

LNCS 15318

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

# Pattern Recognition

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

18 Part XVIII

ICPR  
2024 INDIA



 Springer

MOREMEDIA 

# Lecture Notes in Computer Science

15318

## Founding Editors

Gerhard Goos  
Juris Hartmanis

## Editorial Board Members

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

Wen Gao, *Peking University, Beijing, China*

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

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

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


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


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


# Pattern Recognition

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

*Editors*

Apostolos Antonacopoulos   
University of Salford  
Salford, UK

Rama Chellappa   
Johns Hopkins University  
Baltimore, MD, USA

Saumik Bhattacharya   
IIT Kharagpur  
Kharagpur, India

Subhasis Chaudhuri   
Indian Institute of Technology Bombay  
Mumbai, India

Cheng-Lin Liu   
Chinese Academy of Sciences  
Beijing, China

Umapada Pal   
Indian Statistical Institute Kolkata  
Kolkata, India

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-3-031-78455-2

ISBN 978-3-031-78456-9 (eBook)

<https://doi.org/10.1007/978-3-031-78456-9>

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

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

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

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

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

If disposing of this product, please recycle the paper.

## President's Address

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

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

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

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

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

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

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

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

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

September 2024

Arjan Kuijper  
President of the IAPR

# Preface

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

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

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

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

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

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



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

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

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

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

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

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

October 2024

Umapada Pal  
Josef Kittler  
Anil Jain

# Organization

## General Chairs

Umapada Pal  
Josef Kittler  
Anil Jain

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

## Program Chairs

Apostolos Antonacopoulos  
Subhasis Chaudhuri  
Rama Chellappa  
Cheng-Lin Liu

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

## Publication Chairs

Ananda S. Chowdhury  
Wataru Ohyama

Jadavpur University, India  
Tokyo Denki University, Japan

## Competition Chairs

Richard Zanibbi  
Lianwen Jin  
Laurence Likforman-Sulem

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

## Workshop Chairs

P. Shivakumara  
Stephanie Schuckers  
Jean-Marc Ogier  
Prabir Bhattacharya

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

## **Tutorial Chairs**

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

## **Doctoral Consortium Chairs**

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

## **Organizing Chairs**

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

## **Organizing Committee**

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

## **Finance Chairs**

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

## **Awards Committee Chair**

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

## Sponsorship Chairs

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

## Exhibition and Demonstration Chairs

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

## International Liaison, Visa Chair

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

## Publicity Chairs

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

## Women in ICPR Chairs

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

## Event Manager

Alpcord Network

## **Track Chairs – Artificial Intelligence, Machine Learning for Pattern Analysis**

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

## **Track Chairs – Computer and Robot Vision**

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

## **Track Chairs – Image, Speech, Signal and Video Processing**

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

## **Track Chairs – Biometrics and Human Computer Interaction**

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

## Track Chairs – Document Analysis and Recognition

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

## Track Chairs – Biomedical Imaging and Bioinformatics

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

## Metareviewers (Conference Papers and Competition Papers)

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

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

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



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

## **Reviewers (Competition Papers)**

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

## Reviewers (Conference Papers)

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

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

Arnav Bhavsar  
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 Günsel  
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. Martá-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 Coquenot  
Désiré Sidibé  
Devesh Walawalkar  
Dewan Md. Farid  
Di Ming  
Di Qiu  
Di Yuan  
Dian Jia  
Dianmo Sheng  
Diego Thomas  
Diganta Saha  
Dimitri Bulatov  
Dimpy Varshni  
Dingcheng Yang  
Dipanjan Das  
Dipanjoyoti Paul  
Divya Biligere Shivanna  
Divya Saxena  
Divya Sharma  
Dmitrii Matveichev  
Dmitry Minskiy  
Dmitry V. Sorokin  
Dong Zhang  
Donghua Wang  
Donglin Zhang  
Dongming Wu  
Dongqiangzi Ye  
Dongqing Zou  
Dongrui Liu  
Dongyang Zhang  
Dongzhan Zhou  
Douglas Rodrigues  
Duarte Folgado  
Duc Minh Vo  
Duoxuan Pei  
Durai Arun Pannir Selvam  
Durga Bhavani S.  
Eckart Michaelsen  
Elena Goyanes  
Élodie Puybareau

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

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

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

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



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

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

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

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

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

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

Sneha Shukla	Suraj Kumar Pandey
Snehasis Banerjee	Surendrabikram Thapa
Snehasis Mukherjee	Suresh Sundaram
Snigdha Sen	Sushil Bhattacharjee
Sofia Casarin	Susmita Ghosh
Soheila Farokhi	Swakkhar Shatabda
Soma Bandyopadhyay	Syed Ms Islam
Son Minh Nguyen	Syed Tousiful Haque
Son Xuan Ha	Taegyeong Lee
Sonal Kumar	Taihui Li
Sonam Gupta	Takashi Shibata
Sonam Nahar	Takeshi Oishi
Song Ouyang	Talha Ahmad Siddiqui
Sotiris Kotsiantis	Tanguy Gernot
Souhaila Djaffal	Tangwen Qian
Soumen Biswas	Tanima Bhowmik
Soumen Sinha	Tanpia Tasnim
Soumitri Chattopadhyay	Tao Dai
Souvik Sengupta	Tao Hu
Spiros Kostopoulos	Tao Sun
Sreeraj Ramachandran	Taoran Yi
Sreya Banerjee	Tapan Shah
Srikanta Pal	Taveena Lotey
Srinivas Arukonda	Teng Huang
Stephane A. Guinard	Tengqi Ye
Su O. Ruan	Teresa Alarcon
Subhadip Basu	Tetsuji Ogawa
Subhajit Paul	Thanh Phuong Nguyen
Subhankar Ghosh	Thanh Tuan Nguyen
Subhankar Mishra	Thattapon Surasak
Subhankar Roy	Thibault Napol�on
Subhash Chandra Pal	Thierry Bouwmans
Subhayu Ghosh	Thinh Truong Huynh Nguyen
Sudip Das	Thomas De Min
Sudipta Banerjee	Thomas E. K. Zielke
Suhas Pillai	Thomas Swearingen
Sujit Das	Tianatahina Jimmy Francky Randrianasoa
Sukalpa Chanda	Tianheng Cheng
Sukhendu Das	Tianjiao He
Suklav Ghosh	Tianyi Wei
Suman K. Ghosh	Tianyuan Zhang
Suman Samui	Tianyue Zheng
Sumit Mishra	Tiecheng Song
Sungho Suh	Tilottama Goswami
Sunny Gupta	Tim B�chner

