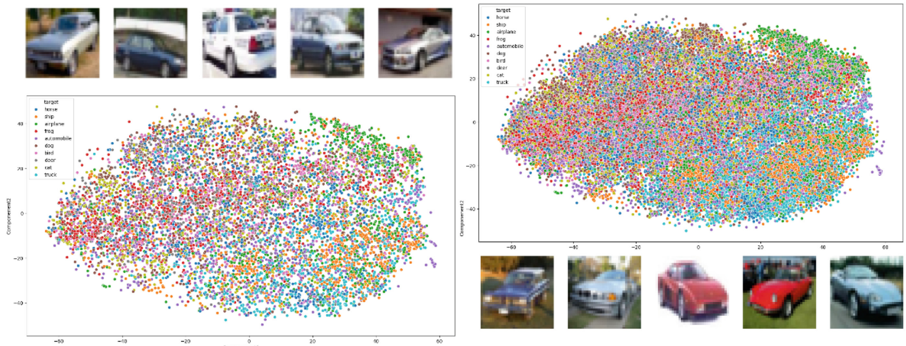
1 Introduction

Generative Adversarial Networks (GANs) [5] have made remarkable strides in generating high-fidelity images. These models are foundational for many vision

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78389-0_25.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 370–385, 2025.
https://doi.org/10.1007/978-3-031-78389-0_25

applications, including data augmentation [1,21], domain adaptation [8,23], image extrapolation [24], image-to-image translation [9,25], and image editing [19,20]. However, the success of GANs often hinges on the availability of large, diverse training datasets, which can be costly and labor-intensive to compile [18].



DROPOUT-GAN with 60% of Mode Coverage

HingeRLC-GAN with up to 88% of Mode Coverage

Fig. 1. Mode Coverage: DROPOUT-GAN (left),HingeRLC-GAN (right). The proposed method is performing up to 30% better in mode capture

A significant challenge associated with GANs is mode collapse, where the generator produces a limited variety of outputs and fails to capture the full diversity of the data distribution [2,12]. This issue drastically reduces the diversity of generated data, limiting the utility of GANs across different applications. Mode collapse is exemplified in Figure 1, which demonstrates how a generator may inadequately represent the modes of the data distribution.

To address the issue of mode collapse in GANs, extensive research has explored a variety of new techniques. These include advanced loss functions, such as Wasserstein loss with Gradient Penalty [6,9], which have shown promise in stabilizing GAN training and enhancing output diversity by overcoming the limitations of traditional loss functions. Research into multi-generator models, like MAD GAN [4], aims to improve diversity by utilizing multiple generators in conjunction with a single discriminator. Additionally, innovative methods, such as employing orthogonal vectors to address mode collapse [10], have further advanced our understanding of GANs and their training challenges.

In this paper, we focus on improving GAN performance to enhance mode coverage, particularly for small datasets. Our approach consists of several stages:

- First, we evaluated various GAN architectures to identify the most effective structures for generating diverse outputs.
- Next, we examined different loss functions and regularization techniques to find the optimal combination for mitigating mode collapse.

- Finally, we developed a model that integrates RLC regularization [22] with Hinge Loss [11], and performed a comprehensive comparison of our HingeRLC-GAN against conventional GAN models.

Our research aims to determine the most effective architectural components, loss functions, and regularization techniques to improve mode coverage and produce high-quality, diverse synthetic images, especially when working with small datasets.

2 Related Work

Arnab Ghosh et al. introduced MAD-GAN, a model that leverages multiple generators and a single discriminator to improve sample diversity [4]. In this setup, the discriminator not only distinguishes real from fake samples but also identifies which generator produced each fake sample, promoting a wider range of generated outputs.

Wei Li et al. developed a method employing orthogonal vectors to address mode collapse in multi-generator frameworks [10]. Their approach involves extracting feature vectors from generator outputs and minimizing their orthogonality to preserve diversity, using a new minimax formula to enhance convergence and balance.

Jae Hyun Lim and Jong Chul Ye proposed Geometric GAN, which redefines adversarial training through geometric steps involving hyperplane separation to overcome issues such as vanishing gradients and instability [11]. This SVM-inspired method improves the reliability and efficiency of training.

Mordido et al. introduced Dropout-GAN, which applies dropout regularization to a discriminator ensemble to combat overfitting and maintain diversity in generated samples [14]. Dropout-GAN demonstrates superior performance compared to other variants by generating diverse and realistic data while minimizing the Fréchet distance.

Sen Pei et al. presented the Pluggable Diversity Penalty Module (PDPM), which enforces diversity in the feature space using normalized Gram matrices [17]. PDPM achieves outstanding results across various tasks, surpassing traditional methods such as ALI, DCGAN, and MSGAN.

Pan et al. introduced UniGAN, which aims to address u-mode collapse by focusing on uniform diversity [16]. This model employs a generator based on Normalizing Flow and a regularization technique to ensure uniform output diversity, allowing for seamless integration with other frameworks.

3 Proposed Method

First, we examine the GAN architecture, focusing on its core components and overall structure. Next, we review the loss functions used during GAN training, emphasizing their roles and how they influence the model's performance.

We then delve into regularization techniques, assessing their impact on stabilizing training and improving the model's generalization capabilities. Finally, we explore the effectiveness of the HingeRLC-GAN by investigating its architectural modifications and their contributions to enhancing diversity and training stability.

3.1 Architectural Overview

Generative Adversarial Networks (GANs) are a class of machine learning frameworks designed for generating realistic data. A GAN comprises two neural networks: the Generator G and the Discriminator D. These networks are trained simultaneously in a competitive framework. The Generator creates synthetic data with the goal of approximating real data, while the Discriminator's task is to distinguish between real and generated data. The GAN framework involves training the Generator G and Discriminator D through a minimax game, which is formalized as follows:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

In this formulation, the Generator G aims to minimize the likelihood of the Discriminator correctly identifying generated data as fake, while the Discriminator D seeks to maximize its ability to differentiate between real and generated data. Here, $p_{\text{data}}(x)$ represents the distribution of real data, and $p_z(z)$ represents the distribution of the input noise vector z. We have experimented with various architectures such as DenseNet, MobileNet, and EfficientNet, and found that the ResNet architecture consistently produces superior results.

Generator The Generator architecture is constructed using ResNet blocks, which incorporate residual connections to support effective gradient flow. The detailed architecture is as follows:

$$\text{Noise } z \xrightarrow{\text{Linear}(128\to 4\times 4\times 128)} \xrightarrow{\text{Reshape}(-1,128,4,4)} \xrightarrow{\text{ResNet Block}_1} \xrightarrow{\text{ResNet Block}_2}$$

$$\xrightarrow{\text{ResNet Block}_3} \xrightarrow{\text{BatchNorm2d}} \xrightarrow{\text{ReLU}} \xrightarrow{\text{Conv2d}(3,3,\text{padding}='same')} \xrightarrow{\text{Tanh}} \xrightarrow{\text{Output Image}}$$

Each ResNet Block consists of:

$$\begin{gathered}\text{CCBN}(128, 10) \to \text{ReLU} \to \text{Upsample}\\ (\text{scale factor} = 2) \to \text{Conv2d}(128, 3, \text{padding} =' same')\end{gathered}$$

Here, $\text{CCBN}(128, 10)$ denotes Conditional Batch Normalization with 128 channels and 10 conditions. The upsampling layer increases the spatial dimensions of the feature maps, and the convolutional layer with a kernel size of 3 and 'same' padding ensures that the output maintains the required dimensions.

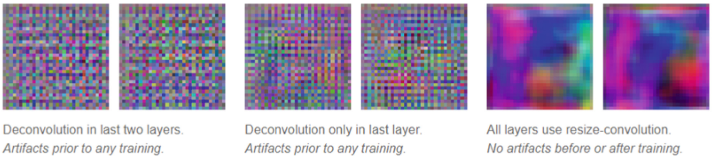

Deconvolution in last two layers.
Artifacts prior to any training.
Deconvolution only in last layer.
Artifacts prior to any training.
All layers use resize-convolution.
No artifacts before or after training.

Fig. 2. Artifact produced by 'Conv2dTranspose' layers with checkerboard patterns

Avoiding the use of 'Conv2dTranspose' layers is crucial for minimizing artifacts in generated images. 'Conv2dTranspose' layers [15] with a stride of 2 are commonly used for upsampling, effectively doubling the size of the images. However, they can introduce artifacts such as checkerboard patterns due to uneven overlapping of the layers. This problem arises from the inherent characteristics of the 'Conv2dTranspose' operation rather than from adversarial training itself.

To address this issue, our architecture utilizes 'Upsample' layers followed by 'Conv2d' layers for upsampling. This approach avoids the artifacts typically associated with 'Conv2dTranspose' layers. Figure 2 illustrates artifacts produced by a generator using 'Conv2dTranspose' in the last two layers. By employing 'Upsample' layers combined with 'Conv2d' layers, we achieve cleaner, artifact-free images and improved overall image quality.

Discriminator The Discriminator uses ResNet blocks with downsampling to classify images. Its architecture is as follows:

$$\text{Image } x \xrightarrow{\text{Concatenate(Embedding } y \text{ to } 32\times32)} \xrightarrow{\text{Conv2d(128,3,padding=}'\textit{same}')} \xrightarrow{\text{ReLU}}$$

$$\xrightarrow{\text{Conv2d(128,3,padding=}'\textit{same}')} \xrightarrow{\text{AvgPool2d(2,2)}} \xrightarrow{\text{ResNet Block Down}_1} \xrightarrow{\text{ResNet Block Down}_2}$$

$$\xrightarrow{\text{ResNet Block Down}_3} \xrightarrow{\text{AdaptiveMaxPool2d}} \xrightarrow{\text{Flatten}} \xrightarrow{\text{Linear(1)}} \xrightarrow{\text{Output Score}}$$

Each ResNet Block Down consists of:

$$\text{CCBN}(128, 10) \rightarrow \text{ReLU} \rightarrow \text{Conv2d}(128, 3, \text{padding} =' \textit{same}') \rightarrow \text{AvgPool2d}(2, 2)$$

Here, CCBN(128, 10) denotes Conditional Batch Normalization with 128 channels and 10 conditions. The 'AvgPool2d' layer performs downsampling by a factor of 2, reducing the spatial dimensions of the feature maps at each ResNet block.

The superiority of the ResNet architecture is attributed to several key components:

1. **Residual Connections**: Residual connections in ResNet blocks, defined as:

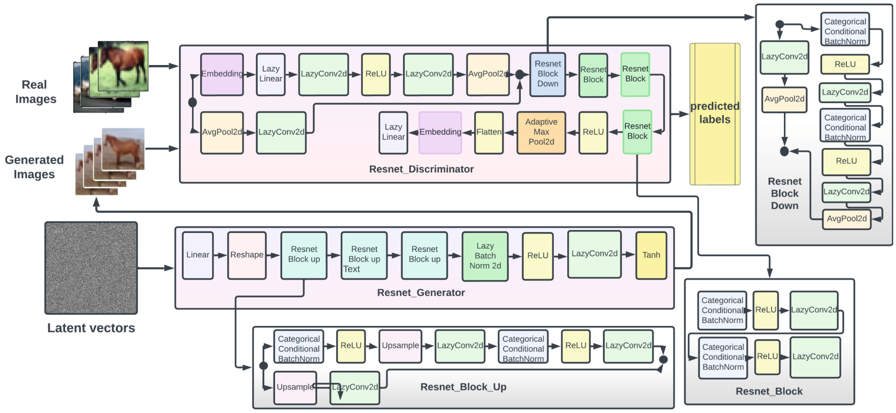


Fig. 3. HingeRLC-GAN Architecture: An illustrative example of the ResNetRLC GAN's internal generator and discriminator workings

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}$$

allow gradients to flow directly through the network, mitigating the vanishing gradient problem. This is crucial for training deeper networks, as it ensures that gradient updates from the loss function propagate effectively through many layers.

2. **Categorical Conditional Batch Normalization (CCBN)**: CCBN [3] conditions the normalization process on class labels, enabling class-specific feature generation. The normalization for each feature map i in class c is:

$$\text{BN}(x_i, \gamma_{c,i}, \beta_{c,i}) = \gamma_{c,i} \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta_{c,i}$$

where γ and β are learned parameters specific to each class, μ_i and σ_i are the mean and variance of the feature maps, and ϵ is a small constant for numerical stability.

3. **Spectral Normalization**: To enforce Lipschitz continuity, spectral normalization [13] is applied to the weights of each layer, constraining the largest singular value of the weight matrix W. This is defined as:

$$\frac{W}{\sigma(W)}$$

where $\sigma(W)$ is the largest singular value of W. Spectral normalization stabilizes the training of the Discriminator, enhancing robustness to variations in the input space.

4. **Regularization**: The architecture includes various regularization techniques, such as dropout layers, weight decay, and batch normalization, to prevent overfitting and improve generalization.

Overall, the integration of these elements results in a more stable training process and the generation of higher-quality images compared to other architectures. The ResNet-based GAN leverages deep residual learning, effective class conditioning, and robust normalization techniques to outperform models like DenseNet, MobileNet, and EfficientNet.

3.2 Loss Functions

We have experimented with several loss functions, including Binary Cross-Entropy (BCE), Wasserstein Loss, and Least Squares Loss. Our findings indicate that Hinge Loss consistently delivers superior results for our HingeRLC-GAN.

Hinge Loss is defined as:

$$\mathcal{L}_{\text{Hinge}} = \max(0, 1 - D(x)) + \max(0, 1 + D(G(z)))$$

This loss function enhances the stability of GAN training by ensuring continuous gradients, even for samples that are correctly classified. This characteristic helps mitigate the vanishing gradient problem, leading to more stable and effective training.

3.3 Regularization

In HingeRLC-GAN, we utilize several regularization techniques to enhance training stability and generalization:

1. **Noise:** Introducing noise to the inputs helps prevent the model from overfitting to the training data, promoting better generalization.
2. **Class Rebalancing:** This technique ensures that the model learns equally from all classes, thereby improving its performance across different categories.
3. **Gradient Penalty:** By encouraging smoothness in the Discriminator's decision boundary, this technique enhances the model's robustness and stability.

The primary regularization technique employed is Regularized Loss Control (RLC), defined as:

$$\mathcal{L}_{\text{RLC}} = \mathcal{L}_{\text{Hinge}} + \lambda \sum_{i} \left(\frac{\partial \mathcal{L}}{\partial \theta_i} \right)^2$$

Here, λ is a hyperparameter that regulates the strength of the regularization. RLC controls the complexity of the model by penalizing large gradients, which helps prevent overfitting and promotes better generalization.

3.4 Mathematical Intuition for Improved Mode Coverage

To illustrate how the HingeRLC-GAN mitigates mode collapse, we provide a theoretical explanation. The combination of Hinge Loss and Regularized Loss Control (RLC) fosters diverse mode coverage by penalizing the Discriminator for overly confident predictions, which encourages the Generator to explore a broader range of the data distribution.

Theoretical Framework Consider the Hinge Loss function for the Discriminator D:

$$\mathcal{L}_D = \mathbb{E}_{x\sim p_{\text{data}}}\left[\max(0, 1 - D(x))\right] + \mathbb{E}_{z\sim p_z}\left[\max(0, 1 + D(G(z)))\right]$$

The gradient of this loss with respect to the Discriminator's parameters θ_D is:

$$\nabla_{\theta_D}\mathcal{L}_D = \mathbb{E}_{x\sim p_{\text{data}}}\left[\mathbf{1}_{D(x)<1}\cdot\nabla_{\theta_D}(-D(x))\right] + \mathbb{E}_{z\sim p_z}\left[\mathbf{1}_{D(G(z))>-1}\cdot\nabla_{\theta_D}D(G(z))\right]$$

where $\mathbf{1}$ is the indicator function. The Generator G minimizes the Hinge Loss:

$$\mathcal{L}_G = -\mathbb{E}_{z\sim p_z}\left[D(G(z))\right]$$

The gradient of the Generator's loss with respect to its parameters θ_G is:

$$\nabla_{\theta_G}\mathcal{L}_G = -\mathbb{E}_{z\sim p_z}\left[\nabla_{\theta_G}D(G(z))\right]$$

The Regularized Loss Control (RLC) term is introduced to the Hinge Loss to form the total loss for the Discriminator:

$$\mathcal{L}_D^{\text{RLC}} = \mathcal{L}_D + \lambda\sum_i\left(\frac{\partial\mathcal{L}_D}{\partial\theta_{D,i}}\right)^2$$

This regularization term penalizes large gradients by adding the squared norms of the gradients to the loss function. It discourages the Discriminator from making overly confident predictions, which in turn compels the Generator to explore a more diverse set of data samples.

Prevention of Mode Collapse The integration of Hinge Loss and RLC in the HingeRLC-GAN plays a crucial role in preventing mode collapse. By discouraging the Discriminator from being too confident and smoothing its decision boundaries, the RLC term forces the Generator to explore a wider variety of data modes. This reduces the likelihood of mode collapse, where the Generator might otherwise focus on generating a limited set of samples.

4 Experimental Analysis

The Frechet Inception Distance (FID) [7] is the most commonly used metric for evaluating GAN performance. FID measures the difference between the distributions of features extracted from real and generated images using the InceptionV3 model. This metric provides a more comprehensive evaluation compared to the Inception Score, which only assesses the quality of generated images based on their own features.

FID evaluates both the mean and variance of the feature distributions from real and generated images. A lower FID indicates that the generated images are of higher quality and have better diversity, resembling the real images from the dataset, such as CIFAR-10. While FID assumes Gaussian distributions for the features, it can still be biased for smaller datasets like CIFAR-10 due to the limited sample size.

Mathematically, FID is calculated as:

$$\text{FID} = \left\|\mu_r - \mu_g\right\|^2 + \operatorname{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r\Sigma_g\right)^{1/2}\right)$$

where, μ_r and μ_g are the means of the feature vectors for real and generated images, respectively. Σ_r and Σ_g are the covariance matrices of the feature vectors for real and generated images, respectively. Tr denotes the trace of a matrix.

This formula measures the distance between two multivariate Gaussians defined by their mean vectors and covariance matrices, providing a quantitative measure of how similar the generated images are to the real ones.

The Kernel Inception Distance (KID) is a metric for evaluating GAN-generated images. Unlike Frechet Inception Distance (FID), which assumes Gaussian distributions, KID uses the squared Maximum Mean Discrepancy (MMD) to measure the distance between feature distributions of real and generated images.

$$\text{KID} = \frac{1}{2}\left(\text{MMD}^2(p_r, p_g) + \text{MMD}^2(p_g, p_r)\right)$$

where MMD is computed using a kernel function.

4.1 Comparison of GAN Architectures

In Table 1, we compare different GAN architectures using the FID score and KID Score. Each architecture varies in the network used for the generator and discriminator.

The Dense + VGG architecture shows the highest FID score of 125, indicating the poorest performance among the architectures tested. MobileNet improves the FID score to 112. EfficientNet further reduces the FID score to 97, showing better image generation quality. The best performance is observed with the ResNet architecture, achieving an FID score of 90, demonstrating its effectiveness in generating high-quality images.

Table 1. Comparison to GAN Architecture. We report the average FID (↓) scores and average KID (↓) scores on the CIFAR datasets

Architecture	Generator	Discriminator	FID↓	KID↓
Dense + VGG	Dense network + BCE	VGG + MinMax	125	0.01
MobileNet	MobileNet + BCE	MobileNet + MinMax	112	0.01
EfficientNet	EfficientNet + BCE	EfficientNet + MinMax	97	0.01
ResNet	ResNet blocks + BCE	ResNet blocks + MinMax	**90**	**0.002**

4.2 Comparison of Loss Functions

Table 2 compares different GAN loss functions, all using the ResNet architecture for both the generator and discriminator.

Table 2. Comparison to GAN Loss Functions. We report the average FID (↓) scores and average KID (↓) scores on the CIFAR datasets

Model	Generator	Discriminator	FID↓	KID↓
ResNet (baseline)	ResNet blocks + BCE	ResNet blocks + MinMax	90	0.002
WGAN-GP with ResNet	ResNet blocks + BCE	ResNet blocks + Wasserstein Loss	35	0.001
lsGAN with ResNet	ResNet blocks + BCE	ResNet blocks + lsGAN	35	0.001
lsGAN with ResNet	ResNet blocks + lsGAN	ResNet blocks + lsGAN	29	0.001
Hinge Loss with ResNet	ResNet blocks + BCE	ResNet blocks + Hinge Loss	**25**	**0.001**

The baseline ResNet model with BCE and MinMax loss functions yields an FID score of 90. Using Wasserstein Loss (WGAN-GP) with ResNet significantly improves the FID score to 35. The least squares GAN (lsGAN) with ResNet achieves a similar FID score of 35 with BCE for the generator. When lsGAN is used for both generator and discriminator, the FID improves to 29. The best performance is achieved with Hinge Loss, bringing the FID score down to 25.

4.3 Comparison of Regularization Methods

Table 3 explores the impact of various regularization methods on the FID scores for the ResNet architecture with Hinge Loss.

Without any regularization, the ResNet model with Hinge Loss achieves an FID score of 25. Adding noise does not change the FID score. Contrastive Regularization (CR) improves the FID to 19, while the Gradient Penalty (GP-0) slightly worsens the FID to 26. Our proposed Regularized Loss Control (RLC) method yields the best FID score of 18.

Table 3. Comparison to GAN Regularization Methods. We report the average FID (↓) scores on the CIFAR datasets

Model	Regularization Methods	FID↓	KID↓
ResNet with Hinge Loss	No regularization	25	0.001
ResNet with Hinge Loss	+ Noise	25	0.001
ResNet with Hinge Loss	+ CR	19	0.001
ResNet with Hinge Loss	+ GP-0	26	0.001
ResNet with Hinge Loss (Ours)	+ RLC	**18**	**0.001**

4.4 Comparison of GAN Models

Table 4 provides a comparison of different GAN models, highlighting the effectiveness of our HingeRLC-GAN model.

Table 4. Comparison of GAN Models. We report the average FID (↓) scores and Inception Score (↑) on the CIFAR datasets

Model	FID↓	Inception Score↑
MGO-GAN	198	6.130
DROPOUT-GAN	66	-
DCGAN	53	6.47
LSGAN	56	6.32
DRAGAN	52	6.44
DFM	52	6.58
HingeRLC-GAN (Ours)	**18**	**6.89**

The MGO-GAN model shows the highest FID score of 198, indicating the poorest performance. DROPOUT-GAN significantly improves the FID score to 66. DCGAN and LSGAN achieve similar FID scores of 53 and 56, respectively. DRAGAN and DFM further improve the FID to 52. Our proposed HingeRLC-GAN achieves the best FID score of 18, demonstrating superior performance in generating high-quality and diverse images.

4.5 Mode Capture Analysis

We first present a t-SNE visualization of the CIFAR-10 dataset images, illustrating the clustering of different classes.

In the visualization:

- Airplanes are distinctly clustered in the top left corner, indicating clear separability from other classes.

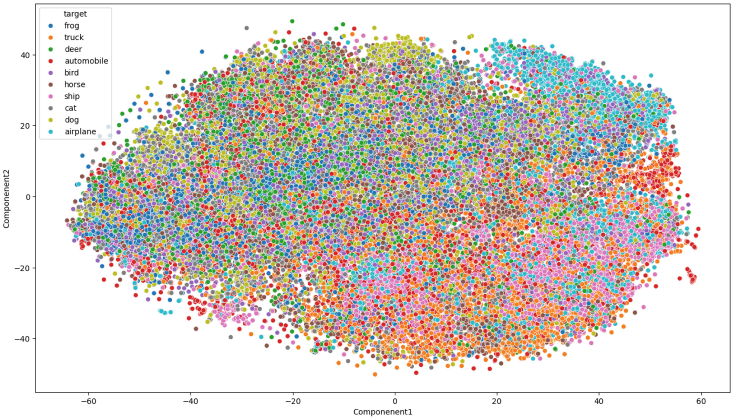


Fig. 4. t-SNE visualization of the CIFAR-10 dataset images.

- Frogs and cats show significant overlap with other categories and are dispersed across the visualization space.
- Automobiles are also spread out, suggesting intra-class variability.
- Trucks and ships, while more distinct from other classes, show a degree of overlap between them, located in the bottom left corner.

Next, we compare the t-SNE visualizations of images generated by DROPOUT-GAN and our HingeRLC-GAN, demonstrating a 30% improvement in mode capture with our model.

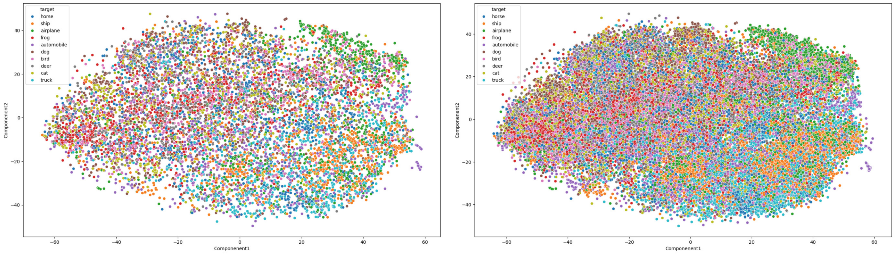


Fig. 5. t-SNE Visualizations: (left) DROPOUT-GAN, (right) HingeRLC-GAN. Mode coverage is 30% better then DROPOUT-GAN

4.6 Evaluation of HingeRLC-GAN

The HingeRLC-GAN evaluation yielded several significant results. Despite slower convergence, the model successfully produced high-quality and diverse images without mode collapse. The FID and KID metrics were recorded as 18 and 0.001, respectively. The training process showed gradual fluctuations in both discriminator and generator losses, indicating stable and balanced training dynamics. The KID metric showed a steady decline until about 60 epochs, after which it plateaued. The learning rate was reduced to 0.00005 after 80 epochs to maintain equilibrium between discriminator and generator losses as shown in Figure 6.

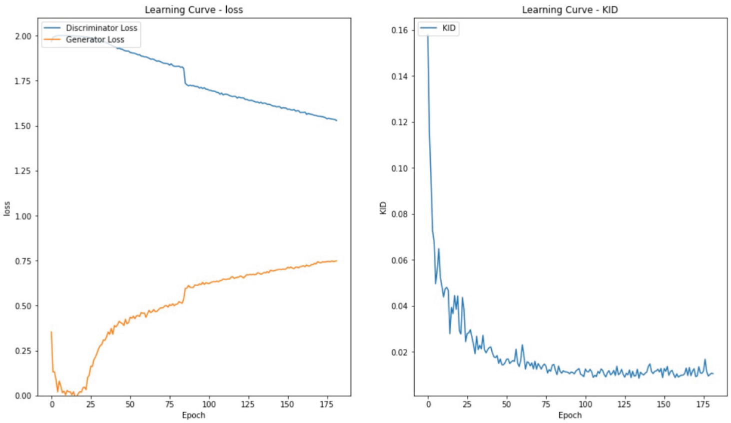


Fig. 6. Generator and Discriminator Loss, Learning Curve for HingeRLC-GAN

4.7 Generated Images

We present 10 images representing 10 different classes generated by the HingeRLC-GAN. Notably, vehicles (especially ships), birds, horses, deers, and dogs appeared realistic. Some anomalies were observed in frog and cat images, likely due to the intrinsic diversity within these classes in the CIFAR-10 dataset as shown in Figure 7. The anomalies in frog and cat images may be due to the intricate variations found within these specific classes in the CIFAR-10 dataset. Despite these anomalies, the HingeRLC-GAN's overall realism across various categories proves its suitability for a wide range of image generation tasks.

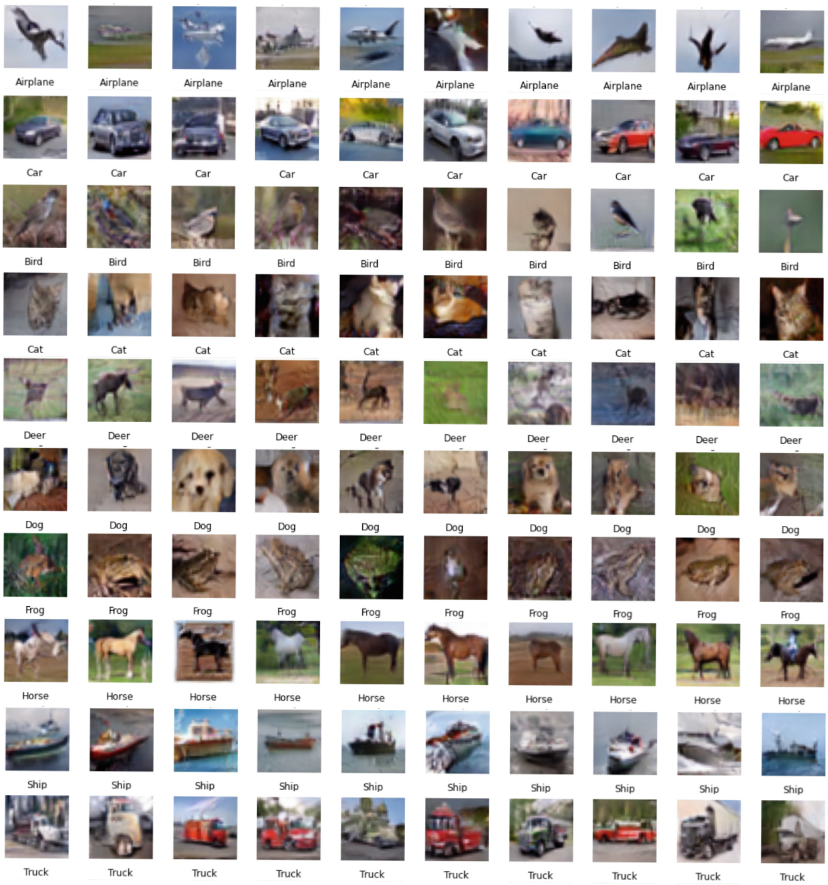


Fig. 7. Sample Images Generated by HingeRLC-GAN

5 Conclusion

In this paper, we introduced HingeRLC GAN, a novel variant that addresses mode collapse by integrating Hinge Loss with Regularized Loss Control (RLC). Our experiments demonstrate that this approach significantly enhances both the diversity and quality of generated images. Through extensive evaluation of various GAN architectures, we found ResNet to be the most effective baseline, and HingeRLC GAN consistently outperformed traditional loss functions like Wasserstein and Least Squares, as well as other regularization techniques, achieving the lowest FID scores. The HingeRLC GAN surpassed state-of-the-art models, including MGO-GAN, DROPOUT-GAN, DCGAN, LSGAN, DRAGAN, and DFM, proving its superiority in generating diverse and high-fidelity images. Overall, HingeRLC GAN provides a robust solution to mode collapse, enhancing training stability and output quality, with future research focusing on its application across diverse datasets and tasks in generative modeling and computer vision.

References

1. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. In: International conference on machine learning. pp. 214–223. PMLR (2017)
3. De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., Courville, A.C.: Modulating early visual processing by language. Advances in Neural Information Processing Systems **30** (2017)
4. Ghosh, A., Kulharia, V., Namboodiri, V.P., Torr, P.H., Dokania, P.K.: Multi-agent diverse generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8513–8521 (2018)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
6. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans (2017)
7. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems. pp. 6626–6637 (2017)
8. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: International conference on machine learning. pp. 1989–1998. PMLR (2018)
9. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
10. Li, W., Fan, L., Wang, Z., Ma, C., Cui, X.: Tackling mode collapse in multi-generator gans with orthogonal vectors. Pattern Recognition **110**, 107646 (2021)
11. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
12. Lin, F., Duan, Z., Deng, L.: Wgan-gp loss for stabilizing the training of multitask learning models for binary and multiclass image classification. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 437–446 (2021)
13. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
14. Mordido, G., Yang, H., Meinel, C.: Dropout-gan: Learning from a dynamic ensemble of discriminators. arXiv preprint arXiv:1807.11346 (2018)
15. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016). https://doi.org/10.23915/distill.00003, http://distill.pub/2016/deconv-checkerboard
16. Pan, Z., Niu, L., Zhang, L.: Unigan: Reducing mode collapse in gans using a uniform generator. Advances in Neural Information Processing Systems **35**, 37690–37703 (2022)
17. Pei, S., Da Xu, R.Y., Xiang, S., Meng, G.: Alleviating mode collapse in gan via diversity penalty module. arXiv preprint arXiv:2108.02353 (2021)
18. Robinson, S.K., Powell, R.D., Gupta, P., Pan, L., Sen, P.: Lapgan: Data augmentation for deep learning applications in melanoma detection. IEEE Access **7**, 84109–84116 (2019)

19. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 9243–9252 (2020)
20. Shen, Y., Yang, C., Tang, X., Zhou, B.: Interfacegan: Interpreting the disentangled face representation learned by gans. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(10), 3414–3426 (2020)
21. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of big data **6**(1), 1–48 (2019)
22. Tseng, H.Y., Jiang, L., Liu, C., Yang, M.H., Yang, W.: Regularizing generative adversarial networks under limited data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7921–7931 (2021)
23. 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)
24. Van Amersfoort, J., Smith, L., Teh, Y.W., Gal, Y.: Image extrapolation with graph neural networks and multi-scale adversarial training. arXiv preprint arXiv:2104.06184 (2021)
25. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)

LDFaceNet: Latent Diffusion-Based Network for High-Fidelity Deepfake Generation

Dwij Mehta(✉), Aditya Mehta, and Pratik Narang

Birla Institute of Technology and Science, Pilani, India
{f20190122,p20230303,pratik.narang}@pilani.bits-pilani.ac.in

Abstract. Over the past decade, there has been tremendous progress in the domain of synthetic media generation. This is mainly due to the powerful methods based on generative adversarial networks (GANs). Very recently, diffusion probabilistic models, which are inspired by non-equilibrium thermodynamics, have taken the spotlight. In the realm of image generation, diffusion models (DMs) have exhibited remarkable proficiency in producing both realistic and heterogeneous imagery through their stochastic sampling procedure. This paper proposes a novel facial swapping module, termed as LDFaceNet *(Latent Diffusion based Face Swapping Network)*, which is based on a guided latent diffusion model that utilizes facial segmentation and facial recognition modules for a conditioned denoising process. The model employs a unique loss function to offer directional guidance to the diffusion process. Notably, LDFaceNet can incorporate supplementary facial guidance for desired outcomes without any retraining. To the best of our knowledge, this represents the first application of the latent diffusion model in the face-swapping task without prior training. The results of this study demonstrate that the proposed method can generate extremely realistic and coherent images by leveraging the potential of the diffusion model for facial swapping, thereby yielding superior visual outcomes and greater diversity.

Keywords: Image Generation · Latent Diffusion Models · Facial Swapping · Guided Diffusion

1 Introduction

Recently, deep learning models of all types have started producing high-quality synthetic media. This media can be visual, audio or video files. Stunning image and audio samples have been created using GANs, autoregressive models, flows, and variational autoencoders (VAEs) [2,12,18,19,26,27]. Recent advances in fields like energy-based modeling and score matching have also started producing synthetic media that are comparable to those of GANs [33].

D. Mehta and A. Mehta—These authors contributed equally to this work.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 386–400, 2025.
https://doi.org/10.1007/978-3-031-78389-0_26

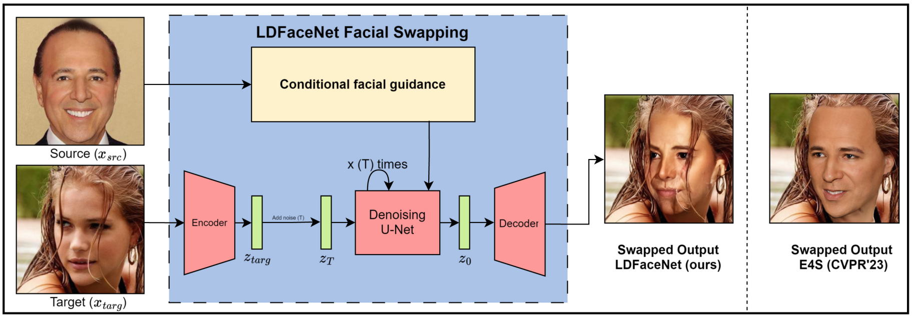

Source (x_{src})
Target (x_{targ})
LDFaceNet Facial Swapping
Conditional facial guidance
x (T) times
Encoder
Add noise (T)
Denoising U-Net
Decoder
z_{targ}
z_T
z_0
Swapped Output LDFaceNet (ours)
Swapped Output E4S (CVPR'23)

Fig. 1. Sample output of LDFaceNet. Compared to recent state-of-the-art methods such as E4S (CVPR'23 [22]), the results produced by LDFaceNet are significantly better. This particular example also illustrates that our method performs much better in handling occlusions over the target face than other generative methods. Further details are in the results Section 5.