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

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

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



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

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

## Contents – Part XVIII

Depth-Enhanced Alignment for Label-Free 3D Semantic Segmentation . . . . .	1
<i>Shangjin Xie, Jiawei Feng, Zibo Chen, Zhixuan Liu, and Wei-Shi Zheng</i>	
Mask-Aware Transformer for Crowd Counting . . . . .	16
<i>Sarah Jad, Marwan Torki, and Ayman Khalafallah</i>	
Unsupervised Real-Time Two-Stage Place Proposal Generation from a Moving Camera Video . . . . .	31
<i>H. İşıl Bozma</i>	
Severity of Flood Damage Estimation from Aerial Scenery . . . . .	46
<i>Tarakeswara Rao Landa and Tushar Sandhan</i>	
Spatio-Temporal Attentive Fusion Unit for Effective Video Prediction . . . . .	62
<i>Binit Singh, Divij Singh, Rohan Kaushal, Sana Vishnu Karthikeya Reddy, Bandam Sai Jaswanth, and Pratik Chattopadhyay</i>	
Few-Shot View Synthesis Based on Geometric and Semantic Consistency . . . . .	80
<i>Mizuki Kojima, Rei Kawakami, and Masatoshi Okutomi</i>	
Fast and Consistently Accurate Perspective-n-Line Pose Estimation . . . . .	97
<i>George Terzakis and Manolis Lourakis</i>	
Wavefront Neural Radiance Fields for Multi-depth Reconstruction . . . . .	113
<i>Tsubasa Nakamura, Ken Sakurada, and Gaku Nakano</i>	
Visibility-Aware Pixelwise View Selection for Multi-View Stereo Matching . . . . .	130
<i>Zhentao Huang, Yukun Shi, and Minglun Gong</i>	
Geometrically Consistent Light Field Synthesis Using Repaint Video Diffusion Model . . . . .	145
<i>Soyoung Yoon and In Kyu Park</i>	
An Empirical Evaluation of the Impact of Solar Correction in NeRFs for Satellite Imagery . . . . .	161
<i>Devjyoti Chakraborty, Kriti Ghosh, Zaki Sukma, In Kee Kim, Lakshmish Ramaswamy, Suchendra M. Bhandarkar, and Deepak R. Mishra</i>	

Skeletal Triangulation for 3D Human Pose Estimation .....	180
<i>YiHeng Jiang, ZhiPeng Wang, YunLong Zhao, Yang Li, and ChunYan Liu</i>	
Reassembling Broken Objects Using Breaking Curves .....	197
<i>Ali Alagrami, Luca Palmieri, Sinem Aslan, Marcello Pelillo, and Sebastiano Vascon</i>	
Fluent and Accurate Image Captioning with a Self-trained Reward Model .....	209
<i>Nicholas Moratelli, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara</i>	
A Benchmark and Chain-of-Thought Prompting Strategy for Large Multimodal Models with Multiple Image Inputs .....	226
<i>Daoan Zhang, Junming Yang, Hanjia Lyu, Zijian Jin, Yuan Yao, Mingkai Chen, and Jiebo Luo</i>	
Improving Multimodal Rumor Detection via Dynamic Graph Modeling .....	242
<i>Xinyu Wu, Xiaoxu Hu, Xugong Qin, Peng Zhang, Gangyan Zeng, Yu Guo, Runbo Zhao, and Xinjian Huang</i>	
Learning to Synthesize Graphics Programs for Geometric Artworks .....	259
<i>Qi Bing, Chaoyi Zhang, and Weidong Cai</i>	
PPCap: A Plug and Play Framework for Efficient Stylized Image Captioning ...	275
<i>Xiangpeng Wei, Yi Li, Guisheng Liu, Yating Liu, and Yanqing Guo</i>	
Harlequin: Color-Driven Generation of Synthetic Data for Referring Expression Comprehension .....	292
<i>Luca Parolari, Elena Izzo, and Lamberto Ballan</i>	
Size-Modulated Deformable Attention in Spatio-Temporal Video Grounding Pipelines .....	308
<i>Hans Tiwari, Selen Pehlivan, and Jorma Laaksonen</i>	
Dual Branch Non-Autoregressive Image Captioning .....	325
<i>Yuanqiu Liu, Hong Yu, Hui Li, Xin Han, and Han Liu</i>	
Distill the Knowledge of Multimodal Large Language Model into Text-to-Image Vehicle Re-identification .....	341
<i>Jianshu Zeng and Chi Zhang</i>	
Audio-Visual Navigation with Anti-Backtracking .....	358
<i>Zhenghao Zhao, Hao Tang, and Yan Yan</i>	

Towards Building Secure UAV Navigation with FHE-Aware Knowledge Distillation .....	373
<i>Arjun Ramesh Kaushik, Charanjit Jutla, and Nalini Ratha</i>	
Zero-Shot Object Navigation with Vision-Language Models Reasoning .....	389
<i>Congcong Wen, Yisiyuan Huang, Hao Huang, Yanjia Huang, Shuaihang Yuan, Yu Hao, Hui Lin, Yu-Shen Liu, and Yi Fang</i>	
Few-Shot Deep Structure-Based Camera Localization with Pose Augmentation .....	405
<i>Cheng-Yu Tsai and Shang-Hong Lai</i>	
Multimodal Point Cloud Completion via Residual Attention Feature Fusion ....	420
<i>Junkang Wan, Hang Wu, and Yubin Miao</i>	
GP-PCS: One-Shot Feature-Preserving Point Cloud Simplification with Gaussian Processes on Riemannian Manifolds .....	436
<i>Stuti Pathak, Thomas Baldwin-McDonald, Seppe Sels, and Rudi Penne</i>	
GSTran: Joint Geometric and Semantic Coherence for Point Cloud Segmentation .....	453
<i>Abiao Li, Chenlei Lv, Guofeng Mei, Yifan Zuo, Jian Zhang, and Yuming Fang</i>	
SPiKE: 3D Human Pose from Point Cloud Sequences .....	470
<i>Irene Ballester, Ondřej Peterka, and Martin Kampel</i>	
<b>Author Index</b> .....	487



# Depth-Enhanced Alignment for Label-Free 3D Semantic Segmentation

Shangjin Xie<sup>1</sup>, Jiawei Feng<sup>1</sup>, Zibo Chen<sup>1</sup>, Zhixuan Liu<sup>1</sup>,  
and Wei-Shi Zheng<sup>1,2,3</sup>(✉)

<sup>1</sup> School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China

{xieshj25, fengjw3, chenzb8, liuzhx8}@mail2.sysu.edu.cn

<sup>2</sup> Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Guangzhou, China  
wszheng@ieee.org

<sup>3</sup> Guangdong Province Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou, China

**Abstract.** Labeling every point in a scene is a laborious journey for 3D understanding. To achieve *annotation-free* training, existing works introduce Contrastive Language-Image Pre-training (CLIP) to transfer the pre-trained capability of visual-linguistic correspondence to 3D-linguistic matching. However, directly adopting this CLIP-driven strategy can inevitably introduce bias: The overrated roles of the color and texture from an RGB image could overshadow the geometric nature of the corresponding 3D scene, resulting in a sub-optimal alignment. We note that different from RGB images, a depth map contains rich geometric information. Inspired by this, we propose Depth-Enhanced Alignment (D-EA) for label-free 3D semantic segmentation. D-EA aims to explore the rich geometric cues in depth maps and mitigate the color and texture biases rooted in the original CLIP-driven strategy. Specifically, we first tune a geometry-enhanced CLIP by aligning its depth prediction to the paired RGB prediction given by the original CLIP. Next, the point cloud feature space is matched with the RGB-Depth aggregated CLIP space by aligning point prediction to RGB and depth predictions. Moreover, to mitigate the semantic ambiguity caused by view-specific noise, we propose a View-Integrated Pseudo Label Generation paradigm. Experiments demonstrate the effectiveness of the proposed D-EA on the ScanNet (indoor) and GraspNet-1Billion (desktop) datasets in the label-free setting. Our method is also competitive in limited annotation semantic segmentation.