GANs have emerged as the state-of-the-art method for image generation tasks. The performance for generative tasks is very subjective and varies from person to person. However, there are two commonly used distribution-based sample quality metrics such as FID [14] and Inception score (IS) [30] that most papers have used to report their results. Recently, GANs have been criticized for their limited diversity capture capabilities, and it has been demonstrated that likelihood-based models outperform GANs in this regard [27]. Additionally, GANs are often difficult to train, and we need to fine-tune their hyperparameters and regularizers to avoid collapse during training. Despite the drawbacks, GANs are still considered the leading method for image generation, but they are still unable to scale and apply to new domains. Consequently, there have been efforts to achieve state-of-the-art sample quality with likelihood-based models, which offer better scalability and ease of training. However, these models still lag behind GANs in terms of visual sample quality, and their sampling process is costlier and slower than that of GANs.

A class of likelihood-based models known as diffusion models [15,24] has recently been shown to produce visually realistic images while offering desirable characteristics such as variety, a stationary training objective, and simple scalability. These models generate samples by gradually eliminating noise from a signal, and their training objective can be described as a re-weighted variational lower bound. Compared to GANs, diffusion models enable more stable training and yield more desirable results in terms of fidelity and diversity. To manage the trade-off between fidelity and diversity, classifier guidance [10] is used to guide the diffusion process.

In the domain of image generation, face swapping is a computer vision task that involves transferring the face of one individual (the source) to another (the target) while preserving the target's facial attributes, such as identity, expression, and pose. This task has various applications in the entertainment industry, particularly in films, where it is used to replace the face of an actor with that of a stunt double or to resurrect deceased actors. Face swapping, also widely known as deepfakes generation can also be used for practical purposes, such as in forensic investigations and for facial reconstruction in the medical domain.

In this paper, we introduce a novel guided diffusion model, LDFaceNet, for deepfake generation. To the best of our knowledge, no prior research has explored face swapping using pre-trained latent diffusion models. Training diffusion models from scratch demands extensive computational resources and careful hyperparameter tuning. Our method, however, eliminates the need for re-training by leveraging the weights provided by Rombach et al. [28] from their LDM trained on the CelebA dataset [23]. We enhance this LDM with a unique facial guidance module. By using embeddings of images generated during intermediate timesteps, our model is constrained and guided through the facial guidance module. Additionally, we implement latent-level blending to ensure a seamless transition at the boundaries of the swapped face. This approach not only proves to be cost-effective but also outperforms existing facial swapping methods in both qualitative and quantitative evaluations by great margins. Furthermore, our method demonstrates robustness in handling faces with occlusions, misalignments, or non-frontal views, making it highly versatile in various challenging scenarios.

2 Related Work

2.1 Models for Image Synthesis

GANs Generative modeling faces unique challenges due to the large size of modern-day images. GANs [12] enable the effective synthesis of visually realistic images with good perceptual quality [2], but they are difficult to optimize and struggle to capture the complete data distribution. While likelihood-based methods prioritize accurate density estimation, their optimization behaves more reliably. Variational autoencoders (VAEs) and flow-based models can synthesize high-resolution pictures effectively, but their sample quality is generally inferior to that of GANs [36]. Autoregressive models (ARMs) [6,38], despite their good performance in density estimation, are limited by their sequential sampling procedure and computationally expensive designs [39], which restrict the resolution of the images they can produce. Maximum-likelihood training expends a disproportionate amount of capacity to model the scarcely perceptible, high-frequency details present at the pixel level, leading to lengthy training durations. To address this, several two-stage approaches first compress an image to a latent image space using ARMs rather than processing raw pixels, allowing for scaling to higher resolutions.

Diffusion Probabilistic Models Recently, Diffusion Probabilistic Models (DM) [15] have produced cutting-edge outcomes in sample quality. When their learned posterior or learned network's backbone is applied as a U-Net, these models are a natural fit for image-like data.

However, the disadvantage of evaluating and optimizing these models in pixel space is low inference speed because of repeated sequential sampling and very high training costs. While the former can be addressed in part by sophisticated sampling techniques like implicit diffusion models [32] and hierarchical approaches [37], training on high-resolution image data always necessitates the calculation of expensive gradients. Latent diffusion models (LDMs) were proposed to address the issue of expensive computations by performing the noising and denoising processes within a reduced latent space. DMs, when combined with classifier guidance [10] have proven effective in generating high-quality images tailored to specific object classes.

2.2 Face Swapping Models

Structural Guidance Based Models Traditional face-swapping methods, which require manual intervention, benefit greatly from structural information. For faces, landmarks, 3D representations, and segmentation, all of these provide strong structural priors which can be used to generate high-quality swapped images. Recent advancements in facial recognition, such as those by Deng et al. [8], have significantly enhanced the performance of traditional deep CNN architectures like ResNet-50 and MobileFace through fine-tuning with modified loss functions. These enhanced models serve as robust structural priors, further contributing to the generation of high-quality face-swapped images. However, these traditional methods [1,25] required manual intervention and could not correctly map the target expressions. 3D structural priors have recently been combined with GANs for an identity agnostic swapping module [17,21]. However, these methods are also limited by the accuracy of the underlying 3D models.

Reconstruction Based Models The original deepfakes model [7] is based on training two separate autoencoders with a shared encoder and different decoders. However, this approach requires retraining for each unique source-target pair. Conversely, GAN-based methods like SimSwap [5] have been developed to overcome the limitations of identity-specific face swapping, offering a more generalized approach that is not restricted to particular pairs of faces. The method proposed in SimSwap [5] involves segregating the identity data from the decoder component, thereby enabling the entire framework to be universally applicable to any given identity. However, this also generates low-quality results under certain conditions. Typically, subject-agnostic models adjust the intermediate features of the target image to incorporate the identity of the source image.

3 Preliminary: Diffusion Models

Diffusion Models (DMs) are generative models trained to reverse the earlier added noise using a parameterized Markovian process. Recent studies have demonstrated that DMs are capable of producing images of superior quality [10, 15]. In the following section, we present a concise summary of DMs.

Starting with any vector z_0, the forward noising process produces a series of latents $z_1, ..., z_T$ by adding Gaussian noise by following a variance schedule depicted by $\beta_t \in (0, 1)$ at time t:

$$q(z_t \mid z_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t} z_{t-1}, \beta_t \mathbf{I}) \tag{1}$$

After sufficient noise is added till timestep T, the last latent z_T is nearly an isotropic Gaussian distribution. The above equation's closed form can be derived using a simple reparametrization trick. Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha_t} = \prod_{i=1}^{t} \alpha_i$. Thus, we get the following:

$$\begin{aligned} q(z_t \mid z_0) \sim \mathcal{N}(\sqrt{\bar{\alpha_t}} z_0, (1-\bar{\alpha_t})\mathbf{I}) \\ z_t = \sqrt{\bar{\alpha_t}} z_0 + \sqrt{1-\bar{\alpha_t}} \epsilon \end{aligned} \tag{2}$$

Starting from the distribution $q(z_T)$, a reverse sequence can be generated by sampling the posteriors $q(z_{t-1}|z_t)$. These posteriors are also Gaussian distributions. To approximate this function, a deep neural network p_θ (a 2D U-Net architecture in the context of synthetic image generation) is trained to predict the mean and variance of z_{t-1} given z_t as input, or to estimate the noise $\epsilon_\theta(z_t, t)$, as proposed by Ho et al. [15]. However, vanilla diffusion models are computationally expensive because they operate directly on the pixel space ($z_t \in \mathbb{R}^{3\times H\times W}$). To address this, Rombach et al. [28] proposed Latent Diffusion Models (LDMs), which denoise in a compressed latent space and then upsample to pixel space using a pretrained VQGAN [11]. Our method uses LDMs along with conditional guidance [10] utilizing a novel facial guidance module.

4 Methodology

In this section, we describe our method, LDFaceNet. Given a source image x_{src}, a target image x_{targ}, and the facial segmentation mask of x_{targ} as $\mathcal{M}$, our goal is to transfer the facial features of the source image onto the target image while keeping all other attributes of the target image the same. More formally, we need to produce a modified $\widehat{x}$ such that $\widehat{x} \odot \mathcal{M}$ is as similar to x_{src} as possible. Furthermore, $\widehat{x} \odot (1 - \mathcal{M}) \approx x_{targ} \odot (1 - \mathcal{M})$ to preserve the background and to keep the complementary area nearly the same as before. Here $\odot$ is element-wise multiplication operator.

In Section 4.1, we extend the latent diffusion approach to support facial editing by incorporating a guiding cosine loss generated by the identity guidance module. Initially, our results indicated that while the swapped images maintained similarity to the source image and preserved the background, they did not map the emotional expressions of the target image.

Subsequently, in Section 4.2, we introduce a Euclidean L2 loss term generated by the segmentation guidance module to address this limitation. Our findings

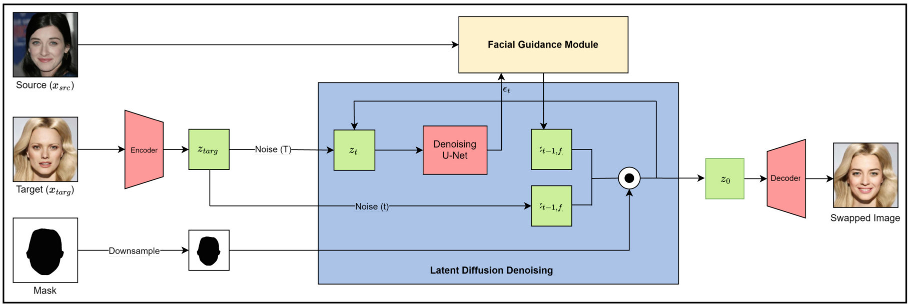


Fig. 2. Proposed sampling process. The sampling process begins by encoding the target image into a latent vector using an encoder. The encoder and decoder used in this method come from the same autoencoder based on previous work by Esser et al. [11]. Noise is added to this latent vector according to the diffusion noise schedule. Subsequently, a pre-trained U-Net is used to denoise this latent vector. The output of the U-Net is then conditioned using our novel facial guidance module. A downsampled facial mask ensures the masked area acquires the necessary facial characteristics through facial guidance while the background remains constant. Finally, after completing the denoising process, we pass the final latent vector z_0 into a decoder to get the swapped image. This entire process is detailed in Algorithm 1.

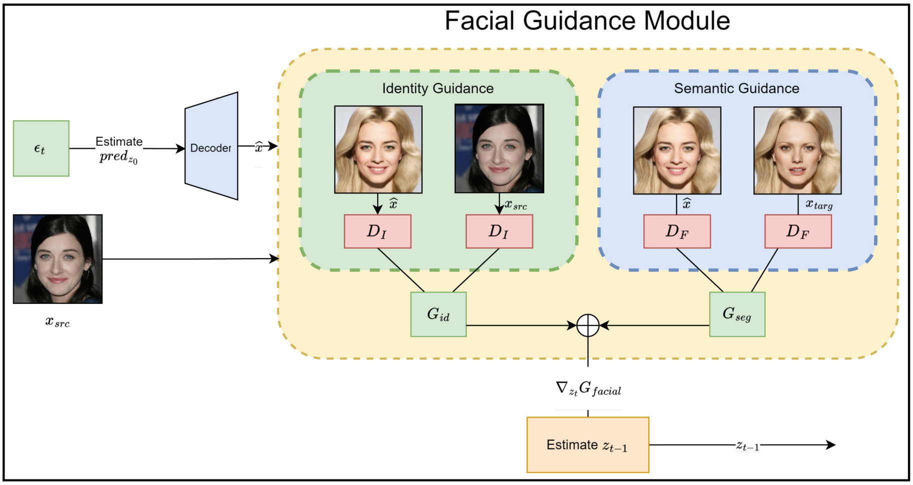


Fig. 3. Facial Guidance Module. The latent vector $pred_{z_0}$ is estimated from the output (ϵ_t) of the denoising U-Net. $pred_{z_0}$ is upsampled to get $\widehat{x}$, which approximates what our swapped image would look like after the entire denoising process. $\widehat{x}$ is then used within the identity and segmentation guided modules since these involve using pretrained classifiers trained on normal images and not latent vectors. The embeddings of $\widehat{x}$, x_{src}, and x_{targ} are then used as given in Algorithm 1 to calculate facial and segmentation guidance loss modules. These are combined to form the complete facial guidance loss term. The gradient of this facial loss term with respect to z_t is used to guide the reverse diffusion process.

demonstrate that our method produces coherent and realistic results. Specifically, the generated images exhibit remarkable similarity to the source image in terms of skin color, eye color, shape, structure, and lighting. Furthermore, they effectively preserve the original facial attributes and emotional expressions of the target image.

To further validate our approach, we conduct a comprehensive ablation study to evaluate the efficacy of the proposed solutions.

4.1 Source Identity Guided Diffusion

We propose applying facial guidance during the denoising process in order to dictate the facial attributes of generated images. One significant benefit of using this method is that, after training, we can control the image produced by the sampling process's guidance. As a result, we can produce the necessary images without having to retrain the LDM. We use external facial recognition modules to provide guidance in order to take advantage of this advantage. We use the embeddings of these facial recognition modules to calculate the guidance loss term.

The preferred approach for face recognition uses Deep Convolutional Neural Network (DCNN) embedding to represent faces [4,8,31,34,35]. For our experiments, we use a ResNet-50 backbone [13] pre-trained on the MS1MV3 dataset using the ArcFace [8] loss. The identity guidance module, denoted as D_I, constrains the ID vector of x_{src} to be closer to $\widehat{x}$, the approximation of the swapped image, which is estimated at each denoising step.

In latent diffusion, the actual denoising occurs in the latent space Z. Since we are utilizing pre-trained models trained on actual images, it is necessary to first calculate $\widehat{z}$, an estimation of z_0 given z_t. Subsequently, we upsample $\widehat{z}$ to obtain an approximation of $\widehat{x}$. This $\widehat{x}$ is then passed as an input into D_I to extract feature embeddings. These embeddings, along with the embeddings of the source image, are processed using a cosine loss.

The overall process is formally described in Algorithm 1 from lines 8 to 10, and the guidance loss is defined as follows:

$$G_{id} = 1 - \cos(D_I(x_{src}), D_I(\widehat{x})) \tag{3}$$

4.2 Target Segmentation Guided Diffusion

Using only the Identity Guidance Loss (G_{id}) fails to preserve the facial expressions of the target image, such as shape, eye structure, lip structure, and overall facial structure. Consequently, the expressions of x_{src} are mapped onto the generated image, which, even though it gives a satisfactory transfer of identity, but with a loss of the target's expressions.

To address this issue, we use BiseNet [41] as a face segmentation model (D_F), which predicts pixel-wise probabilities for facial components (such as the nose, eyebrows, and eyes). This allows us to explicitly match the facial expressions of

Algorithm 1 LDFaceNet sampling, given a latent diffusion model $\epsilon_\theta(z_t, t)$, Encoder ε, Decoder $\mathcal{D}$, ArcFace Identifer D_I and BiseNet Parser D_F

```
**Input**: Source image $x_{src}$, Target image $x_{targ}$, Target mask $\mathcal{M}$, diffusion steps $k$
**Output**: Face swapped image $\widehat{x_0}$
$z_0 = \varepsilon(x_{targ})$
$z_k \leftarrow$ sample from $\mathcal{N}(\sqrt{\bar{\alpha_k}}z_0, \sqrt{1-\bar{\alpha_k}}\mathbf{I})$
$m \leftarrow$ downsampled from $\mathcal{M}$
**for** $t$ from $k$ to 1 **do**
    $\epsilon_t = \epsilon_\theta(z_t, t)$
    $\widehat{z} = \frac{1}{\sqrt{\bar{\alpha_t}}}\left(z_t - \sqrt{1-\bar{\alpha_t}}\epsilon_t\right)$
    $\widehat{x} = \mathcal{D}(\widehat{z})$
    $G_{id} = 1 - \cos(D_I(x_{src}), D_I(\widehat{x}))$
    $G_{seg} = \|D_F(x_{targ}) - D_F(\widehat{x})\|_2^2$
    $G_{fac} = \lambda_{id}(t)G_{id} + \lambda_{seg}(t)G_{seg}$
    $\epsilon_t = \epsilon_t + \sqrt{1-\bar{\alpha_t}}\nabla_{z_t}G_{fac}$          ▷ guide $\epsilon_t$ using the gradient of $G_{fac}$
    $\widehat{z} = \frac{1}{\sqrt{\bar{\alpha_t}}}\left(z_t - \sqrt{1-\bar{\alpha_t}}\epsilon_t\right)$
    $z_{t-1,fg} \leftarrow$ sample from $\mathcal{N}\left(\sqrt{\alpha_{\bar{t-1}}}\widehat{z} + \sqrt{(1-\alpha_{\bar{t-1}}-\Sigma^2)}\epsilon_t, \Sigma\right)$
    $z_{t-1,bg} \leftarrow$ sample from $\mathcal{N}(\sqrt{\bar{\alpha_k}}z_0, \sqrt{1-\bar{\alpha_k}}\mathbf{I})$
    $z_{t-1} = z_{t-1,fg} \odot m + z_{t-1,bg} \odot (1-m)$
**end for**
$\widehat{x_0} = \mathcal{D}(z_0)$
**return** $\widehat{x_0}$
```

the synthesized image to those of the target. Through segmentation guidance, the generated image retains similarity to the target image in terms of expression, pose, and shape. The Euclidean L2 distance between the two segmentation maps is then calculated. The formal segmentation guidance loss term is defined as follows:

$$G_{seg} = ||D_F(x_{targ}) - D_F(\widehat{x})||_2^2 \tag{4}$$

The final facial guidance loss term (G_{fac}) is calculated by combining the identity guidance loss (G_{id}) and the segmentation loss (G_{seg}), each weighted by their respective lambdas. These lambdas are not constants. We discovered that using a decreasing step function for these lambdas as denoising progresses performs better than keeping them constant. In our experiment, we decrease the lambdas according to a stepwise decreasing schedule. G_{fac} is formally defined as follows:

$$G_{fac} = \lambda_{id}(t)G_{id} + \lambda_{seg}(t)G_{seg} \tag{5}$$

4.3 Background Preservation

Burt and Adelson [3] demonstrated that images can be effectively blended by combining each level of their Laplacian pyramids separately. Building on this method, we blend at different timesteps at the latent level to introduce varying amounts of noise as the denoising process progresses. The rationale is that, during each step of this sampling process, we superimpose noisy latents in the form of the background onto a set of naturally noisy images. Directly merging two noisy images from the same timestep often results in incoherence due to differing distributions. However, the subsequent diffusion step projects the result onto the manifold of the next level, enhancing coherence.