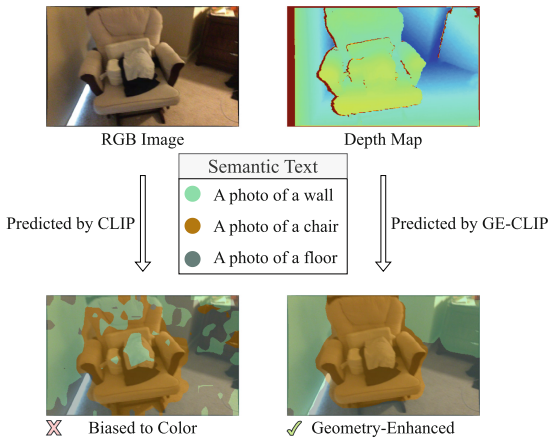
**Keywords:** 3D Semantic Segmentation · Label-Free 3D Semantic Segmentation · RGB-D

## 1 Introduction

3D semantic segmentation is a key cornerstone for numerous real-world applications, *i.e.*, autonomous driving [1, 2], robotics grasping [3, 4], human-robot inter-

action [5]. Thanks to the large amount of high-quality human 3D data annotations, the research on 3D semantic segmentation has achieved an astonishing performance [6–8]. However, the manual annotation of 3D data is quite cumbersome and time-consuming, which hinders their real-world application.

Some remarkable works [9, 10] have introduced Contrastive Language-Image Pre-training (CLIP) for *label-free* 3D semantic segmentation to avoid the tedious data annotation process. CLIP [11] is a two-stream foundation model trained on billions of text-RGB paired data from websites by aligning visual and linguistic feature spaces. The powerful capability to match images with corresponding texts enables CLIP to generalize to downstream 2D vision tasks in a zero-shot manner. Based on the visual-linguistic matching ability of CLIP, existing methods [9, 10] aim at adapting this ability to 3D-linguistic matching. This is achieved by aligning 3D feature space to CLIP visual and linguistic feature spaces<sup>1</sup>. A naive way is to take the most matching text of each pixel in an image as the pseudo label for the corresponding point cloud.

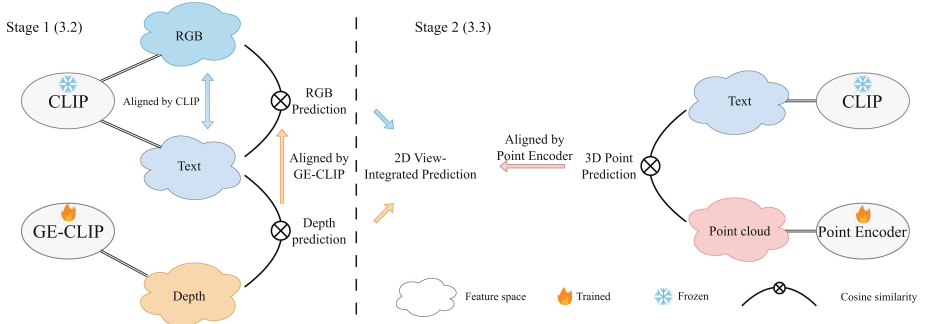


**Fig. 1.** Visualization of prediction with an RGB image (depth map) given by CLIP (GE-CLIP). GE-CLIP is a geometry-enhanced CLIP tuned on depth maps described in Sect. 3.2. While part of the chair is labeled as ‘wall’ probably due to the color and texture bias in RGB prediction, the tuned GE-CLIP can correct this failure.

However, directly adopting this 2D prior knowledge from CLIP may inevitably introduce bias to a 3D understanding model. We claim that the over-rated roles of the color and texture from an RGB image could overshadow the geometric nature of the corresponding 3D scene, resulting in a sub-optimal alignment. As illustrated on the left of Fig. 1, considering an RGB image of an indoor scene, we can obtain the semantic class of each pixel predicted by CLIP. It can

<sup>1</sup> Under the assumption of CLIP, we use ‘text feature space’, ‘linguistic feature space’, and ‘semantic feature space’ interchangeably.

be found that part of the chair is labeled as “wall” probably due to the wall-like color and texture. As a result, the 3D model would tend to ignore geometric cues if it is only supervised by this RGB-biased pseudo label.



**Fig. 2.** Overall methodology of Depth-Enhanced Alignment. **(Stage 1)** We first match the depth feature space to CLIP space by tuning a geometry-enhanced CLIP (GE-CLIP) to align depth prediction with RGB prediction. **(Stage 2)** Then the point cloud feature space is matched with the RGB-Depth aggregated CLIP space by aligning point prediction with RGB and depth predictions. In this work, we define the prediction as the most matching class predicted by the model.

Compared to RGB images, *depth maps* contain richer geometric information of 3D scenes. Besides, there exist some remarkable applications [12–14] that have successfully adapted depth maps to CLIP. To illustrate this, we first tune a geometry-enhanced CLIP image encoder using depth maps (denoted as GE-CLIP in the figure) and then conduct a similar visualization on the right of Fig. 1. Given the fact that CLIP with an RGB image has the bias problem described above, we observe that with its corresponding depth map, the tuned GE-CLIP can correctly predict the wrong part. Motivated by this observation, in this work, we expect to build the depth modality into the CLIP-driven label-free segmentation strategy for transferring semantic knowledge of CLIP *in a geometry-enhanced manner*.

We illustrate the overall idea of this work in Fig. 2. To apply the depth modality to semantic segmentation, we attempt to match the depth prediction to the RGB prediction to achieve an implicit alignment of the depth, RGB, and semantic feature spaces. After the depth feature space is aligned with the CLIP space, we consider aligning the point prediction with the RGB and depth predictions as shown on the right of Fig. 2 for achieving an implicit alignment of the point cloud and RGB-Depth aggregated CLIP space. Moreover, as the same part of a point cloud would appear in multiple (camera) views and correspond to multiple regions of images (depth maps), we propose a View-Integrated Pseudo Label Generation paradigm to reduce the semantic ambiguity caused by view-specific noise.



Experimental results for ScanNet [15] show that our method is effective in indoor scene 3D Semantic Segmentation without any 3D annotation for training. To adapt to robotic scenarios, we also evaluate our method on GraspNet-1Billion [16], a desktop-scenario dataset with RGB images, depth maps, object models, and labeled grasp poses. Our method outperforms the state-of-the-art method. With limited annotations, our method can also achieve great performance. Our contributions are summarized as follows:

- To reduce the color and texture bias and discover more geometric cues, we propose Depth-Enhanced Alignment that matches point cloud feature space to an RGB-Depth aggregated CLIP space for label-free 3D semantic segmentation.
- To mitigate the multi-view semantic ambiguity, we propose a View-Integrated Pseudo Label Generation, which integrates the pseudo labels of multiple pixels for 3D points.
- Our method achieves state-of-the-art performance on both indoor 3D scene dataset ScanNet and desktop-scenario dataset GraspNet-1Billion in label-free tasks. We also achieve great performance in limited annotations semantic segmentation.

## 2 Related Work

### 2.1 CLIP For 2D Zero-Shot Semantic Segmentation

Due to the powerful ability of the pre-trained vision-language model CLIP [11] in zero-shot learning, a series of works have applied CLIP in zero-shot segmentation. Zegformer [17] and zsseg [18] use an extra generator to generate proposals for CLIP to classify, causing inevitable computational cost for each proposal [19]. In contrast, for directly getting pixel-level dense semantic prediction, MaskCLIP [20] firstly extracts patch-level image features from the image encoder of CLIP, and then the text encoder of CLIP is used to construct semantic-aware text features with category names and fixed text prompts which are regarded as a classifier to predict final semantic masks. SAZS [21] indicates the importance of shape-awareness, which utilizes the eigenvectors of Laplacian matrices constructed from self-supervised pixel-wise features to enhance shape-awareness. However, this shape-awareness conducted by some edge detectors on RGB images still lacks the geometric information for 3D tasks. As is well-known, Depth maps naturally contain geometric and disparity information. Thus, the introduction of Depth maps in our method could discover meaningful geometric cues more accurately for label-free 3D semantic segmentation.

### 2.2 3D Semantic Segmentation

The purpose of 3D semantic segmentation is to divide 3D data into different parts corresponding to different semantic labels. Recent works utilize point clouds [22–24] or voxels [25–27] as inputs to train the segmentation models with detailed

semantic annotation. However, enormous amounts of semantic annotation are quite cumbersome and time-consuming for 3D data. Inspired by the superior ability of CLIP in 2D zero-shot and open-vocabulary tasks, CLIP2Scene [9] firstly leverages pre-trained knowledge of CLIP to 3D scene understanding and proposes a novel semantic-driven framework for distilling 2D image knowledge. CLIP-FO3D [10] designs a superpixel-based method to extract dense pixel-level features from CLIP and trains the 3D scene understanding model with feature distillation. Although these methods have shown promising performance in label-free 3D semantic segmentation, this direct CLIP-driven transfer strategy neglects the geometric information of 3D data and overestimates the importance of color and texture in RGB images. Thus, we propose introducing depth maps to enhance the original CLIP knowledge geometrically and transferring semantic knowledge containing the geometric and color information to the 3D model for label-free semantic segmentation.

### 3 Depth-Enhanced Label-Free 3D Segmentation

In this section, we introduce the proposed Depth-Enhanced Alignment framework for exploring the rich geometric cues in depth maps and reducing the color and texture biases from the original CLIP-driven label-free 3D segmentation strategy [9, 10].

#### 3.1 Preliminary

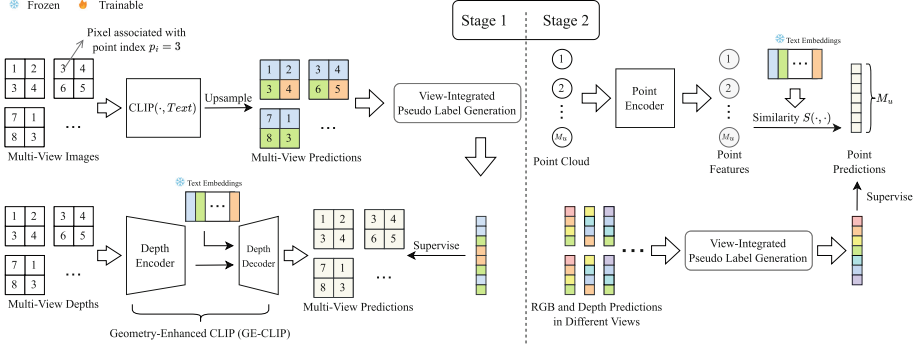
We focus on training a 3D model to segment the 3D data semantically without any human annotations. For each scene, we can get the point clouds of the 3D scene  $\mathcal{P} \in \mathbb{R}^{N \times 6}$  with  $N$  points represented by 3D coordinates and RGB colors, where  $N$  is the number of points in the point cloud of a scene. A sequence of RGB images  $\mathcal{I} \in \mathbb{R}^{L \times H \times W \times 3}$  and depth maps  $\mathcal{D} \in \mathbb{R}^{L \times H \times W}$  are captured by RGB-D camera in different camera poses  $\mathcal{T} \in \mathbb{R}^{L \times 4 \times 4}$ , where  $L$  is the size of image sequences,  $H$  is the image height, and  $W$  is the image width. We notice that an object in a scene can be captured as images from multiple views. For ease of introducing our method, we define pixel-point corresponding pair as follows. Specifically, with the camera pose  $\mathcal{T}$  and camera intrinsics  $\mathcal{K}$ , we transform point clouds of the 3D scene from world coordinate to pixel coordinate and build the corresponding pairs of pixel-point indexes,

$$O_{multiple} = \{x_i, p_i\}_{i=1}^M, \quad (1)$$

where  $M$  is the number of the corresponding pairs,  $x_i \in \{1, 2, \dots, LHW\}$  and  $p_i \in \{1, 2, \dots, N\}$  are the pixel index in the scene images and point index in the scene point cloud of  $i$ -th corresponding pair. A point can be associated with multiple pixels from different views, i.e., a 3D point  $p_i$  may appear in more than one pair in  $O_{multiple}$ . To get a set of unique point indexes, we denote a pixel index set for each point to represent this many-to-one mapping  $\{p_k\}_{k=1}^{M_u}$ ,  $M_u$

is the number of unique point indexes. For each unique point index  $p_k$ , we can define the associated pixel indexes set  $C_{p_k} = \{x_j | p_j = p_k\}_{j=1}^M$ , thus the  $O_{multiple}$  can be transformed to

$$O_{unique} = \{C_{p_i}, p_i\}_{i=1}^{M_u}. \quad (2)$$



**Fig. 3.** Depth-Enhanced Alignment is a two-stage training method. **(Stage 1)** We forward the CLIP and use text embedding to obtain the pixel-wise RGB multi-view predictions. The View-Integrated Pseudo Label Generation will generate the RGB View-Integrated pseudo labels. We then adopt the above view-integrated pseudo labels to train the Geometry-Enhanced CLIP. **(Stage 2)** The point encoder is trained by RGB and depth predictions in different views. We also integrate the predictions of pixels associated with the same points to generate the view-integrated pseudo labels. Given a point with index  $p_i = 3$ , it can correspond to multiple pixels in multiple images. Different colors represent different prediction classes.  $S(\cdot, \cdot)$  is the cosine similarity.  $M_u$  is the number of unique point indexes.

Our 2D feature extractor is CLIP, following [20], we use the representation of the final attention layer for segmentation. Firstly, We use pre-defined templates to describe the target class names, such as “a photo of a [table]”, and forward the text encoder to generate class-aware text embeddings  $t \in \mathbb{R}^{K \times d}$  as the classifier, where  $K$  is the number of categories in the dataset and  $d$  is the feature dimension. Then we will encode image features with the visual encoder and calculate the cosine similarity between the text embeddings and image features as the pixel-level class logits.

### 3.2 Align Depth Feature Space to CLIP

Existing methods [9, 10] directly distill 2D knowledge of CLIP to 3D modality. However, this manner could make the 3D model tend to ignore geometric cues because it is only supervised by the RGB-biased knowledge as we illustrated in Sect. 1. In this work, we make an attempt to introduce depth maps to mine the geometric knowledge from the pre-trained CLIP.

Due to the huge modality gap between the depth maps and RGB images, directly feeding depth maps into the CLIP encoder could fail to extract meaningful representations that are aligned to CLIP semantic space when dealing with label-free 3D semantic segmentation. To achieve this, we design a Geometry-Enhanced CLIP (GE-CLIP) model as shown in Stage 1 of Fig. 3. With the depth map as the input modality, we attempt to align the depth feature to the text-RGB feature space. Specifically, we first denote  $f_{rgb}$  as the clip image encoder, and the encoder of GE-CLIP can be denoted as  $f_{dep}$ . Motivated by the observation of recent work [20], we combine the  $f_{dep}$  with the lightweight transformer decoder of ZegCLIP [19] as the decoder  $g_{dep}$ . The text embedding  $t$  was integrated into the  $f_{dep}$  to calculate pixel-level semantic logits. For tuning the GE-CLIP,  $f_{rgb}$  and  $f_{dep}$  will be loaded with the pre-trained CLIP weights, and depth decoder  $g_{dep}$  is randomly initialized.