Formally, starting with a noisy latent z_t, we execute a guided diffusion step that produces a latent $z_{t-1,fg}$. Concurrently, we obtain $z_{t-1,bg}$ using Equation (2). These two latents are then blended using a mask m, which is downsampled from the original target mask $\mathcal{M}$:

$$z_{t-1} = z_{t-1,fg} \odot m + z_{t-1,bg} \odot (1 - m) \tag{6}$$

5 Results and Discussion

In this section, we provide a comprehensive analysis of the LDFaceNet model. We assess the proposed approach from both a quantitative and qualitative perspective to ascertain its robustness. In addition, we perform a few ablation experiments to evaluate the relative contributions of different components of the model, which highlight the importance of their presence.

5.1 Quantitative and Qualitative results

To generate the results, we use LDFaceNet with the pre-trained LDM model, ArcFace identity extractor [8], and BiseNet face parser [41]. The generated images are obtained through the sampling process detailed in Algorithm 1. For quantitative analysis we use three metrics. The ability to transfer structural attributes is indicated by the pose error and expression error. These errors are represented as L2 distances between the pose and expression feature vectors of the swapped image and target image. Pose and expression vectors are generated using pre-trained estimators, specifically Hopenet [29] and a 3D face reconstruction model [9], respectively. We also calculate the ID similarity score, which is the cosine similarity between swapped faces and their corresponding sources.

We present the results side-by-side for each pair of source and target images in figure 4. It clearly demonstrates the ability of LDFaceNet to generate realistic images by transferring the facial features and expressions of the target image onto the source image. The generated images are compared with other state-of-the-art methods for a thorough analysis. Further figure 4 analyses the quantitative performance by showing the cosine similarity (higher the better), pose error (lower the better) and expression error (lower the better). The numbers also unequivocally demonstrate our model's superior performance compared to other recent face-swapping models. It is evident that LDFaceNet outperforms the previous techniques, including recent models like E4S (CVPR'23) by a considerable margin.

Overall, the results demonstrate that LDFaceNet can produce high-quality images that closely resemble the source image while retaining the characteristics

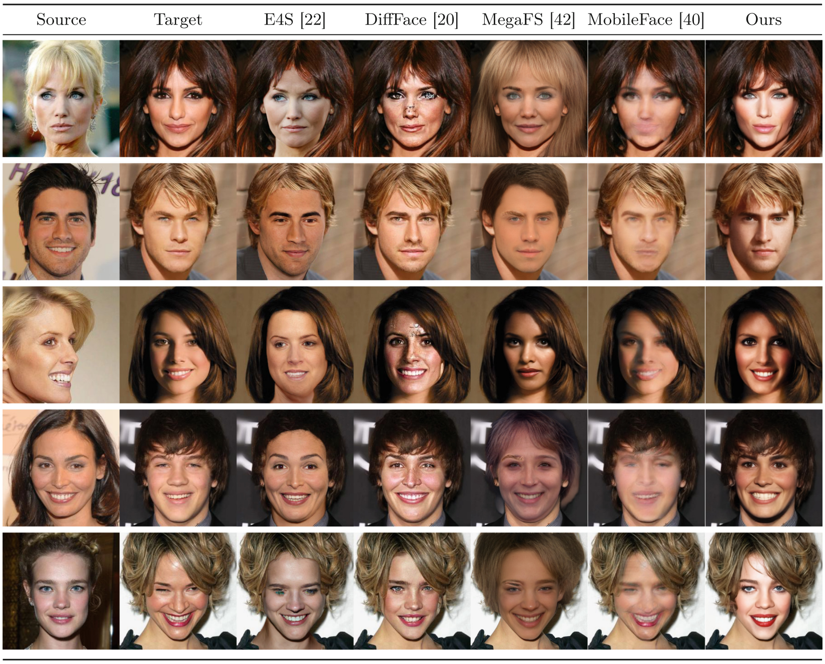


Fig. 4. Qualitative Results. Our method achieves high-fidelity results, better preserving source identity and target facial attributes than other methods. It also handles occlusions and partial views robustly.

of the target image. The generated images show realistic facial expressions, lighting, and background, which are crucial for creating realistic face swaps. These results highlight the potential of LDFaceNet as a powerful tool for image manipulation and face swapping (Table 1).

5.2 Ablation Study

To assess the significance of the identity and segmentation guidance modules, we conducted experiments with three different configurations: disabling only the segmentation module, disabling both modules, and enabling both modules. The results of these experiments are shown in Figure 6, highlighting the importance of both modules. Specifically, when the segmentation module is disabled, the source's facial expression is copied onto the result image, and the target's facial expression is lost. This demonstrates the essential role of the segmentation module in preventing the loss of target's facial expression.When both modules are disabled, our model attempts to reconstruct the source image without any guidance. As a result, the generated image appears visually similar to the target image, with no discernible change in facial features or attributes. This highlights the crucial role played by the identity and segmentation guidance modules in

Table 1. Comparative results of the LDFaceNet and other existing face swapping methods over CelebA dataset [23]Comparative results of the LDFaceNet and other existing face swapping methods over CelebA dataset [23].

Method	ID similarity ↑	Pose ↓	Expr. ↓
MobileFace (AAAI'22) [40]	0.25	2.52	3.72
MegaFS (CVPR'21) [42]	0.26	2.48	3.27
DiffFace [20]	0.55	2.40	2.71
E4S (CVPR'23) [22]	0.61	2.31	2.80
LDFaceNet	**0.67**	**2.18**	**2.55**

achieving facial swapping with controllable and desirable results. We present these obeservations and quantitative scores through Figure 5 for a better visualization of contribution of the two components.

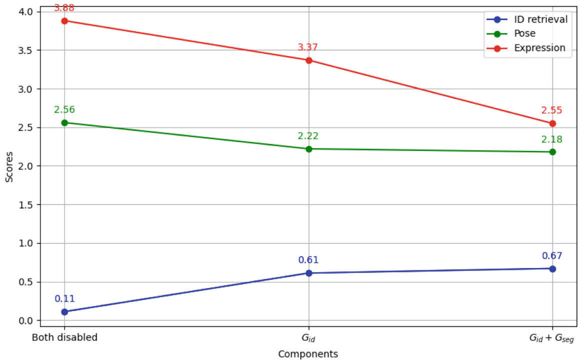


Fig. 5. Comparative performance of ablation experiments. x-axis represents the three variants of LDFaceNet. The three lines describe the performance of each variant for three metrics.

Our ablation experiments indicate that the identity and segmentation guidance modules are critical for achieving high-quality facial swapping with LDFaceNet. By incorporating facial guidance, we achieve better results in visual fidelity and attribute preservation. Additionally, our model can apply different levels of guidance to balance identity and attribute preservation. While LDFaceNet performs excellently, there are opportunities for further enhancement. One direction is to train a new diffusion model on CelebA using classifier-free guidance [16]. Incorporating more identity and face-parser networks into an ensemble could also create a more robust guidance loss function, further refining our model's capability for face swaps.

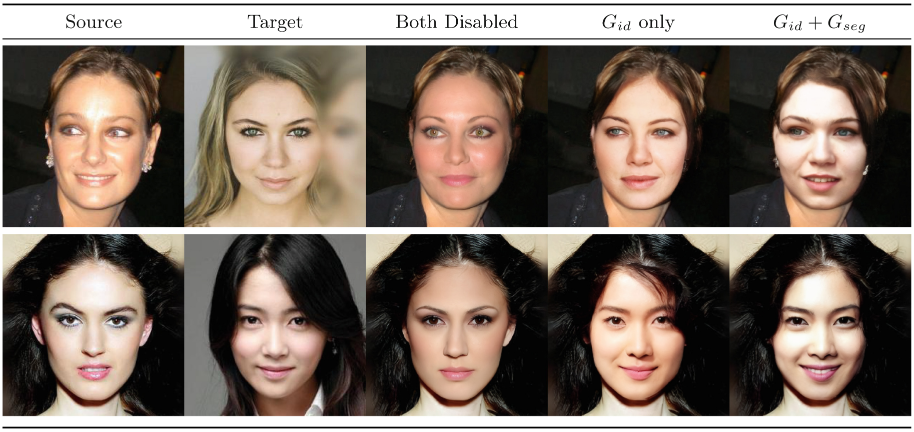


Fig. 6. Ablation study generated examples. Qualitative results of the ablation study demonstrate the importance of the individual guidance modules within our comprehensive facial guidance module. Disabling both modules causes the model to behave like a simple LDM, attempting to reconstruct the target image as it is. When only the identity guidance is enabled, the identity of the target is mapped, but the source's facial expressions are lost. Further, when both modules are enabled, the model successfully preserves both the source's identity and the target's facial expressions.

6 Conclusion

LDFaceNet is a guided diffusion model for facial swapping that leverages facial segmentation and facial recognition modules for a conditioned denoising process. With its unique guidance loss functions, LDFaceNet offers directional guidance to the diffusion process, and can incorporate supplementary facial guidance for desired outcomes without retraining. LDFaceNet improves upon previous GAN-based approaches by utilizing the potential of the diffusion model for facial swapping, resulting in superior visual outcomes and greater diversity.

In conclusion, LDFaceNet offers a promising new approach to facial swapping by utilizing guided diffusion, segmentation, and recognition modules. The results demonstrate the proposed method's efficacy and highlight the diffusion model's potential for face swapping tasks. This study represents a significant contribution to the field of face swapping using diffusion models and serves as a foundation for future research in this area.

References

1. Blanz, V., Scherbaum, K., Vetter, T., Seidel, H.P.: Exchanging faces in images. In: Computer Graphics Forum. vol. 23, pp. 669–676. Wiley Online Library (2004)
2. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)

3. Burt, P.J., Adelson, E.H.: The laplacian pyramid as a compact image code. In: Readings in computer vision, pp. 671–679. Elsevier (1987)
4. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). pp. 67–74. IEEE (2018)
5. Chen, R., Chen, X., Ni, B., Ge, Y.: Simswap: An efficient framework for high fidelity face swapping. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2003–2011 (2020)
6. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509 (2019)
7. Deepfakes: Deepfakes/faceswap: Deepfakes software for all (2021), https://github.com/deepfakes/faceswap
8. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019)
9. 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: IEEE Computer Vision and Pattern Recognition Workshops (2019)
10. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Adv. Neural. Inf. Process. Syst. **34**, 8780–8794 (2021)
11. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
12. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2014)
13. 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)
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
16. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
17. Jiang, L., Li, R., Wu, W., Qian, C., Loy, C.C.: Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2889–2898 (2020)
18. Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., van den Oord, A., Dieleman, S., Kavukcuoglu, K.: Efficient neural audio synthesis. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 2410–2419. PMLR (10–15 Jul 2018), https://proceedings.mlr.press/v80/kalchbrenner18a.html
19. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation (2018)
20. Kim, K., Kim, Y., Cho, S., Seo, J., Nam, J., Lee, K., Kim, S., Lee, K.: Diffface: Diffusion-based face swapping with facial guidance. arXiv preprint arXiv:2212.13344 (2022)

21. Li, Y., Ma, C., Yan, Y., Zhu, W., Yang, X.: 3d-aware face swapping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12705–12714 (June 2023)
22. Liu, Z., Li, M., Zhang, Y., Wang, C., Zhang, Q., Wang, J., Nie, Y.: Fine-grained face swapping via regional gan inversion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8578–8587 (2023)
23. Liu, Z., Luo, P., Wang, X., Tang, X.: Large-scale celebfaces attributes (celeba) dataset. Retrieved August **15**(2018), 11 (2018)
24. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. pp. 8162–8171. PMLR (2021)
25. Nirkin, Y., Masi, I., Tuan, A.T., Hassner, T., Medioni, G.: On face segmentation, face swapping, and face perception. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). pp. 98–105. IEEE (2018)
26. Prenger, R., Valle, R., Catanzaro, B.: Waveglow: A flow-based generative network for speech synthesis. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3617–3621 (2019). https://doi.org/10.1109/ICASSP.2019.8683143
27. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf
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. Ruiz, N., Chong, E., Rehg, J.M.: Fine-grained head pose estimation without keypoints. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 2074–2083 (2018)
30. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Advances in neural information processing systems **29** (2016)
31. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
32. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
33. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf
34. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. Advances in neural information processing systems **27** (2014)
35. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1701–1708 (2014)
36. Vahdat, A., Kautz, J.: Nvae: A deep hierarchical variational autoencoder. Adv. Neural. Inf. Process. Syst. **33**, 19667–19679 (2020)
37. Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. Adv. Neural. Inf. Process. Syst. **34**, 11287–11302 (2021)

38. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International conference on machine learning. pp. 1747–1756. PMLR (2016)
39. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
40. Xu, Z., Hong, Z., Ding, C., Zhu, Z., Han, J., Liu, J., Ding, E.: Mobilefaceswap: A lightweight framework for video face swapping. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 2973–2981 (2022)
41. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 325–341 (2018)
42. Zhu, Y., Li, Q., Wang, J., Xu, C.Z., Sun, Z.: One shot face swapping on megapixels. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4834–4844 (2021)

Adaptive Graph Convolutional Fusion Network for Skeleton-Based Abnormal Gait Recognition

Liang Wang and Jianning Wu(✉)

College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China

qsx20221304@student.fjnu.edu.cn, jianningwu@fjnu.edu.cn

Abstract. Dynamic interactions between human joints and bones convey significant information for skeleton-based abnormal gait recognition. Existing graph convolutional networks (GCNs)-based methods either only consider the locomotion information of the joints or treat the motion information of joints and bones independently, failing to explore the implicit dynamic interactions between joints and bones effectively. These interactions also contain rich and useful abnormal gait representation information. In this work, we propose a novel adaptive graph convolutional fusion network (AGCFN) for skeleton-based abnormal gait recognition. The joint motion information and bone motion information are modelled as a joint spatiotemporal gait graph and a bone spatiotemporal gait graph, respectively. Our AGCFN is designed to explore the interaction information between joints and bones through learning the inter-graph relationships between the above two gait graphs, so as to obtain more discriminative gait feature representations. Extensive experiments on our abnormal gait dataset demonstrate that the generalization performance of our model exceeds the state-of-the-art by a significant margin.

Keywords: Graph fusion network · Graph convolutional network · Adaptive learning · Abnormal gait

1 Introduction

Abnormal gait is related to impaired interactions between joints, bones and muscles under the control of the central nervous system [1]. It may seriously affect human walking function in daily life. The accurate identification of abnormal gait has become an active research area in recent years, as it contributes to the diagnosis and treatment of diseases related to gait abnormality.

In the past decade, skeleton-based abnormal gait recognition methods have been widely investigated and have attracted increasing attention, since skeleton data contains richer locomotion information about human anatomical structure [2–11]. Certain approaches model human skeleton data as a sequence of

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78389-0_27.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 401–415, 2025.
https://doi.org/10.1007/978-3-031-78389-0_27

coordinate vectors or 2D grid, then feed it into CNNs or LSTMs to capture spatial-temporal gait features for prediction [5–11]. However, such deep learning methods ignore spatial structural information of the human skeleton. Actually, the human skeleton can be naturally structured as a graph topology in non-Euclidean space, where graph vertices denote all the joints of the human body and edges represent the physical connection between them. The previous methods cannot handle such graph structure data and exactly learn irregular gait characteristics in non-Euclidean space.

Recently, graph convolutional networks (GCNs) [12] have achieved considerable success in various graph learning tasks [13–21]. For the abnormal gait recognition task, inspired by the study of spatial-temporal graph convolutional network (ST-GCN) [22], researchers have explored the feasibility of using GCN-based graph models to extract spatial and temporal dynamic features of abnormal gait [23–25]. In these studies, human joint locomotion information is abstracted into a gait graph, and spatial-temporal graph deep learning models are developed to discover the most discriminative gait abnormality representation associated with the interactions between human joints. However, the above methods only focus on the locomotion information of human joints and ignore bone motion information, which has been proven to be important and informative [26]. Currently, certain works [27] have attempted to construct multiple abnormal gait graphs to represent joint and bone locomotion information within human skeleton data. For example, Guo et al. [28] proposed a two-stream spatial-temporal attention graph convolutional network (2s-ST-AGCN) to discover the best spatial-temporal gait dynamic representation from two-stream gait-graph including joint gait-graph and bone gait-graph.

Although significant progress has been made in current research on abnormal gait recognition using two-stream networks, some shortcomings remain. In these two-stream networks, the joint stream and the bone stream are treated completely independently, ignoring the interactions between joints and bones. That is, these methods only consider the intra-graph relationships within the joint gait graph and bone gait graph separately, i.e. the interactions between joints and the interactions between bones. They do not account for the inter-graph relationships between the joint gait graph and the bone gait graph. This inter-graph relationships also contain rich and discriminative gait abnormality characteristics associated with dynamic interactions between the joints and bones. Considering the implicit interactions between the joints and bones may be beneficial for the pattern recognition of abnormal gait.