With the pre-trained text-RGB pairing knowledge of the CLIP, we can tune the GE-CLIP to align with the text-RGB feature space. The input of  $f_{rgb}$  is RGB images and the input of  $f_{dep}$  is depth maps. To adapt the image encoder of CLIP, we extend the depth map to three identical channels  $\mathcal{D} \in \mathbb{R}^{L \times H \times W \times 3}$ . Then we have  $h^{rgb} = f_{rgb}(\mathcal{I})$  and  $h^{dep} = f_{dep}(\mathcal{D})$ , where  $h^{rgb}$  and  $h^{dep} \in \mathbb{R}^{L \times H \times W \times d}$  are the encoder features of  $f_{rgb}$  and  $f_{dep}$ . After that, we calculate the cosine similarity between the RGB features and text embeddings  $t \in \mathbb{R}^{K \times d}$  to get dense pixel-level predictions over all categories and choose the maximum probability class as the pixel pseudo labels. It can be formulated as:

$$\hat{y}_{rgb} = \arg \max(\mathcal{S}(h^{rgb}, t)), \quad (3)$$

where  $\hat{y}_{rgb} \in \mathbb{R}^{L \times H \times W}$  is the pseudo labels from CLIP,  $\mathcal{S}(\cdot, \cdot)$  is the cosine similarity and  $t$  is the text embeddings. Then, we use the RGB pseudo labels to supervise the GE-CLIP. The objective function is as follows:

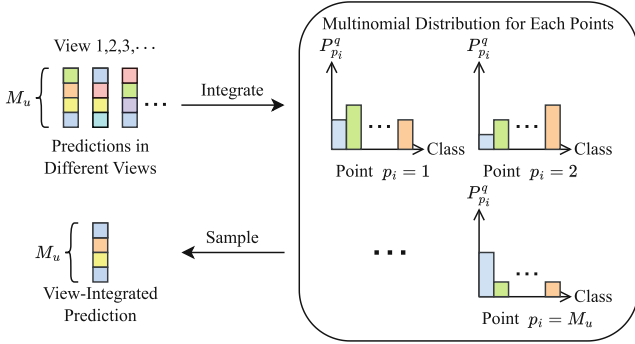
$$\mathcal{L}_{\mathcal{I} \rightarrow \mathcal{D}} = CE(g_{dep}(h^{dep}, t), \hat{y}_{rgb}), \quad (4)$$

where  $g_{dep}(h^{dep}, t) \in \mathbb{R}^{L \times H \times W \times K}$  is the depth pixel-level logits from depth decoder.  $CE(\cdot, \cdot)$  is the Cross-Entropy loss function. After that, the GE-CLIP acquires the geometry-enhanced CLIP knowledge from the depth maps and pre-trained CLIP text-RGB knowledge. Though supervised by RGB-biased labels, GE-CLIP achieves an implicit geometric constraint by using depth modality as the input. In the next section, we will consider how to aggregate the knowledge from both CLIP and GE-CLIP to align the 3D model.

### 3.3 Align Point Feature Space to RGB-Depth Aggregated CLIP

Instead of directly distilling 2D knowledge of CLIP to 3D modality, we introduce the depth modality and propose D-EA that aggregates the knowledge from both CLIP and GE-CLIP to align the 3D model with the color information and geometry information.

**Depth-Enhanced Alignment.** We denote the 3D model as  $f_{3D}$ , and point clouds of the 3D scene are passed into the 3D model. We can get  $h^{3D} = f_{3D}(\mathcal{P})$ ,



**Fig. 4.** View-Integrated Pseudo Label Generation. With the predictions in different views, we integrate the predictions of multiple pixels that are associated with the same 3D point. Then, we regard the pseudo-label frequency of multiple pixels as the probability of the class multinomial distribution to sample the final class as the view-integrated pseudo label which will be used to supervise each 3D point.  $P$  is the view-integrated probability of the class multinomial distribution.

where  $h^{3D} \in \mathbb{R}^{N \times d}$  are the point features. With the pixel-point corresponding pairs  $\mathcal{O}_{multiple}$  introduced in Sect. 3.1, we get the RGB-3D pairs pseudo labels with CLIP and depth-3D pairs pseudo labels with GE-CLIP, this can be formalized as  $\hat{y}^q \in \mathbb{R}^M$ , where  $q \in [\mathcal{I}, \mathcal{D}]$  for RGB (depth) and  $M$  is the number of pixel-point corresponding pairs in  $\mathcal{O}_{multiple}$ .

Finally, the 3D network will be supervised by RGB pseudo labels and depth pseudo labels with Cross-entropy Loss, respectively. The objective function is as follows:

$$\mathcal{L}_{\mathcal{I}, \mathcal{D} \rightarrow \mathcal{P}} = \frac{1}{M} \sum_{i=1}^M CE(\mathcal{S}(h^{3D}, t)[p_i], \hat{y}^q[x_i]), \quad (5)$$

where  $q \in [\mathcal{I}, \mathcal{D}]$ ,  $\hat{y}^q[x_i]$  are the pseudo labels of  $x_i$ -th RGB/depth pixel and  $\mathcal{S}(h^{3D}, t)[p_i]$  are the logits of  $p_i$ -th point.

**View-Integrated Pseudo Label Generation.** We mentioned in Sect. 3.1 that with the camera moving in the 3D scene to capture RGB images and depth maps, one object would be captured in multiple images. Thus, in the pixel-point corresponding pairs  $\mathcal{O}_{multiple}$ , one point can be associated with multiple pixels from different views. A 3D object viewed from a bright perspective can be correctly recognized by CLIP but may have different predictions from a dark view. Due to this view-specific noise, these multiple pixels associated with the same point may get multiple pseudo labels from the 2D model. And Eq. (5) will confuse the 3D model for aligning text feature space. In light of this, we propose a View-Integrated Pseudo Label Generation to alleviate this issue.

Specifically, we first adapt the  $\mathcal{O}_{multiple}$  to the unique point pixel-point corresponding pairs  $\mathcal{O}_{unique}$ . For each point index  $p_k$  in  $\mathcal{O}_{unique}$ , we convert the associated predictions of RGB/depth pixels to one-hot form. Then we sum the

one-hot of pixels to indicate the target pseudo label frequency  $F_{p_i}^q$  of  $p_i$ -th point which appears in different views.

Based on the Law of Large Numbers, we regard the frequency as the probability of the class multinomial distribution  $P_{p_i}^q$ . Then we sample the final class index  $\tilde{y}_{p_i}^q$  as the pseudo label for  $p_i$ -th point with the multinomial distribution. Formally, this can be written as:

$$\tilde{y}_{p_i}^q \sim P_{p_i}^q. \quad (6)$$

**Table 1.** Compared with recent state-of-the-art Label-free Semantic Segmentation on ScanNet and GraspNet-1Billion datasets. D-EA (Depth-Enhanced Alignment) denotes the method we proposed. We did not reproduce the result of CLIP-FO3D because it has not been open-sourced.

Method	ScanNet		GraspNet-1Billion	
	mIoU(%)	mAcc(%)	mIoU(%)	mAcc(%)
CLIP2Scene [9]	28.1	46.6	21.8	33.9
CLIP-FO3D [10]	30.2	49.1	-	-
D-EA(ours)	<b>35.1</b>	<b>52.5</b>	<b>22.8</b>	<b>36.0</b>

To align points feature space to RGB-Depth aggregated CLIP, we utilize the aggregate pseudo labels from the CLIP and GE-CLIP in a view-integrated way:

$$\tilde{y}_{p_i}^{\mathcal{I},\mathcal{D}} \sim (P_{p_i}^{q=\mathcal{I}} + P_{p_i}^{q=\mathcal{D}})/2, \quad (7)$$

As shown in Fig. 4, the 3D backbone will be optimized by the aggregate pseudo labels finally:

$$\mathcal{L}_{\mathcal{I},\mathcal{D} \rightarrow \mathcal{P}}^{unique} = \frac{1}{M_u} \sum_{i=1}^{M_u} CE(\mathcal{S}(h^{3D}[p_i], t), \tilde{y}_{p_i}^{\mathcal{I},\mathcal{D}}). \quad (8)$$

**Remark.** During the GE-CLIP tuning in Stage 1, we can also leverage the View-Integrated Pseudo Label Generation, as shown in Stage 2 of Fig. 3. Equation (4) can be transformed to:

$$\mathcal{L}_{\mathcal{I} \rightarrow \mathcal{D}}^{unique} = \frac{1}{M_u} \sum_{i=1}^{M_u} \sum_{x_j \in C_{p_i}} CE(g_{dep}(h^{dep}, t)[x_j], \tilde{y}_{p_i}^{\mathcal{I}}), \quad (9)$$

where  $\tilde{y}_{p_i}^{\mathcal{I}} \sim P_{p_i}^{\mathcal{I}}$  is the unique RGB pseudo label from the CLIP.

## 4 Experiments

To evaluate the effectiveness of the proposed Depth-Enhanced Alignment framework, we conducted experiments on the indoor 3D scene dataset ScanNet and the robotics grasping dataset GraspNet-1Billion.

**Table 2.** The results(mIoU) of limited annotations semantic segmentation on ScanNet. The ‘‘Sup’’ denotes the supervision of points semantic labels during pre-training. The **bold** indicates the best result and underline denotes the second-best result. \*Compared to the remaining methods including ours, VIBUS additionally utilized a fine-grained pseudo-label fine-tuning stage.

Method	Sup	Number of labeled points			
		20	50	100	200
Scratch	-	46.3	58.3	62.8	65.4
CSC [28]	✗	54	60.7	65.6	68.3
LangGround [29]	✓	55.1	62.4	66	68.2
VIB [30]	✗	57	63.6	66.8	68.5
VIBUS* [30]	✗	<b>61.6</b>	<b>65.6</b>	<b>68.9</b>	<b>69.6</b>
CLIP-FO3D [10]	✗	57.6	64.3	68.2	<u>69.5</u>
D-EA init(ours)	✗	<u>57.7</u>	<u>64.5</u>	<u>68.7</u>	69.4

## 4.1 Experimental Setup

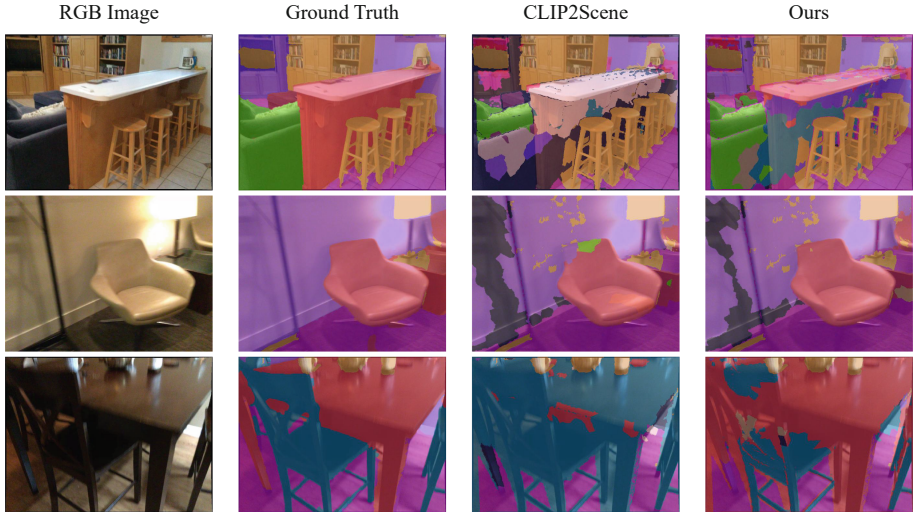
**Datasets.** ScanNet [15] dataset is a large-scale indoor dataset, which contains 1201 scans for training and 312 scans for validation, totaling 20 classes. Besides, we also perform experiments on the challenging robotics grasping dataset GraspNet-1Billion [16], from which 128 and 50 scenes are selected for training and testing.

**Implementation Details.** We applied the ViT-based [31] CLIP as the image encoder of our GE-CLIP. The text encoder was passed with the text prompts of categories to generate text embeddings. We used the 80 hand-craft prompts same with MaskCLIP. Moreover, we also adopted the decoder of ZegCLIP [19] as the decoder of GE-CLIP which can directly output the pixel-level class logits. For 3D data, we used the MinkowskiEngine [26] to build a point cloud sparse convolution model ResUNet14 [12] as the 3D backbone. When tuning the GE-CLIP, the RGB encoder was frozen. When training the 3D backbone, both the CLIP and GE-CLIP were frozen. The optimizer was AdamW with a cosine scheduler. Following the CLIP2Scene [9], we also applied some data augmentations, including random rotation around the z-axis and random flip on the point cloud, random horizontal flip, and random crop on the images.

## 4.2 Label-Free 3D Semantic Segmentation

We evaluated our method on the ScanNet and GraspNet-1Billion datasets for 3D label-free semantic segmentation.

**Datasets Setup.** For the ScanNet dataset, we conducted experiments on standard benchmarks. The 3D network was trained on the training set without any labels and evaluated on the validation set. Following CLIP-FO3D [10], we



**Fig. 5.** Visualization of label-free 3D semantic segmentation on ScanNet dataset. We used the intrinsic and extrinsic parameters of the camera to project the 3D model predictions into 2D views.

removed the “other furniture” category that has no specific semantics for correcting the text embeddings. To adapt our method to desktop robotic grasping scenes, we also evaluated our method on the GraspNet-1Billion dataset. Similar to the ScanNet, some objects have no specific semantic categories. Thus, we removed some semantic-unclear categories, such as “mount1” and “part1”. We also aggregated some sub-categories into a single one, for example, integrating the “cracker box” and “sugar box” into the “box”. We used the metric of mean Intersection over Union (mIoU) and mean Average class accuracy (mAcc) to evaluate our method on the 3D label-free semantic segmentation.