To the best of our knowledge, no study has attempted to explore the inter-graph relationships between the joint gait graph and the bone gait graph. To this end, this study presents an adaptive graph convolutional fusion network (AGCFN) for skeleton-based abnormal gait recognition. It parameterizes a cross-adaptive adjacency matrix, which is trained and updated simultaneously with convolutional parameters of the model, to fuse the joint gait graph and the bone gait graph into a joint-bone fusion gait graph. With this fusion gait graph, our model could discover more discriminative gait abnormality features associated with the interactions between joints and bones hidden in skeleton data. We employed Kinect sensor data of six mimic abnormal gaits to evaluate the feasibility of our proposed model. The experiment results show that our model achieves state-of-the-art performance on our dataset. The main contributions of our work are three-fold:

1. A novel graph fusion network is developed to capture the most discriminative spatiotemporal dynamic gait abnormality features for high-generalization.
2. With the cross-adaptive adjacency matrix, our model can learn the optimal gait abnormality representation associated with the interactions between joints and bones hidden in skeleton data.
3. The proposed model reaches the best performance with the lowest computation cost compared to recent state-of-the-art models.

2 Related work

2.1 Skeleton-based abnormal gait recognition

Recently, numerous studies have focused on investigating the feasibility of applying advanced deep learning models to skeleton-based abnormal gait recognition in an end-to-end manner. Some researchers initially utilized traditional deep learning models, such as CNNs and LSTMs, to discover spatial and temporal gait abnormality features by modelling human skeleton data. LSTM-based methods [5–8] usually model the skeleton data as a sequence of coordinate vectors representing human body joints. CNN-based methods [9,10] usually treat the skeleton data as a pseudo-image. Though these works have made considerable progress, they cannot explore the irregular spatial-temporal gait abnormality features because the traditional deep learning algorithms mainly depend on shift-invariant and local correlation for spatial or temporal feature extraction in Euclidean space.

To overcome the above shortages, some studies have attempted to model human skeleton data based on graph deep learning models such as GCNs [22–25]. They aim to take advantage of the excellent graph learning capability to capture the implicit irregular interaction features between joints and bones in non-Euclidean space. Additionally, in order to fully explore the joint and bone movement information in skeleton data, the multi-stream graph learning models were investigated to improve the generalization of abnormal gait recognition [27, 28]. However, how to learn the implicit motion interactions between the joints and bones has not been studied in sufficient depth.

2.2 Graph fusion network

Graph fusion learning networks have been proposed for discovering the most representative features from multiple graph topological structures [29–32]. Its advantage is to integrate multiple graph topological structures with different types of data into a unified graph topological structure with richer representation information. Many relevant studies have demonstrated the feasibility of graph fusion learning networks for capturing the most representative features with richer information based on multimodal data. For instance, Hu et al. [33] proposed a novel graph fusion neural network for multi-modal freezing of gait detection. Dhawan et al. [34] designed a graph attention fusion network for multimodal fake news detection. In addition, Tu et al. [35] proposed a joint-bone fusion graph convolutional network to discover the motion transmission between joints and bones. These remarked achievements made in current studies motivate us to investigate the feasibility of developing the novel gait-graph fusion learning network for capturing the richer coupling action representation information from skeleton data.

3 Methods

3.1 Pipeline overview

In the present study, a novel adaptive graph convolutional fusion network (AGCFN) is developed to accurately classify skeleton-based abnormal gait patterns, as shown in Fig. 1. The input to the model consists of a joint spatial-temporal gait graph and bone spatial-temporal gait graph. Then, three abnormal gait feature extraction blocks are designed. Each block consists of an adaptive graph convolutional fusion layer, a temporal convolutional layer, a residual connection, two batch normalization layers, and two ReLU layers. A global average pooling layer and a fully connection layer are then employed to map the gait representation to the most distinctive feature space. The final output is sent to a SoftMax classifier to obtain the prediction. The cross-entropy loss is used to train the entire network end to end. We will now go over the components of the AGCFN model.

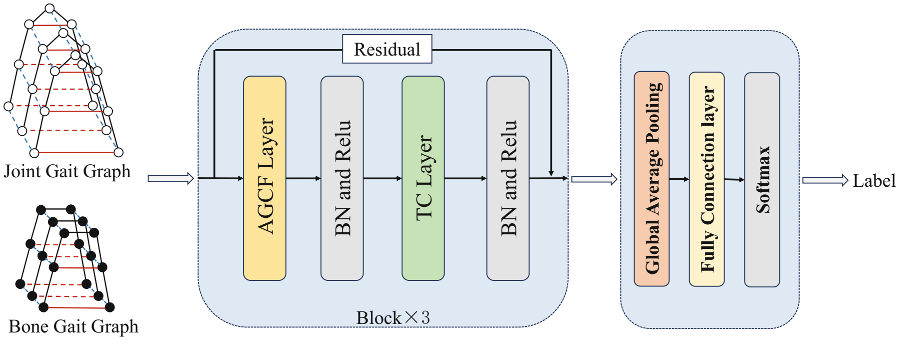


Fig. 1. Illustration of our proposed model

3.2 Spatiotemporal gait graph construction

In this work, we employ spatial-temporal graph topology structure to model skeleton sequence of the lower limbs, as shown in Fig. 2. The hip, knee and ankle joints of right and left lower limbs, and pelvis joint were considered. A joint spatiotemporal gait graph $G^j_{st} = \{V^j, E^j\}$ is first defined, as shown in Fig. 2(a). In this graph, $V^j = \{v_{t,i} | t = 1, ..., T; i = 1, ..., N^j\}$ represents the sequence of graph vertices, where N^j is a total of lower limb joints selected and T denotes the frame length of one gait cycle. The edge set $E^j = \{E^j_1, E^j_2, E^j_3\}$. The first type of edge represents natural bone connections between two joints at each frame, denoted as E^j_1. The second type of edge set represents gait symmetry connections between symmetrical joints of the two lower limbs, denoted as E^j_2, The last type of edge set connects the same joints in consecutive frames, denoted as E^j_3. The connectivity between nodes v_i and v_j can be defined by adjacency matrix $A^j \in R^{N^j \times N^j}$ at each frame. The coordinate information of all joints in spatiotemporal gait graph can be defined as $X^j \in R^{T \times N^j \times C^j}$, where C^j is the dimension of joint coordinate vector. Therefore, the joint spatiotemporal gait graph is defined as $G^j_{st} = \{X^j_t, A^j\}^T_{t=1}$. After that, we defined a bone spatiotemporal gait graph $G^b_{st} = \{V^b, E^b\}$, as shown in Fig. 2(b). In this graph, $V^b = \{v_{t,i} | t = 1, ..., T; i = 1, ..., N^b\}$ represents the sequence of graph vertices, where N^b is a total of bones between joints in lower limbs, which is equal to 6. Here, each bone can be denoted as a vector pointing to its target joint to its source joint. For example, given a bone with its target joint $v_t = (x_1, y_1, z_1)$ and its source joint $v_s = (x_2, y_2, z_2)$, the bone vector is defined as $e = (x_1 - x_2, y_1 - y_2, z_1 - z_2)$. Like the joint gait graph, there are three same types of edge sets in bone graph. The connectivity of bone gait graph can be denoted by adjacency matrix $A^b \in R^{N^b \times N^b}$. The coordinate information of all bones in spatiotemporal gait graph can be defined as $X^b \in R^{T \times N^b \times C^b}$, where C^b is the same as the dimension of joint coordinate vector. Thus, the bone spatiotemporal gait graph could be defined as $G^b_{st} = \{X^b_t, A^b\}^T_{t=1}$.

3.3 Adaptive graph convolutional fusion layer

In our model, the adaptive graph convolutional fusion layer is adopted for extracting spatial gait abnormality features by fusion learning of joint spatial-temporal gait graph and bone spatial-temporal gait graph, as shown in Fig. 3. Given that the inputs include joint spatiotemporal gait graph $G^j_{st} = \{X^j_t, A^j\}^T_{t=1}$ and bone spatiotemporal gait graph $G^b_{st} = \{X^b_t, A^b\}^T_{t=1}$. To construct the joint-bone fusion gait graph, the cross-adaptive adjacency matrix $C = \{C^j, C^b\}$ is defined to learn the connections between joint gait graph and bone gait graph. C^j represents the connection from joint gait graph to bone gait graph and C^b denotes the connection from bone gait graph to joint gait graph. They represent the cross connection between joints and bones of lower limbs. Note that the elements in matrix C can be arbitrary values. That is, they indicate not only the connectivity between two vertices but also the strength of the connectivity.

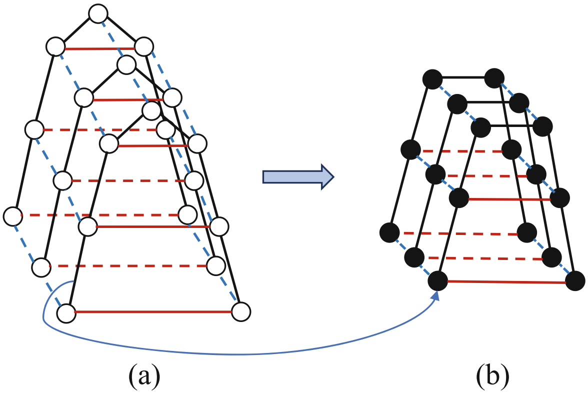


Fig. 2. Illustration of spatiotemporal gait graph. (a). Joint spatiotemporal gait graph; (b). Bone spatiotemporal gait graph

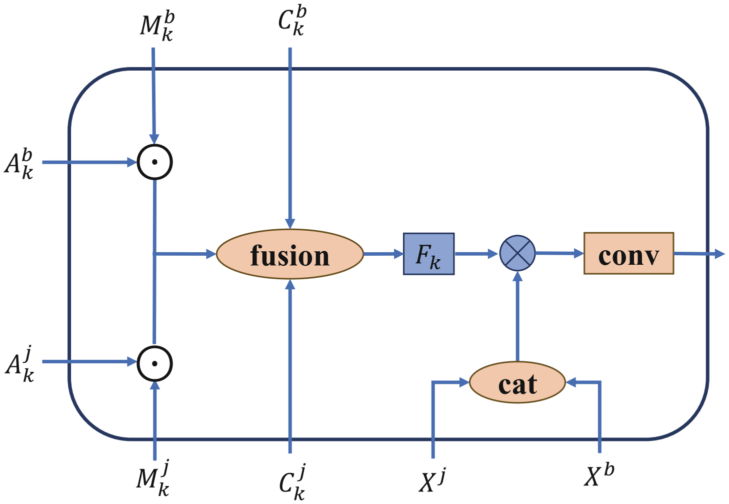


Fig. 3. Illustration of the adaptive graph convolutional fusion layer

Besides, we add a learnable mask $M = \{M^j, M^b\}$ to scale the different contribution of a node's feature to its neighboring nodes for joint gait graph and bone gait graph. Here, we can obtain the fusion adjacency matrix F, which is defined as follows:

$$F = \{A^j \odot M^j, A^b \odot M^b, C^j, C^b\} \tag{1}$$

where $\odot$ denotes the dot product, and $F \in R^{(N^j+N^b)\times(N^j+N^b)}$. The construction process of fusion adjacency matrix is illustrated in Fig. 4. As shown in Fig. 4, the fusion adjacency matrix consists of four parts: the result of the dot product of A^j and M^j, the result of the dot product of A^b and M^b, C^j, and C^b. Based on this adaptive fusion adjacency matrix F, we can determine the topology structure of the joint-bone fusion gait graph. Thus, the fusion spatiotemporal gait graph can be defined as $G_{st} = \{X_t, F\}_{t=1}^{T}$, where X can be defined as follows:

$$X = X^j \|_H X^b \tag{2}$$

where $\|$ denotes the concatenation operation on the node quantity dimension H, and $X \in R^{H \times T \times (N^j + N^b)}$.

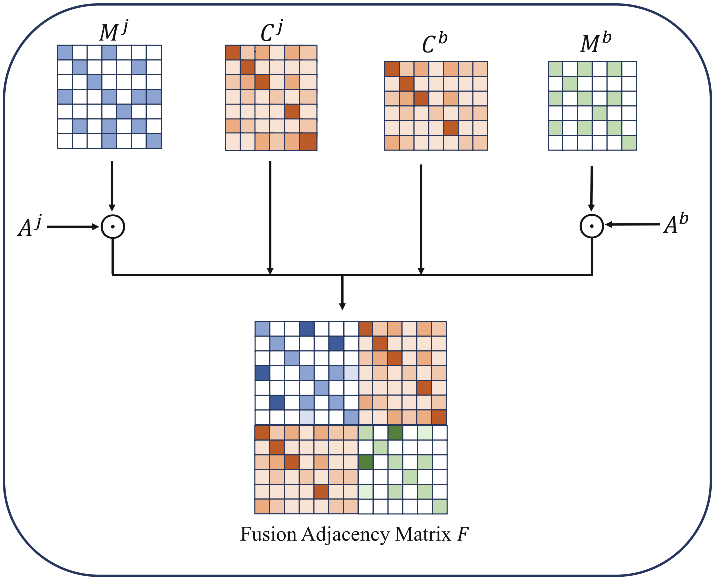


Fig. 4. Illustration of learning process of fusion gait graph

The graph convolution operation on vertex $v_{t,i}$ in fusion spatiotemporal gait graph can be formulated as follows:

$$f_{out}(v_{t,i}) = \sum\nolimits_{v_{t,i} \in B(v_{t,i})} \frac{1}{z_{t,i}(v_{t,i})} f_{in}(v_{t,i}) \cdot w(l_{t,i}(v_{t,i})) \tag{3}$$

where $v_{t,i}$ represents i-th graph vertex at t-th frame, f is the vertex feature map. The neighbor set $B(v_{t,i}) = \{v_{t,j} | d(v_{t,i}, v_{t,j}) \leq D\}$ denotes the sample region of convolution operation on vertex $v_{t,i}$. In this work, we set $D = 1$, that is, 1-distance neighbour vertices are considered. w is a weight function that provided weight parameters. $l_{t,i}$ is a label mapping function that partitions the neighbor set $B(v_{t,i})$ into a fixed number of K subset. In the layer, we divided the neighbor set into three subsets based on graph spatial configuration. The first subset contains the root node itself. The second subset contains the neighboring nodes that are closer to gravity center node selected (pelvis node) than root node. The third subset contains the neighboring nodes that are farther from the gravity center node than root node. Here, the implementation of graph convolution is formulated as follows:

$$X' = \textstyle\sum_{k=1}^{K} W_k X \tilde{F_k} \tag{4}$$

$$\tilde{F_k} = D_k^{-\frac{1}{2}} F_k D_k^{-\frac{1}{2}} \tag{5}$$

where K represents the spatial kernel size. It is equal to the number of subsets of neighbor set according to the partition strategy. F_k is the fusion adjacency matrix. $D_k^{ij} = \sum_j F_k^{ij} + \alpha$ represents the normalized diagonal matrix. α is used to avoid empty rows. W_k denotes the parameter matrix of convolution operation. Thus, the new spatial-temporal fusion gait graph representation with spatial gait abnormality features can be denoted as $G'_{st} = \{X'_t, F'\}_{t=1}^{T}$, where $X'_t \in R^{H' \times T \times (N^j + N^b)}$. Besides, a batch normalization layer is utilize to stabilize and accelerate the training of neural networks by normalizing the gait graph representation. A ReLU layers is used to introduces non-linearity in neural networks, overcoming the vanishing gradient problem and enabling faster convergence. Then, the G'_{st} serves as the input for the temporal convolution layer.

For the temporal convolutional layer, since the number of neighbors for each node is fixed as 2, it is straightforward to perform the graph convolution similar to the classical convolution operation. Specifically, a standard 2D convolution with 7×1 kernel is performed on the learned feature map X'.

4 Experiments

4.1 Dataset

In this work, 57 health participants were recruited to collect skeleton data of abnormal gait using a Kinect sensor, which accurately captures three-dimensional motion coordinate information of 32 human body joints. Each participant was asked to walk with six abnormal gait patterns (steppage gait, antalgic gait, circumduction gait, waddling gait, in-toeing gait, and out-toeing gait) and one normal gait on a treadmill. Our dataset contains a total of 53,988 gait samples, with each sample representing the locomotion coordinate information of 32 joints of the human body within a gait cycle.

For greater credibility and robustness of our model, the public walking gait dataset [37] has been used in this study. Nguyen et al. used 5-cm, 10-cm, and 15-cm soles and a 4-kg weight to cause abnormal gaits. There are 1 normal and 8 abnormal gaits acquired by 9 participants in this dataset. Each gait pattern contains 1200 frames of the skeleton data collected by Kinect v2.

The leave-one-subject-out (LOSO) cross-validation approach is employed to divide training data and test data. All samples for one subject were kept as testing data, while the samples from the remaining subjects were used for training the model.

4.2 Training setting and evaluation metrics

All experiments were conducted on the PyTorch deep learning framework with 1 single RTX 3090 GPU. Stochastic Gradient Descent (SGD) with Nesterov momentum (0.9) was employed as the optimization strategy. The training comprised 200 epochs with a learning rate initialized at 0.001 for the first 20 epochs, followed by a decrease to 0.0001 for the remaining epochs. The batch size was set to 256. To avoid overfitting, early-stopping method was considered in the training process.

For our multi-class gait classification task, model performance was evaluated by accuracy, macro-averaged Recall (macro-R), macro-averaged Precision (macro-P), and macro-averaged F1-score (macro-F1).

4.3 Evaluation of the generalization performance of our proposed method

We first compare our model with the state-of-the-art deep learning models in recent studies on our dataset. These models used for comparison include LSTM [5], CNN-LSTM [9], ST-GCN [22], STJA-GCN [27], 2s-ST-AGCN [28], and CTR-GCN [36]. The classification results of comparison are given in Table 1. Our model achieves the best performance across all four evaluation metrics with a large margin (macro-P of 98.62%, macro-R of 98.81%, macro-F1 of 98.56% and Accuracy of 98.74%), verifying the superiority of our proposed model. Also, we evaluate our model with these state-of-the-art methods on the public walking gait dataset. The classification results of comparison are given in Table 2. Our model achieved 100% on all four evaluation metrics. These results demonstrated that our model has excellent graph fusion learning capability to discover the most discriminative gait abnormality representations hidden in skeleton data.