**Results.** The performance of our method is shown in Table 1. To ensure a fair comparison, we selected the following methods which did not introduce large models that used pixel-level labels for pre-training as auxiliary information. Specifically, we compared our method with CLIP2Scene and CLIP-FO3D on the ScanNet and compared different setups of our methods on the GraspNet-1Billion. **D-EA** is our method that uses View-Integrated Pseudo Labels Generation (Sect. 3.2) to tune depth network and aggregate the view-integrated pseudo labels of RGB and depth to train 3D network (Sect. 3.3), optimized by (9) and (8). We reproduced the result of CLIP2Scene on scanNet and GraspNet-1Billion.

Our method outperformed the previous state-of-the-art method by 7%, 4.9% mIoU on the ScanNet, and 1.0% mIoU on GraspNet-1Billion. It shows that our method of **D-EA** that introduces the depth map can enhance the CLIP to extract more reasonable semantic features for label-free semantic segmentation. The improvements of our method was insignificant compared to experiments in



the ScanNet possibly due to the size of the objects was much smaller than the objects in the ScanNet dataset. The depth map of GraspNet-1Billion can be easily affected by the distortion from the depth camera. However, our method can also have an improvement over CLIP2Scene.

**Visualization.** The Visualization of label-free 3D semantic segmentation on ScanNet dataset is shown in Fig. 5. For a more straightforward comparison, we used the camera intrinsic and extrinsic parameters to project the predictions of 3D models into 2D views. As shown in Fig. 5, CLIP2Scene has the color and texture bias inherited from image-based CLIP and can not correctly segment the chairs and tables that have similar colors. In comparison, our method utilized rich geometric information from depth maps and achieved a better performance.

**Table 3.** The individual improvements of our method performed on ScanNet dataset with Label-free Semantic Segmentation setting. “View-Int” represents the View-Integrated Pseudo Label Generation paradigm.

Exp	Components				mIoU(%)
	Depth	View-Int	RGB-Frozen	RGB-Tuned	
1			✓		31.2
2				✓	29.6
3	✓		✓		33.6
4	✓	✓	✓		<b>35.1</b>

### 4.3 Limited Annotation Semantic Segmentation

Recently, some methods [32, 33] attempt to learn 3D semantic segmentation with a subset of labels. In this section, we also evaluated the proposed D-EA in Limited Annotations (only a few labels were used for training) on the ScanNet, which randomly annotated 20, 50, 100, and 200 points of each scene for training. Following [28], the baseline of this setting is training with a classification loss using the labels mentioned above. We denoted this baseline as “Scratch” in Table 2. To evaluate the effectiveness of the proposed D-EA, we simply used the model parameters trained by D-EA as initialization and tuned them by the same classification loss. We denoted this method as “D-EA init” in Table 2.

The results are shown in Table 2, our method achieves the mIoU improvement of 11.4%, 6.2%, 5.9%, and 4.0% over the training from Scratch. We outperformed the CLIP-FO3D that used CLIP for pre-training at the 20, 50, and 100 annotated points settings. With the increase of annotated points, the improvement of “D-EA init” compared to “Scratch” became minor as the ground-truth supervision overshadowed the prior knowledge from D-EA, thus smoothing out the gap between “D-EA init” and “Scratch”. Note that VIB [30] is the method for the pre-training stage. Compared to the remaining methods including ours, VIBUS

added a fine-grained pseudo-label fine-tuning stage. Our method, similar to VIB, focuses on the pre-training stage, which is orthogonal to VIBUS and can be integrated together in different training stages. Compared to VIB, our pre-training method still achieves better performance.

#### 4.4 Ablation Study

In this section, we tested the effectiveness of different components in D-EA. All the experiments were conducted on the ScanNet validation set, under the 3D label-free semantic segmentation setting. The results are shown in Table 3 and described as follows.

**Effectiveness of Introducing Depth Maps.** Exp 1 is the baseline method that we used the frozen CLIP to generate RGB pseudo labels  $\hat{y}_{rgb}$  in (3) to optimize the 3D backbone. In Exp 3, the GE-CLIP is tuned by  $\mathcal{L}_{\mathcal{I} \rightarrow \mathcal{D}}$ . The 3D backbone will be supervised both by  $\mathcal{L}_{\mathcal{I}, \mathcal{D} \rightarrow \mathcal{P}}$ . Comparing Exp 3 with Exp 1, the method using depth maps to enhance the pre-trained text-RGB pairing knowledge of CLIP in a geometrical manner improves the results by 2.4% mIoU.

To further validate the effectiveness of introducing depth maps rather than fine-tuning CLIP, we design the Exp 2. Specifically, we first construct a CLIP with the same encoder-decoder architecture as the GE-CLIP. Then we use the RGB pseudo labels  $\hat{y}_{rgb}$  to tune it. Finally, the RGB pseudo labels from the tuned CLIP will be used to optimize the 3D network. Comparing Exp 2 and Exp 4, fine-tuning depth is more effective than fine-tuning RGB. More importantly, comparing Exp 2 to Exp 1, it is shown that using the pseudo label of RGB to tune an RGB-input CLIP would make the color and texture biases further exacerbated, which could lead to a degradation of performance.

**Effectiveness of View-Integrated Pseudo Label Generation.** We compared Exp 4 with Exp 3. Exp 4 is our proposed method D-EA. In Exp 4, we added the View-Integrated Pseudo Label Generation (introduced in Sect. 3.3) to the framework in Exp 3. Specifically, we used  $\mathcal{L}_{\mathcal{I} \rightarrow \mathcal{D}}^{unique}$  to tune the depth network and use  $\mathcal{L}_{\mathcal{I}, \mathcal{D} \rightarrow \mathcal{P}}^{unique}$  to train the 3D backbone. Exp 4 improved the results by 1.5% mIoU. The result shows that View-Integrated Pseudo Label Generation utilizes the multiple pixels pseudo label and is effective for easing the view-specific noise.

## 5 Conclusions

In this work, we have proposed a novel Depth-Enhanced Alignment (D-EA) method to build depth modality into the CLIP-driven strategy geometrically for label-free 3D semantic segmentation, where we are the first to introduce the depth maps to enhance the original CLIP with geometric information. We further design a View-Integrated Pseudo Label Generation to mitigate the semantic ambiguity caused by view-specific noise. Our method achieves state-of-the-art performance on both the 3D indoor scene dataset ScanNet and the robotic grasping dataset Graspnet-1Billion in label-free tasks. Our model also achieves great performance in limited annotations semantic segmentation.