Table 1. Comparison with state-of-the-art methods on our gait dataset

Methods	macro-P (%)	macro-R (%)	macro-F1 (%)	accuracy (%)
LSTM [5]	90.43	90.55	89.72	89.96
CNN-LSTM [9]	90.22	89.78	89.52	89.68
CTR-GCN [36]	91.78	91.31	91.22	91.82
ST-GCN [22]	92.44	92.56	92.15	92.33
STJA-GCN [27]	94.32	94.23	94.01	94.12
2s-ST-AGCN [28]	96.11	96.15	95.96	96.02
Our method	**98.62**	**98.81**	**98.56**	**98.74**

Besides, we further evaluate the recognition performance of our model for each gait pattern based on our gait dataset, using 2s-ST-AGCN and ST-GCN for comparison. The comparative results are presented in the confusion matrices

Table 2. Comparison with state-of-the-art methods walking gait dataset

Methods	macro-P (%)	macro-R (%)	macro-F1 (%)	accuracy (%)
LSTM [5]	93.34	93.51	93.39	93.85
CNN-LSTM [9]	94.31	94.78	95.29	94.72
CTR-GCN [36]	99.34	99.56	99.25	99.64
ST-GCN [22]	100.0	100.0	100.0	100.0
STJA-GCN [27]	99.11	99.85	99.76	99.78
2s-ST-AGCN [28]	100.0	100.0	100.0	100.0
Our method	**100.0**	**100.0**	**100.0**	**100.0**

in Fig. 5. Our model accurately identifies almost every gait pattern, while 2s-ST-AGCN and ST-GCN show poor recognition capabilities for antalgic gait. These results further show that our proposed model achieves the best graph fusion learning ability to discover the gait abnormality representation with rich interaction information between joints and bones.

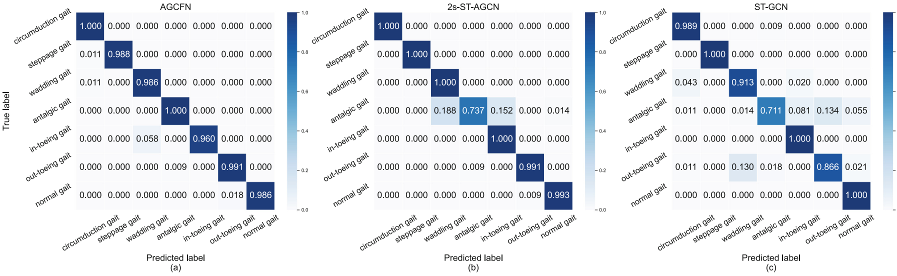


Fig. 5. Confusion matrices of different methods for recognition accuracy of each gait pattern

4.4 Evaluation of the complexity of our proposed method

In this experiment, we compare the complexity (including parameters (Params), floating point operations per second (FLOPS), and memory overhead (Memory)) of different graph-based models, as shown in Table 3. Params, FLOPS, and Memory are all calculated through an open-source tool called torchstat. Compared to these graph-based models, our model accelerates the training process and reduces the memory overhead and computational time. Overall, our model achieves both high generalization and low complexity.

Table 3. Comparative results of the complexity of different state-of-the-art graph-based methods

Methods	Param (M)	FLOPS (G)	Memory (MB)
ST-GCN [22]	1.78	0.213	7.14
CTR-GCN [36]	1.41	0.193	12.81
STJA-GCN [27]	2.83	0.356	8.81
2s-ST-AGCN [28]	3.43	0.458	9.72
Our method	**0.774**	**0.192**	**6.87**

4.5 Ablation experiments

In this experiment, we examine the effectiveness of the proposed cross-adaptive adjacency matrix. First, we set all the values of the cross-adaptive adjacency matrix to 0 (variant 1) to evaluate its graph fusion learning capability. Next, we set all the values to 1 (variant 2) to verify its adaptive graph learning capability. As shown in Table 4, removing the cross-adaptive adjacency matrix will harm the model's performance. This indicates that graph fusion learning and adaptive graph learning based on our proposed cross-adaptive adjacency matrix are beneficial for abnormal gait recognition. Additionally, the final adaptive fusion adjacency matrix learned by our model is shown in Fig. 6. The color intensity of each element in the matrix indicates the strength of the connection. This figure shows that our model is able to explore not only the intra-graph relationships within the joint gait graph and the bone gait graph but also their inter-graph relationships, which contain rich gait abnormality features associated with the impaired interactions of joints and bones. These results prove our view that the implicit interaction information between the joints and bones is also important for abnormal gait recognition.

Table 4. Ablation experiment results on the cross-adaptive adjacency matrix

Methods	macro-P (%)	macro-R (%)	macro-F1 (%)	accuracy (%)
Variant 1	94.87	94.67	94.81	94.98
Variant 2	96.77	96.81	96.63	96.51
Our method	**98.62**	**98.81**	**98.56**	**98.74**

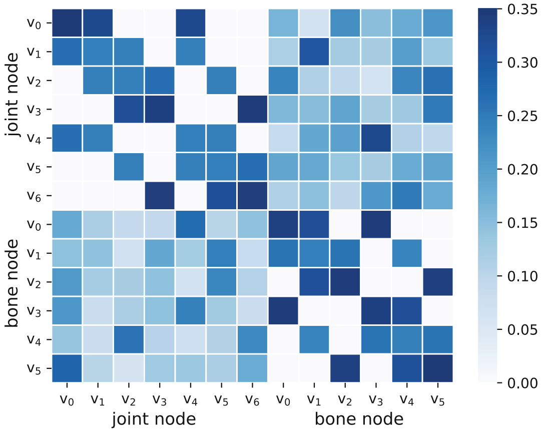


Fig. 6. Illustration of the learned adaptive fusion adjacency matrix

5 Conclusion

In this work, we proposed a novel adaptive graph convolutional fusion network (AGCFN) for skeleton-based abnormal gait recognition. The model parameterizes a cross-adaptive adjacency matrix to adaptively fuse the joint gait graph and the bone gait graph into a fusion gait graph. With this fusion gait graph, our model can discover not only the intra-graph relationships (i.e., the interaction information between human joints and the interaction information between human bones) within the joint gait graph and the bone gait graph but also their inter-graph relationships (i.e., the interaction information between human joints and bones). The inter-graph relationships contain rich abnormal gait feature information, greatly improving the generalization performance of our model. On our abnormal gait dataset, the proposed AGCFN achieves state-of-the-art performance and the lowest computation cost compared with other advanced methods.

Acknowledgements. This research was supported by the National Natural Science Foundation of China (Grant Number 82072043).

References

1. Saboor, A., Kask, T., Kuusik, A., Alam, M.M., Le Moullec, Y., Niazi, I.K., Ahmad, R.: Latest research trends in gait analysis using wearable sensors and machine learning: A systematic review. Ieee Access **8**, 167830–167864 (2020)
2. Nguyen, T.N., Huynh, H.H., Meunier, J.: Skeleton-based abnormal gait detection. Sensors **16**(11), 1792 (2016)
3. Chaaraoui, A. A., Padilla-López, J. R., & Flórez-Revuelta, F. (2015, May). Abnormal gait detection with RGB-D devices using joint motion history features. In 2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG) (Vol. 7, pp. 1-6). IEEE

4. Nieto-Hidalgo, M., Ferrández-Pastor, F.J., Valdivieso-Sarabia, R.J., Mora-Pascual, J., García-Chamizo, J.M.: A vision based proposal for classification of normal and abnormal gait using RGB camera. J. Biomed. Inform. **63**, 82–89 (2016)
5. Khokhlova, M., Migniot, C., Morozov, A., Sushkova, O., Dipanda, A.: Normal and pathological gait classification LSTM model. Artif. Intell. Med. **94**, 54–66 (2019)
6. Jun, K., Lee, D.W., Lee, K., Lee, S., Kim, M.S.: Feature extraction using an RNN autoencoder for skeleton-based abnormal gait recognition. IEEE Access **8**, 19196–19207 (2020)
7. Jun, K., Lee, Y., Lee, S., Lee, D.W., Kim, M.S.: Pathological gait classification using kinect v2 and gated recurrent neural networks. Ieee Access **8**, 139881–139891 (2020)
8. Lee, D. W., Jun, K., Lee, S., Ko, J. K., & Kim, M. S. (2019, July). Abnormal gait recognition using 3D joint information of multiple Kinects system and RNN-LSTM. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 542-545). IEEE
9. Sadeghzadehyazdi, N., Batabyal, T., Acton, S.T.: Modeling spatiotemporal patterns of gait anomaly with a CNN-LSTM deep neural network. Expert Syst. Appl. **185**, 115582 (2021)
10. Gao, J., Gu, P., Ren, Q., Zhang, J., Song, X.: Abnormal gait recognition algorithm based on LSTM-CNN fusion network. IEEE Access **7**, 163180–163190 (2019)
11. Elkholy, A., Hussein, M.E., Gomaa, W., Damen, D., Saba, E.: Efficient and robust skeleton-based quality assessment and abnormality detection in human action performance. IEEE J. Biomed. Health Inform. **24**(1), 280–291 (2019)
12. Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
13. Li, Y., Fu, X., & Zha, Z. J. (2021). Cross-patch graph convolutional network for image denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4651-4660)
14. Chen, C., Ma, W., Zhang, M., Wang, Z., He, X., Wang, C., ... & Ma, S. (2021, May). Graph heterogeneous multi-relational recommendation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 5, pp. 3958-3966)
15. Zhu, Y., Ma, J., Yuan, C., Zhu, X.: Interpretable learning based dynamic graph convolutional networks for alzheimer's disease analysis. Information Fusion **77**, 53–61 (2022)
16. Shehnepoor, S., Togneri, R., Liu, W., Bennamoun, M.: HIN-RNN: a graph representation learning neural network for fraudster group detection with no handcrafted features. IEEE transactions on neural networks and learning systems **34**(8), 4153–4166 (2021)
17. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., & Guo, M. (2018, October). Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In Proceedings of the 27th ACM international conference on information and knowledge management (pp. 417-426)
18. Yang, C., Pal, A., Zhai, A., Pancha, N., Han, J., Rosenberg, C., & Leskovec, J. (2020, August). MultiSage: Empowering GCN with contextualized multi-embeddings on web-scale multipartite networks. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2434-2443)
19. Luo, Y., Zou, J., Yao, C., Zhao, X., Li, T., & Bai, G. (2018, July). HSI-CNN: A novel convolution neural network for hyperspectral image. In 2018 International Conference on Audio, Language and Image Processing (ICALIP) (pp. 464-469). IEEE

20. Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926
21. Yao, L., Mao, C., & Luo, Y. (2019, July). Graph convolutional networks for text classification. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 7370-7377)
22. Yan, S., Xiong, Y., & Lin, D. (2018, April). Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1)
23. Tian, H., Li, H., Jiang, W., Ma, X., Li, X., Wu, H., & Li, Y. (2024). Cross-Spatiotemporal Graph Convolution Networks for Skeleton-Based Parkinsonian Gait MDS-UPDRS Score Estimation. IEEE Transactions on Neural Systems and Rehabilitation Engineering
24. Tian, H., Ma, X., Wu, H., Li, Y.: Skeleton-based abnormal gait recognition withspatio-temporalattention enhanced gait-structural graph convolutional networks. Neurocomputing **473**, 116–126 (2022)
25. Wu, J., Huang, J., Wu, X., Dai, H.: A novel graph-based hybrid deep learning of cumulative GRU and deeper GCN for recognition of abnormal gait patterns using wearable sensors. Expert Syst. Appl. **233**, 120968 (2023)
26. Shi, L., Zhang, Y., Cheng, J., & Lu, H. (2019). Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 12026-12035)
27. Yin, Z., Jiang, Y., Zheng, J., Yu, H.: STJA-GCN: A Multi-Branch Spatial-Temporal Joint Attention Graph Convolutional Network for Abnormal Gait Recognition. Appl. Sci. **13**(7), 4205 (2023)
28. Guo, R., Shao, X., Zhang, C., Qian, X.: Multi-scale sparse graph convolutional network for the assessment of Parkinsonian gait. IEEE Trans. Multimedia **24**, 1583–1594 (2021)
29. Pan, J., Lin, H., Dong, Y., Wang, Y., Ji, Y.: MAMF-GCN: Multi-scale adaptive multi-channel fusion deep graph convolutional network for predicting mental disorder. Comput. Biol. Med. **148**, 105823 (2022)
30. He, Y., Liu, X., Cheung, Y. M., Peng, S. J., Yi, J., & Fan, W. (2021, July). Cross-graph attention enhanced multi-modal correlation learning for fine-grained image-text retrieval. In Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval (pp. 1865-1869)
31. Pan, Z., Wu, F., & Zhang, B. (2023). Fine-grained image-text matching by cross-modal hard aligning network. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 19275-19284)
32. Duhme, M., Memmesheimer, R., & Paulus, D. (2021, September). Fusion-gcn: Multimodal action recognition using graph convolutional networks. In DAGM German conference on pattern recognition (pp. 265-281). Cham: Springer International Publishing
33. Hu, K., Wang, Z., Martens, K.A.E., Hagenbuchner, M., Bennamoun, M., Tsoi, A.C., Lewis, S.J.: Graph fusion network-based multimodal learning for freezing of gait detection. IEEE Transactions on Neural Networks and Learning Systems **34**(3), 1588–1600 (2021)
34. Dhawan, M., Sharma, S., Kadam, A., Sharma, R., & Kumaraguru, P. (2022). Game-on: Graph attention network based multimodal fusion for fake news detection. arXiv preprint arXiv:2202.12478

35. Tu, Z., Zhang, J., Li, H., Chen, Y., & Yuan, J. (2022). Joint-bone fusion graph convolutional network for semi-supervised skeleton action recognition. IEEE Transactions on Multimedia
36. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., & Hu, W. (2021). Channel-wise topology refinement graph convolution for skeleton-based action recognition. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 13359-13368)
37. Nguyen, T. N., & Meunier, J. (2018). Walking gait dataset: point clouds, skeletons and silhouettes. DIRO, University of Montreal, Tech. Rep, 1379

ConDGAD: Multi-augmentation Contrastive Learning for Dynamic Graph Anomaly Detection

Siqi Xia[1], Sutharshan Rajasegarar[1](✉), Lei Pan[1,2], Christopher Leckie[3], Sarah M. Erfani[3], and Jeffrey Chan[4]

[1] School of IT, Deakin University, Geelong, Australia
{xiasiq,srajas,l.pan}@deakin.edu.au
[2] Deakin Cyber Research and Innovation Centre, Deakin University, Geelong, Australia
[3] Computing and Information Systems, University of Melbourne, Melbourne, Australia
{caleckie,sarah.erfani}@unimelb.edu.au
[4] School of Computing Technologies, RMIT, Melbourne, Australia
jeffrey.chan@rmit.edu.au

Abstract. Anomaly detection on dynamic graphs is crucial for monitoring the security of industrial systems. The challenge in identifying anomalies in time-varying data arises from complex and flexible structures, compounded by the absence of labelling in the data. In particular, the representation of graph patterns and capturing the evolving nature of graphs become challenging due to time varying nature, i.e., dynamic graphs. Contrastive learning in graph-related contexts has gained considerable traction recently, primarily attributed to its label independence and the robustness in representations. In order to address the limitations in dynamic graph representation and anomaly detection, we propose a novel Contrastive learning-based dynamic graph anomaly detection framework (ConDGAD) to improve the time series data representation learning and prediction through dynamic graphs. This enables detection of multivariate time series anomalies at specific time window of measurement levels. ConDGAD first converts the multivariate time series data into dynamic graphs. Then multiple graph augmentations are performed and a novel contrastive learning process is applied on the dynamic graphs. This enable to train a model that can effectively capture the graph dynamics and perform accurate prediction, which subsequently is used for anomaly detection. Evaluation performed on widely used time series datasets, including SWaT and WADI, reveal that the ConDGAD has achieved improved recall and F1 scores for anomaly detection over the state-of-the-art methods. Ablation studies reveal the significance of the our proposed multi-augmented constrastive learning process in achieving the improved performance for anomaly detection on time series data via dynamic graphs.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 416–431, 2025.
https://doi.org/10.1007/978-3-031-78389-0_28

Keywords: Dynamic Graphs · Graph Augmentation · Anomaly Detection · Contrastive Learning

1 Introduction

Significant amount of inter-connected sensors and devices have been deployed in recent years, such as in Internet of Things (IoT), Cyber networks, transport networks and power grids. These devices generate voluminous time-series data. A challenge here is to develop efficient representations to capture the dynamics of the multivariate data for achieving improved prediction, classification and anomaly detection performances.

When facing with data having more complex patterns and dimensions, it is increasingly difficult for humans to manually process the information. This creates a need for automated methods to identify anomalies in such high-dimensional data in a timely manner. Further, the methods need to be robust to the changing data patterns over time, leading to stable detection results facing various anomaly situations.

The advancements in computing and deep learning have significantly enhanced anomaly detection in recent times. For example, Autoencoder (AE) based methods [1,9,10] use reconstruction error as a metric for detecting outliers. Long Short-Term Memory (LSTM)-based methods and various extensions, [20,21], have shown promising results in detecting anomalies in multivariate data. However, a common limitation of these methods is their inability to explicitly learn the hidden patterns between different time intervals or devices when the data is evolving.

Traditional algorithms that are designed to find patterns in data are often represented as a fixed length vector of features. In contrast, the data can be represented in the form of graphs [31]. Graph Convolutional Neural Networks (GCNs) [13,14] and graph attention networks (GATs) [29] are common deep learning based graph representation methods that can be used for time series data representations. However, facing the time-series data, typical GNNs are unable to model the changing nature of relations and the dynamic behavior of nodes and edges effectively. Hence, dynamic graph representation are suitable in these circumstances, that can capture the time dynamics efficiently.

In [24], EvAnGCN is proposed that uses evolving graph for anomaly detection in blockchains. However, generalised embeddings have not been obtained for improving robust representations. In [23], one class GCN has been used for anomaly detection in blockchain data, however, dynamic graphs is not considered for analysis. In order to take advantages of dynamic graph representations for time series data and also the use of contrastive learning to obtain more robust embedding and prediction, we propose a novel **Con**trastive Learning-based **D**ynamic **G**raph **A**nomaly **D**etection framework (ConDGAD), as shown in Figure 1. The proposed system first transform the time series data into dynamic graphs, followed by graph augmentations and novel contrastive learning process to obtain embeddings with enhanced robustness. A transformer and a predictor module have also been integrated for efficiently process the dynamic graph embeddings, capturing the long-term dynamics of the time series data. The learned predictor is subsequently used for detecting the anomalies.

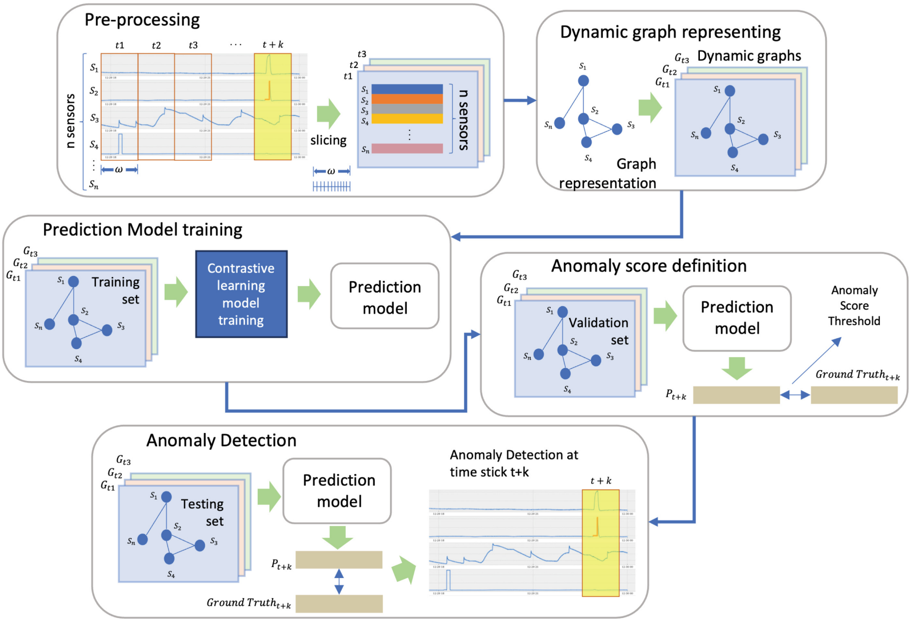


Fig. 1. ConDGAD framework: Time series data from multiple sensors are first sliced into multiple time windows of measurements during the pre-processing stage. Each slice is then converted into a graph, and the collections of these graphs over time produces the dynamic graphs. A novel contrastive learning process, involving multi-augmentation, encoding and transformation, is used to obtain the dynamic graph embeddings and the prediction model. The trained model is then used for detecting the anomalous time window of measurements in the multi-variate time series

In summary, the main contributions of this work are:

- We propose a dynamic graph representation and forecasting framework for time series anomaly detection. The core component of the framework comprises a novel contrastive learning methodology based on multi-augmentation for dynamic graphs and a transformer module built upon series of encoded graph embeddings under different augmentation schemes to obtain a compact embedding (concentrating module) that capture the changing graph patterns and the feature variations among time series.
- subsequently, a time series forecasting process (predictor) is learned to make long-term time-series predictions, which benefits from the evolving pattern of the short-term dynamic graphs. The concentrating and forecasting models are trained with contrastive learning using multiple-augmentations. The trained predictor is then used for anomaly detection.

- Evaluations are performed on real-world industrial benchmark datasets. The results reveal that the proposed model is superior compared to other existing methods, in terms of detecting accuracy. Ablation study further confirms that the proposed contrastive learning and prediction components contribute significantly for achieving improved anomaly detection performance.

2 Related Work

We briefly review the recent works on dynamic graphs and anomaly detection, in this section.

A significant advancement in spatial and temporal representations is the development of methods that effectively combine both temporal and spatial aspects in dynamic graphs. In DyRep, [28], an inductive deep representation learning framework is proposed that efficiently produce low-dimensional node embeddings that evolve over time, driving the dynamics of communication and association between nodes in dynamic graphs. The Dynamic-GTN model, [11], is designed to learn node embeddings in a continuous-time dynamic graph. Some of the temporal focused works includes TGNs [25] and Temporal-GAT [27]. They combine the time series model with graph structured data.

An advanced way to represent a dynamic graph is to combine both temporal and spatial aspects. In [4], a node representation learning architecture based on graph convolutional networks (GCNs) is presented, which integrates multi-dimensional features of node degree, clustering coefficient, and time evolution for dynamic networks, and utilises algorithms like LSTM and Multi-Head Attention to capture the time evolution patterns. In [7], dynamic spatial-temporal graph convolutional networks is proposed, addressing challenges in traffic flow prediction by considering the dynamic nature of spatial dependencies in traffic networks. In [35], authors discuss temporal knowledge graphs and propose time-aware representation learning models for inferring missing temporal facts, addressing the dynamic interactions of entities along a timeline.

Several deep learning based anomaly detection has been proposed in the literature. Graph Autoencoders (GAEs) [12] are an unsupervised learning method, mapping nodes to a potential vector space and reconstructing graph information. The reconstruction errors are used to detect anomalies. Temporal Convolutional Networks (TCNs) have shown advantages in addressing temporal dependencies. Enhancements to TCNs [16] have been proposed to better capture anomalies within domains like traffic flow, where understanding long-term temporal correlations and spatial characteristics. GDN, in [5], introduces an attention-based graph neural network (GNN) method that learns the dependency relationships among sensors and effectively detects and elucidates anomalies in these relationships. In [3], a forecasting framework for detecting anomalies in multivariate time series is introduced, which centered around a dynamic graph encoder. This encoder utilizes evolving graphs to analyze both the short-term changes and long-term consistent relationships among different time series, enhancing anomaly detection capabilities. In this work, a dynamic graph anomaly detection method is proposed, where an anomalous graph will correspond to the time window of measurements that exhibits abnormal behaviors contributed collectively from multiple sensors during that time window of measurements.

Graph Contrastive Learning is at the forefront of research, establishing a new paradigm for learning graph representations without the need for human annotations. These approaches aim to learn informative and discriminate embeddings for nodes or graphs without requiring labeled data. In [18], MaskGAE is proposed, which is a self-supervised learning framework for graph-structured data. It differs from previous graph autoencoders by using masked graph modeling as a pretext task, aiming to reconstruct missing parts of a graph. In [19], $S^3 - CL$ is proposed that combines structural and semantic contrastive learning. This approach allows even simple neural networks to learn expressive node representations that capture valuable global structural and semantic patterns. Contrastive Graph Few-Shot Learning [33], describes a framework where a GNN is pre-trained using contrastive learning and then applied to few-shot node classification. This methodology leverages self-distilled learning phases to enhance the GNN's performance on few-shot tasks. Graph Contrastive Learning with Augmentations [32] is an example to include different graph augmentation methods for contrastive learning to train graph representations with better generalizability, transferrability, and robustness. Our work shares similarity with GraphCL, as we also include graph augmentations for contrastive learning while the difference is in our contrastive learning process for dynamic graphs. In general, the above existing works do not consider effective contrative learning that can capture generalised robust embeddings for the graph dynamics and achieve high anomaly detection.

3 Methodology

The proposed framework comprises a dynamic graph creation component, a training component and an application component. The dynamic graph creation component creates the dynamic graph from time series. The dynamic graph data is processed via multi augmentation, novel contrastive learning process and prediction process to obtain a robust embedding, as well as an effective predictor of the embeddings. The trained model is then further used for anomaly detection purposes.

3.1 Creating Dynamic Graphs from Time Series

Before the training process of ConDGAD framework, we convert the time series signals to dynamic graph representations. The original data are composed of sensor data (i.e., multivariate time series) from N sensors collected over a time period of T_{train}.The time period is divided into several window of measurements, with temporal interval size (window size) of ω. This sensor data is denoted as $\mathbf{n}_{\text{train}} = [\mathbf{n}_{\text{train}}^{(t1)}, \cdots, \mathbf{n}_{\text{train}}^{(T_{\text{train}})}]$, which is used to train our approach. At each time tick t, the sensor values $\mathbf{n}_{\text{train}}^{(t)} \in \mathbf{R}^N$ form a N-dimensional vector representing the values from N sensors. Following the usual unsupervised anomaly detection formulation, the training data is assumed to consist solely of normal data. Our goal is to train the representations and then detect anomalies in the (unseen) testing data, which comes from the same N sensors but over a separate set of T_{test} time ticks. The test data is denoted as $\mathbf{n}_{\text{test}} = [\mathbf{n}_{\text{test}}^{(t1)}, \cdots, \mathbf{n}_{\text{test}}^{(T_{\text{test}})}]$.

We extent the graph learning process in [5] to design a graph based framework by representing sensors in graph relations. The representations of sensor i are represented by the attributes $\mathbf{v}_i$. This embedding $\mathbf{v}_i$ is defined by slicing from time slot ω. We sliced the time slot to n evenly and concatenate the mean measure at each time spot to form the attributes for the sensor as $\mathbf{v}_i$.

In the absence of prior information, the candidate relations for sensor i include all sensors except itself. To determine the relations of sensor i among these candidates, we compute the similarity between the embedding vector of node i and the attributes of its candidates $j \in C_i$, where C_i is the sensor set except node i:

$$e_{ji} = \frac{\mathbf{v}_i^\top \mathbf{v}_j}{\|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\|} \quad \text{for } j \in C_i \tag{1}$$

$$A_{ji} = 1\{j \in \text{TopAlpha}(\{e_{ki} : k \in C_i\})\} \tag{2}$$

Here, e_{ji} is the normalized dot product between the embedding vectors of sensor i and candidate $j \in C_i$. We then select the top Alpha normalized dot products, where TopAlpha denotes the indices of the top-Alpha values among its input (i.e., the normalized dot products). A is the adjacency matrix of the graph. The nodes of the graph are the sensors, and the similarity values obtained above are used to form the edges between the nodes, there by forming a single graph representation for the time window ω of time series measurements. The collection of these graphs formed over the observed time period (number of windows) forms a set of dynamic graphs.

3.2 ConDGAD Process

After the dynamic graphs are built, we use these dynamic graphs for prediction model training. Figure 2 shows the overall framework for the training. The dynamic graphs at different time slots $t_1, t_2 ... t_n$ are first augmented using different augmentation methods (e.g., adding/deleting nodes or edges) for subsequent contrastive learning purposes. Following this, the augmented graphs are encoded into hidden representations through a graph encoder module including the information of graph patterns and the node features. The time series encoded representations are then forwarded to a transformer structure to generate compact embeddings. The objective of the previous steps are to predict the future graph embeddings at time $t + k$ using prior graphs as a basis. The compact embeddings and the predicted embeddings from various augmentated methods are jointly used to optimise the framework contrastively. At the same time, the predicted embeddings are also optimised by comparing with the original graph embeddings after augmentation at time $t + k$. After the model has been trained, we are able to obtain more robust, generalised and compressed representations for time-series data, which is then used to train the prediction model.

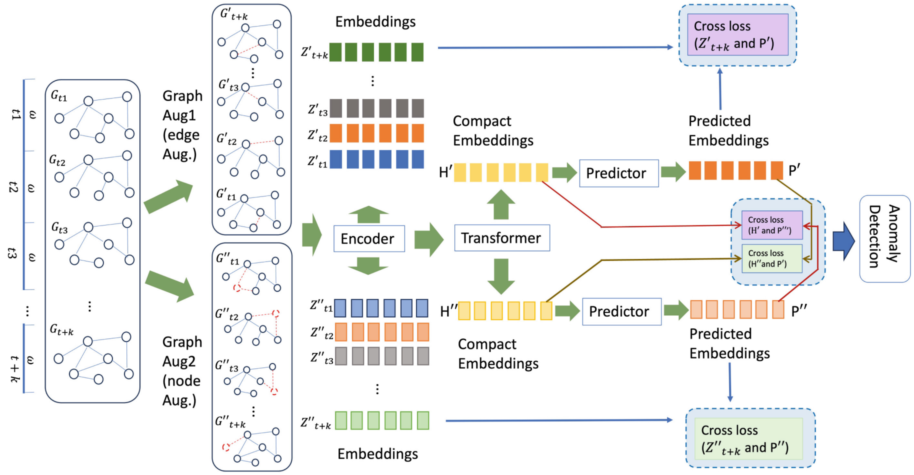


Fig. 2. ConDGAD training process: this ConDGAD framework takes the dynamic graph snapshots from original data to train a model that can effectively predict the graph embedding at $(t + k)$. This training process includes graph augmentation based novel contrastive learning. The output predicted embeddings is further used for anomaly detection

Figure 2 illustrates the comprehensive workflow of our graph-based deep learning model, which involves multiple stages of graph augmentation, encoding, embedding transformation, and predictor model training.

A sequence of graphs (G_{t1}, G_{t2}, G_{t3}), representing the graphs at different time slots t_1, t_2, t_3 is obtained from the timeseries data, as explained in the above sections. They are then subjected to two distinct augmentation strategies, resulting in two sets of augmented graphs; Graph Aug1 and Graph Aug2. These augmented graphs, denoted as $G'_{t1}, G'_{t2}, G'_{t3}$ for the edge augmentation and $G''_{t1}, G''_{t2}, G''_{t3}$ for the node augmentation, undergo structural modifications, as depicted by red dashed lines indicating alterations in the graph topology.

Subsequently, an encoder processes these augmented graphs to generate embeddings $Z'_{t1}, Z'_{t2}, Z'_{t3}$ and $Z''_{t1}, Z''_{t2}, Z''_{t3}$, respectively, which represent the encoded features of the graphs at respective time steps. These embeddings are then fed into a transformer, which produces compact embeddings H' and H''. These compact embeddings are then passed through a predictor to generate predicted embeddings P' and P''.

The entire process is orchestrated to facilitate model training, wherein the goal is to learn effective (robust and generalised) representations and predictions from the graph sequences. The timeline on the left, labeled $t1, t2, t3$ with intervals ω, suggests a temporal aspect to the graph data, indicating that the model is designed to handle dynamic or time-evolving graphs. This methodology highlights the integration of graph augmentation techniques, deep encoding mechanisms, and advanced embedding transformations to enhance the predictive performance of the model. This temporal aspect underscores the model's capability to handle dynamic, time-evolving graphs.

The final stage involves model training, where the objective is to learn effective representations and predictions from the sequence of graph data. The model training involves contrastive learning between the compact and predicted embeddings from the augmentations. The learning is conducted cross-wise, which means the compact and predicted embeddings for contrastive learning are from different augmentation methods. At the same time, the predicted embeddings are compared with the original $t + k$ graph augmentation embeddings. After model training, predicted embeddings re generated that can better represent the dynamic graphs and further used for anomaly detection.

Graph Augmentation In the graph augmentation section, two different augmentation method has been used based on the dynamic graphs.

The first augmentation is edge augmentation. Based on graph in different time slides, some edges are randomly chosen and the connections are removed. The second augmentation is node augmentation. Also based on graph in different time slides, we randomly select some nodes and dropout the selected nodes using attribute masking methods. For both augmentation methods, inputs are the dynamic graph from different time slots G_{t1}, G_{t2}, G_{t3} and the outputs are also graphs, but having different graph edge relations, denoted as$G'_{t1}, G'_{t2}, G'_{t3}$ and different nodes, denoted as $G''_{t1}, G''_{t2}, G''_{t3}$.

Graph Encoder Module The graph encoder module transforms dynamic correlation graphs into low-dimensional feature representations to capture temporal dynamics between graphs and the intrinsic structure within each graph. Subsequently, a graph convolution layer is applied to each graph, using the latent representation of its corresponding segment as node features to derive the hidden representations for each graph. The encoder is the GCN based graph encoder to represent the graph patterns. The encoder has inputs of augmented graph with nodes, edges and attributes of each graphs as $G'_{t1}, G'_{t2}, G'_{t3}$ and $G''_{t1}, G''_{t2}, G''_{t3}$. The outputs are embeddings for these graph slides $Z'_{t1}, Z'_{t2}, Z'_{t3}$ and $Z''_{t1}, Z''_{t2}, Z''_{t3}$.

Temporal Contrastive Learning The Temporal Contrasting module uses a novel contrastive loss process. A contrastive loss is used to extract temporal features in the latent space using an autoregressive model, extending the work in [8] for time series signals. In the work [8], the contrastive loss used aims to reduce the dot product between the predicted representation and the true representation of the same sample, while increasing the dot product with the representations of other samples in the minibatch. While for the loss proposed in our approach, it is built from the contrastive loss of maximizing the difference between compact embeddings from the dynamic graphs generated based on different augmentations and predictions from another augmentation methods. The cross entropy loss is for minimizing the predicted embeddings with the augmented embeddings in next time slot. The contrastive loss is designed to learn more robust representations for the dynamic graphs with time differences while the cross entropy loss has the purpose to mimic the original augmented dynamic graph embeddings in the future to enhance the predictions. The novel contrastive loss constructed based on compact and prediction embeddings helps the model to learn an effective predictive embedding that can be further used for anomaly detection.

Given the embedding representations for the augmented graphs $Z'_{t1}, Z'_{t2}, Z'_{t3}$ and $Z''_{t1}, Z''_{t2}, Z''_{t3}$, the autoregressive model f_{a} summarizes all $Z'_{\leq t}$ and $Z''_{\leq t}$ into compact vectors H' and H''. The compact vectors H' and H'' are then used to predict the future representations. To predict future time steps, we use a log-bilinear model that preserves the mutual information between the input Z'_{t+k} or Z''_{t+k} and H', or H'' defined as $f_k(Z, H) = \exp\left((W_k(H))^T Z_{t+k}\right)$, Z_{t+k} is the true future embeddings from the original graph and W_k is a linear function that maps H back into the same dimension as Z_t. The contrastive loss aims to maximize the dot product between the predicted representation and the true one of the same sample, while maximizing the dot product with other samples $\mathcal{N}_{t,k}$ within the mini batch, which is the compact representations cross-over. Accordingly, we propose two losses $\mathcal{L}^{min}$ and $\mathcal{L}^{max}$ based on two different augmentation methods and add them up for final definition. $\mathcal{L}^{max}$ represents the contrastive learning loss while $\mathcal{L}^{min}$ is the cross entropy loss between predictions and oringal ground truth embeddings after augmentations.

$$\mathcal{L}^{min} = CE(Z'_{t+k}, P') + CE(Z''_{t+k}, P''), \tag{3}$$

where, $CE(.)$ is the contrastive error. Similarly, $\mathcal{L}^{max}$ is defined as :

$$\mathcal{L}^{max} = \mathcal{L}^{max}_{A1} + \mathcal{L}^{max}_{A2} \tag{4}$$

$$\mathcal{L}^{max}_{A1} = -\frac{1}{K}\left[\sum_{k=1}^{K} \log \frac{\exp\left((W_k(H'))^T P''\right)}{\sum_{n\in\mathcal{N}_{t,k}} \exp\left((W_k(H'))^T Z''_n\right)}\right] \tag{5}$$

$$\mathcal{L}^{max}_{A2} = -\frac{1}{K}\left[\sum_{k=1}^{K} \log \frac{\exp\left((W_k(H''))^T P'\right)}{\sum_{n\in\mathcal{N}_{t,k}} \exp\left((W_k(H''))^T Z'_n\right)}\right] \tag{6}$$

Inspired by [8], we employ a Transformer as our autoregressive model. The Transformer architecture comprises repeated blocks of multi-headed attention (MHA) followed by an MLP block. The MLP block includes two fully-connected layers with a ReLU activation function and a dropout layer in between. Our Transformer adopts pre-norm residual connections to ensure more stable gradient updates [30]. We stack L identical layers to produce the final feature set.

Inspired by the BERT model [6], we add a token $c \in \mathbb{R}^h$ to the input, acting as a representative context vector in the output. The Transformer's process starts by applying the input features $\mathbf{z}_{\leq t}$ to a linear projection layer W_{tran}, which maps the features to the hidden dimension: $W_{\text{tran}} : \mathbb{R}^d \rightarrow \mathbb{R}^h$. The output of this projection is $\tilde{\mathbf{z}} = W_{\text{tran}}(\mathbf{z}_{\leq t})$, where $\tilde{\mathbf{z}} \in \mathbb{R}^h$. We then concatenate the context vector with the transformed features, forming the input to the first layer: $\psi_0 = [c; \tilde{\mathbf{z}}]$.

The sequence ψ_0 is processed through the Transformer layers as follows:

$$\tilde{\psi}^{(\ell)} = \text{MHA}(\text{Norm}(\psi^{(\ell-1)})) + \psi^{(\ell-1)}, \quad 1 \leq \ell \leq L; \tag{7}$$

$$\psi^{(\ell)} = \text{MLP}(\text{Norm}(\tilde{\psi}^{(\ell)})) + \tilde{\psi}^{(\ell)}, \quad 1 \leq \ell \leq L. \tag{8}$$

Finally, we extract the context vector from the last layer's output, denoted as $c_t = \psi_0^{(L)}$. This context vector serves as the input to the subsequent contextual contrasting module.

3.3 Anomaly Detection Process

After training the model, the aim is to determine the anomaly at time tick t for the overall sensor systems.

After training the model, validation data is passed through the pre-trained model to get the predictions P^{val} which is then compared with the original graph embeddings Z^{val} at $t + k$. The error is calculated using the MSE as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (P_i^{val} - Z_i^{val})^2 \tag{9}$$

A threshold for the error is required to determine the normal and anomaly. The Threshold TH is determined with the validation errors as $TH = mean + 2 * sdv$, where $mean$ is the mean of the errors and std is the standard deviation of the errors.

During testing process, testing data will encoded to form dynamic graphs, and then passed through to the trained model to obtain the prediction P^{test}. The error between the predicted embedding and the original (data) graph embedding Z^{test} at $t + k$ is calculated, and compared with the threshold TH. If the error exceeds the threshold, the time tick at $t + k$ will be declared as anomaly, which represents that the graph at time $t+k$ is anomalous. In other words, the collection of sensor measurements obtained from the set of sensors at time $t + k$ has been showing anomalous behavior, collectively.

4 Experiments

4.1 Experiments Design

In our experiments, we utilised the publicly available implementations of baseline models, and the hyper-parameters of these models are based on the values specified in their original research papers, if accessible. The baselines and the proposed models are tested on two time-series data for representation and anomaly detection based on various evaluation matrices.

Datasets: In this study, two real-world benchmark datasets are used for the evaluation of time series anomaly detection method

1. SWAT: The Secure Water Treatment (SWaT) dataset [22] originates from a real-world water treatment facility that produces filtered water and includes 51 features. The dataset captures 11 days of per-second operational data, with 7 days representing normal operation and 2 days containing attack scenarios. For our experiments, we downsampled this data to one data point every 10 seconds. Because the unstability for the beginning of the data, the first 6 hours data are eliminated.
2. WADI: The Water Distribution (WADI) dataset [2] was gathered from a water distribution testbed featuring 127 attributes. This dataset serves as an extension of SWaT, incorporating a larger number of sensors and actuators. It encompasses 16 days of operational data, with 14 days of normal operations and 2 days of attack scenarios. Similarly, we downsampled this dataset to one data point every 10 seconds for our experiments.

Additionally, datasets are normalized by min-max scaler before the training.

4.2 Evaluation Matrix

We evaluate the detection performance with metrics, i.e., The F1-score is defined as: $F1 = \frac{2\times\mathrm{Prec}\times\mathrm{Rec}}{\mathrm{Prec}+\mathrm{Rec}}$ where Precision (Prec) and Recall (Rec) are given by:$\mathrm{Prec} = \frac{TP}{TP+FP}$, $\mathrm{Rec} = \frac{TP}{TP+FN}$. Here, TP, TN, FP, and FN are the numbers of true positives, true negatives, false positives, and false negatives, respectively. It evaluate the performance by dynamically assessing the detector's true positive rate and false positive rate across different thresholds which is a commonly employed measure to validate the performance of anomaly detectors.

4.3 Experiment Results

Anomaly Detection Accuracy Table 1 presents the results of anomaly detection using different methods on two distinct datasets: SWaT and WADI. The performance of each method is evaluated using three metrics: Precision (Pre), Recall (Rec), and F1-Score (F1). The methods compared include Principal Component Analysis (PCA) [26], Autoencoder (AE) [1], Fast-Bayesian (FB) [15], Long Short-Term Memory Variational Autoencoder (LSTM-VAE) [20], Multiple Additive Regression Trees Generative Adversarial Network (MAD-GAN) [17], Graph Deviation Network (GDN) [5], , correlation-aware spatial-temporal graph learning (CST-GL) [34] and our proposed method ConDGAD.

Results with SWaT Dataset: Table 1 presents the results of anomaly detection using various methods on two different datasets, SWaT and WADI, which contain ground-truth labeled anomalies. The performance of each method is evaluated using three metrics: Precision (Prec), Recall (Rec), and F1-score (F1).

Table 1. Results of Anomaly detection based on different dataset with ground-truth labelled anomalies

Method	SWaT			WADI		
	Prec	Rec	F1	Prec	Rec	F1
PCA	24.92	21.63	0.23	39.53	5.63	0.10
KNN	7.83	7.83	0.08	7.76	7.75	0.08
FB	10.17	10.17	0.10	8.60	8.60	0.09
AE	72.63	52.63	0.61	34.35	34.35	0.34
LSTM-VAE	93.52	56.78	0.71	84.61	20.52	0.33
MAD-GAN	95.45	60.74	0.74	48.62	30.29	0.37
GDN	**97.32**	65.79	0.78	**92.62**	34.4	0.50
CST-GL	89.28	74.57	0.81	80.15	**46.89**	0.58
ConDGAD	96.31	**75.76**	**0.84**	90.47	45.72	**0.61**

Part of the results, PCA, KNN, FB and MAD-GAN are from [17]. For the SWaT dataset, the methods' performances vary significantly. PCA achieves a low Precision of 24.92%, Recall of 21.63%, and an F1-score of 0.23. KNN and FB methods also show low performance with F1-scores of 0.08 and 0.10, respectively. AE shows a better balance with an F1-score of 0.61. The lower four baselines are more advanced methods. LSTM-VAE and MAD-GAN perform better, with F1-scores of 0.71 and 0.74, respectively. GDN and CST-GL also shows a strong performance with an F1-score of 0.78 and 0.81. The best F1 score on SWaT is achieved by ConDGAD of 0.84 as well as the best Recall 75.76% while the best Precision is from GDN of 97.32%

Results with WADI Dataset: For the WADI dataset, the methods generally show lower performance compared to SWaT. PCA, KNN, and FB have very low F1-scores of 0.10, 0.08, and 0.09, respectively. AE performs moderately with an F1-score of 0.34. LSTM-VAE, and MAD-GAN show a little bit higher performance, with F1-scores of 0.33, and 0.37, respectively. GDN and CST-GL are more advanced methods achieved a relatively better F1-score of 0.50 and 0.58. The highest performance on WADI is again by ConDGAD, with an F1-score of 0.61.

In summary, ConDGAD not only achieves the highest F1-scores across both datasets but also exhibits a more balanced performance in terms of Precision and Recall. This significant performance gap, especially when compared to more basic methods like PCA, KNN, and FB, underscores ConDGAD's superior capability in accurately detecting anomalies. The differences in F1-scores highlight ConDGAD's robustness and effectiveness, making it the most reliable method among those evaluated. These findings suggest that ConDGAD provides significant improvements in anomaly detection tasks, highlighting its potential for applications in complex datasets.

Ablation Study

Effectiveness of Different Parts To study the effectiveness of various parts in the training model process, we conducted an ablation study. We remove or replace part of the training model process and investigate how the changes will influence the anomaly detection results. Firstly, we focus on the most significant part is the contrastive learning from different augmentations. Secondly, the transformer is further be replaced by a simple neural network. The last trial is only using GCN for representation learning and prediction.

- The augmentations and contrastive learning part is removed while the process of encoder, transformer and predictor are kept and learnt. The encode embeds the dynamic graphs and further process it by the transformer and predictor, same as previously defined.
- The transformer is also removed to study its influence. A simple neural network is used to process the dynamic graph embedding outputs from the encoder based on the historic time slots for predictor. Thus, no transformer mechanism is used to obtain the compact embeddings.
- The predictor is further removed and the prediction is defined and represented by the embeddings of all historic time slot dynamic graphs through GCN. These embeddings is further used for anomaly detection, same way as in the testing period, as discussed before.

Table 2. Anomaly detection accuracy for different datasets based on various changes to the training framework (ablation study)

Method	SWaT			WADI		
	Prec	Rec	F1	Prec	Rec	F1
ConDGAD	**96.31**	**75.76**	**0.84**	**90.47**	**45.72**	**0.61**
w/o CONT	85.53	54.70	0.67	83.15	34.96	0.49
w/o TRANS	73.24	31.03	0.44	71.62	22.49	0.34
Only GCN	54.31	19.51	0.34	43.56	10.87	0.17

Table 2 shows the results of ablation study. The results highlight the differences in anomaly detection accuracy for the SWaT and WADI datasets when various components of the ConDGAD model are modified or removed. The ConDGAD method consistently achieves the highest performance across both datasets. This indicates its robust ability to detect anomalies accurately.

When the contrastive learning component is removed (w/o CONT), there is a significant drop in performance is observed. For the SWaT dataset, the F1-score decreases from 0.84 to 0.67, and for the WADI dataset, it drops from 0.61 to 0.49. This reduction underscores the importance of the contrastive learning component in enhancing the model's detection capabilities. Further, removing the transformer component (w/o TRANS) results in even lower performance, with F1-scores of 0.44 for SWaT and 0.34 for WADI. This shows that the transformer component is crucial for maintaining the model's effectiveness in anomaly detection. The "Only GCN" configuration, which relies solely on the GCN component, exhibits the lowest performance. This significant decline highlights that the GCN component alone is insufficient for accurate anomaly detection and that the integration of multiple components in the ConDGAD model is essential for achieving high performance.

TopAlpha Influence In creating dynamic graphs from time series data, we have selected the top Alpha normalized dot products, where TopAlpha denotes the indices of the top-Alpha values among its inputs. We aim to investigate the dynamic changes of graphs for each batch that the value of Alpha is chosen to form graphs, which are further processed. We conducted trials to investigate the influence of different choice of Alphas. From Table 3 results, it is illustrated that the value of Alpha can influence the final results and for larger choice of Alpha, only the recall will be influenced while if the Alpha is too small, both precision and recall can be influenced. This may result from that the too many indices will result in more unnecessary data for predictions and influence the final detection results and too little indices will not be able to provide sufficient information for latter embeddings and contrastive learning to learn.

Table 3. Anomaly detection accuracy for SWaT based on different TopAlpha choices (ablation study)

Method	SWaT		
cline2-4	Prec	Rec	F1
Alpha = 5	**96.31**	**75.76**	**0.84**
Alpha = 1	77.53	35.62	0.49
Alpha = 9	95.95	65.51	0.77

5 Conclusion

A novel Contrastive Learning-based Dynamic Graph Anomaly Detection framework is proposed to enhance the performance of time series data representation learning and robust predictions of anomalies through dynamic graphs. ConDGAD encodes the time series data into dynamic graph representations and incorporates novel contrastive learning using multiple graph augmentation methods and a transformer to embed and predict the dynamic graph data structure. Experiments using various datasets have been carried out, and the ConDGAD has shown advantages compared to other time series anomaly detection methods.

References

1. Aggarwal, C.C., et al.: Data mining: the textbook, vol. 1. Springer (2015)
2. Ahmed, C.M., Palleti, V.R., Mathur, A.P.: Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In: Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks. pp. 25–28 (2017)
3. Chen, K., Feng, M., Wirjanto, T.S.: Multivariate time series anomaly detection via dynamic graph forecasting. arXiv preprint arXiv:2302.02051 (2023)
4. Chen, Y., Ding, F., Zhai, L.: Multi-scale temporal features extraction based graph convolutional network with attention for multivariate time series prediction. Expert Syst. Appl. **200**, 117011 (2022)
5. Deng, A., Hooi, B.: Graph neural network-based anomaly detection in multivariate time series. In: Proc. of the AAAI. vol. 35, pp. 4027–4035 (2021)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Diao, Z., Wang, X., Zhang, D., Liu, Y., Xie, K., He, S.: Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 890–897 (2019)
8. Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C.K., Li, X., Guan, C.: Time-series representation learning via temporal and contextual contrasting. arXiv preprint arXiv:2106.14112 (2021)
9. Erfani, S.M., Rajasegarar, S., Karunasekera, S., Leckie, C.: High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. Pattern Recogn. **58**, 121–134 (2016)
10. Hdaib, M., Rajasegarar, S., Pan, L.: Quantum deep learning-based anomaly detection for enhanced network security. Quantum Machine Intelligence **6**(1), 26 (2024)
11. Hoang, T.L., Ta, V.C.: Dynamic-gtn: Learning an node efficient embedding in dynamic graph with transformer. In: Pacific Rim International Conference on Artificial Intelligence. pp. 430–443. Springer (2022)
12. Hou, Z., Liu, X., Cen, Y., Dong, Y., Yang, H., Wang, C., Tang, J.: Graphmae: Self-supervised masked graph autoencoders. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 594–604 (2022)
13. Jack, D., Erfani, S., Chan, J., Rajasegarar, S., Leckie, C.: It's pagerank all the way down: Simplifying deep graph networks. In: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM). pp. 172–180. SIAM (2023)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
15. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. pp. 157–166 (2005)
16. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 156–165 (2017)
17. Li, D., Chen, D., Jin, B., Shi, L., Goh, J., Ng, S.K.: Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In: International conference on artificial neural networks. pp. 703–716. Springer (2019)
18. Li, J., Wu, R., Sun, W., Chen, L., Tian, S., Zhu, L., Meng, C., Zheng, Z., Wang, W.: What's behind the mask: Understanding masked graph modeling for graph

autoencoders. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1268–1279 (2023)

19. Liu, S., Qu, M., Zhang, Z., Cai, H., Tang, J.: Structured multi-task learning for molecular property prediction. In: International conference on artificial intelligence and statistics. pp. 8906–8920. PMLR (2022)
20. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: Lstm-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148 (2016)
21. Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al.: Long short term memory networks for anomaly detection in time series. In: Esann. vol. 2015, p. 89 (2015)
22. Mathur, A.P., Tippenhauer, N.O.: Swat: A water treatment testbed for research and training on ics security. In: 2016 international workshop on cyber-physical systems for smart water networks (CySWater). pp. 31–36. IEEE (2016)
23. Patel, V., Pan, L., Rajasegarar, S.: Graph deep learning based anomaly detection in ethereum blockchain network. In: International conference on network and system security. pp. 132–148. Springer (2020)
24. Patel, V., Rajasegarar, S., Pan, L., Liu, J., Zhu, L.: Evangcn: Evolving graph deep neural network based anomaly detection in blockchain. In: International Conference on Advanced Data Mining and Applications. pp. 444–456. Springer (2022)
25. Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., Bronstein, M.: Temporal graph networks for deep learning on dynamic graphs. arXiv preprint arXiv:2006.10637 (2020)
26. Shyu, M.L., Chen, S.C., Sarinnapakorn, K., Chang, L.: A novel anomaly detection scheme based on principal component classifier. In: Proc. of the IEEE foundations and new directions of data mining workshop. pp. 172–179. IEEE Press (2003)
27. Tak, H., Jung, J.w., Patino, J., Kamble, M., Todisco, M., Evans, N.: End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. arXiv preprint arXiv:2107.12710 (2021)
28. Trivedi, R., Farajtabar, M., Biswal, P., Zha, H.: Dyrep: Learning representations over dynamic graphs. In: Intl. conf. on learning representations (2019)
29. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
30. Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D.F., Chao, L.S.: Learning deep transformer models for machine translation. arXiv preprint arXiv:1906.01787 (2019)
31. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**(1), 4–24 (2020)
32. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Adv. in neural info. processing sys. **33**, 5812–5823 (2020)
33. Zhang, C., Liu, H., Li, J., Ye, Y., Zhang, C.: Contrastive graph few-shot learning. arXiv preprint arXiv:2210.00084 (2022)
34. Zheng, Y., Koh, H.Y., Jin, M., Chi, L., Phan, K.T., Pan, S., Chen, Y.P.P., Xiang, W.: Correlation-aware spatial-temporal graph learning for multivariate time-series anomaly detection. IEEE Trans. on Neural Networks and Learning Systems pp. 1–15 (2023)
35. Zhu, C., Chen, M., Fan, C., Cheng, G., Zhang, Y.: Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In: Proc. of the AAAI. vol. 35, pp. 4732–4740 (2021)

Author Index

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15325, pp. 433–434, 2025.
https://doi.org/10.1007/978-3-031-78389-0