**Acknowledgements.** This work was supported partially by the Guangdong NSF Project (No. 2023B1515040025).

## References

1. Xu, J., Zhang, R., Dou, J., Zhu, Y., Sun, J., Pu, S.: Rpvnet: a deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16004–16013 (2021)
2. Ando, A., Gidaris, S., Bursuc, A., Puy, G., Boulch, A., Marlet, R.: Rangevit: towards vision transformers for 3D semantic segmentation in autonomous driving. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5240–5250 (2023)
3. Ückermann, A., Haschke, R., Ritter, H.: Real-time 3D segmentation of cluttered scenes for robot grasping. In: 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), pp. 198–203. IEEE (2012)
4. Ückermann, A., Elbrechter, C., Haschke, R., Ritter, H.: 3D scene segmentation for autonomous robot grasping. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1734–1740. IEEE (2012)
5. Ückermann, A., Haschke, R., Ritter, H.: Realtime 3D segmentation for human-robot interaction. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2136–2143. IEEE (2013)
6. Guo, Y., Wang, H., Hu, Q., Liu, H., Bennamoun, M.: Deep learning for 3D point clouds: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020)
7. Yang, Y.-Q., et al.: Swin3d: a pretrained transformer backbone for 3D indoor scene understanding. *arXiv*, vol. abs/2304.06906 (2023)
8. Engel, N., Belagiannis, V., Dietmayer, K.C.J.: Point transformer. *IEEE Access* **9**, 134826–134840 (2020)
9. Chen, R., et al.: Clip2scene: towards label-efficient 3D scene understanding by clip. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7020–7030 (2023)
10. Zhang, J., Dong, R., Ma, K.: Clip-fo3d: learning free open-world 3D scene representations from 2D dense clip. *arXiv*, vol. abs/2303.04748 (2023)
11. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
12. Zhang, Z., Liu, Q., Wang, Y.: Road extraction by deep residual u-net. *IEEE Geosci. Remote Sens. Lett.* **15**, 749–753 (2017)
13. Hu, X., Zhang, C., Zhang, Y., Hai, B., Yu, K., He, Z.: Learning to adapt clip for few-shot monocular depth estimation. *arXiv*, vol. abs/2311.01034 (2023)
14. Zhang, R., et al.: Pointclip: point cloud understanding by clip. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8542–8552 (2021)
15. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: richly-annotated 3D reconstructions of indoor scenes. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2432–2443 (2017)
16. Fang, H., Wang, C., Gou, M., Lu, C.: Graspnet-1billion: a large-scale benchmark for general object grasping. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11441–11450 (2020)

17. Ding, J., Xue, N., Xia, G., Dai, D.: Decoupling zero-shot semantic segmentation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11573–11582 (2021)
18. Xu, M., et al.: A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. arXiv, vol. abs/2112.14757 (2021)
19. Zhou, Z., Lei, Y., Zhang, B., Liu, L., Liu, Y.: Zegclip: towards adapting clip for zero-shot semantic segmentation. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11175–11185 (2022)
20. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: European Conference on Computer Vision (2021)
21. Liu, X., et al.: Delving into shape-aware zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2999–3009 (2023)
22. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.-W., Jia, J.: Pointgroup: dual-set point grouping for 3D instance segmentation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4866–4875 (2020)
23. Qi, C. R., Chen, X., Litany, O., Guibas, L.J.: Imvotenet: boosting 3D object detection in point clouds with image votes. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4403–4412 (2020)
24. Thomas, H., Qi, C., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: flexible and deformable convolution for point clouds. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6410–6419 (2019)
25. Maturana, D., Scherer, S.A.: Voxnet: a 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922–928 (2015)
26. Choy, C.B., Gwak, J., Savarese, S.: 4D spatio-temporal convnets: minkowski convolutional neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3070–3079 (2019)
27. Zhou, Y., Tuzel, O.: Voxnet: end-to-end learning for point cloud based 3D object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4490–4499 (2017)
28. Hou, J., Graham, B., Nießner, M., Xie, S.: Exploring data-efficient 3D scene understanding with contrastive scene contexts. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15582–15592 (2020)
29. Rozenberszki, D., Litany, O., Dai, A.: Language-grounded indoor 3D semantic segmentation in the wild, arXiv, vol. abs/2204.07761 (2022)
30. Tian, B., Luo, L., Zhao, H., Zhou, G.: Vibus: data-efficient 3D scene parsing with viewpoint bottleneck and uncertainty-spectrum modeling. *ISPRS J. Photogramm. Remote. Sens.* **194**, 302–318 (2022)
31. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv, vol. abs/2010.11929 (2020)
32. Liu, Z., Qi, X., Fu, C.-W.: One thing one click: a self-training approach for weakly supervised 3D semantic segmentation. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1726–1736 (2021)
33. Wei, J., Lin, G., Yap, K.-H., Hung, T.-Y., Xie, L.: Multi-path region mining for weakly supervised 3D semantic segmentation on point clouds. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4383–4392 (2020)



# Mask-Aware Transformer for Crowd Counting

Sarah Jad<sup>(✉)</sup>, Marwan Torki, and Ayman Khalafallah

Computer and Systems Engineering Department, Alexandria University, Alexandria, Egypt

{es-sarah.jad1217,mtorki,ayman.khalafallah}@alexu.edu.eg

**Abstract.** Crowd counting problem is a challenging task in computer vision and image analysis. It has many applications in the real world such as crowd management, public safety, and urban planning. Our proposal in this paper is a mask-aware transformer-based network for crowd counting that uses the background/foreground mask information to improve density regression accuracy. Our backbone network is a pyramid vision transformer. Our proposed Mask-aware transformer (M-Trans) takes into consideration the background/foreground mask information. We further improve the performance by applying a greedy ensemble strategy. Our experimental evaluation shows that our mask-aware network achieves state-of-the-art performance on standard benchmarking datasets for crowd counting such as ShanghaiTech and UCF-QNRF datasets.

**Keywords:** Crowd counting · Vision Transformer · Mask-aware model · Density map

## 1 Introduction

*Crowd counting* aims to count the number of people within an image or a video frame. It has many applications in the real world such as crowd management, public safety, and urban planning. Crowd counting is a challenging problem in computer vision and image analysis. It is challenging because of the large variations in crowd density, scale, and perspective.

There are different approaches to addressing the crowd-counting problem. Some methods treat the problem as a regression task. Other methods treat the problem as a detection problem. However, in the literature, the annotations for the crowd-counting datasets are done on the dot level per object. Because extremely complex annotations, like object bounding boxes or instance-level segmentation masks, are difficult to provide, dot-level annotations are recommended during the dataset construction process. Because of these difficulties, the traditional detection-based approach is less accurate. To solve this problem, most existing methods choose to estimate a density map from the dot-level annotation. In Fig. 1 we show an example of an image with its estimated density map. By simply integrating the density values in the map, an object count can be calculated once the density map has been accurately estimated. Integrating

an image mask into the density map regression method improves the output’s accuracy in counting.

In this paper, we propose a mask-aware transformer model (M-Trans) that integrates the background/foreground mask information into the transformer-based model. We process the input image and an additional mask to extract mask features as well as other features driven from a pyramid vision transformer-based model. Both extracted features are then combined to feed a density map estimation head. The utilization of the mask-extracted features proved to be effective in creating more accurate density maps as we show in our experiments. Next, we count the number of persons present in the scene using these maps. The main contributions of our work can be summarized as follows:



**Fig. 1.** Original image of a highly crowded scene and corresponding crowd density map.

1. We propose a novel mask-aware transformer-based architecture that incorporates the benefits of foreground/background masks in the training.
2. We improve the performance by applying a greedy ensembling strategy.
3. We compare the performance of our proposed model to several baselines on the standard crowd-counting benchmarks with remarkable improvement over many baselines.

The rest of the paper is organized as follows. We present the related work on the crowd-counting problem in Sect. 2. Next, we describe our novel methodology in Sect. 3. In Sect. 4, we show our experimental evaluation of our novel architecture on standard benchmarks for crowd counting. Lastly, in Section we show the conclusions of our study.

## 2 Related Work

*Several significant contributions* are present in the computer vision research community for the crowd-counting problem. Regression-based methods are a common approach such as [3, 4]. In Regression-based methods, a CNN is trained to directly predict the count of people in an image using regression techniques.

This method doesn't rely on density maps but instead learns to calculate the count directly from the provided image.

Adaptive approaches [5] dynamically adjust the network architecture or hyperparameters based on the characteristics of the input image. For example, switching mechanisms may choose different pathways in the network based on crowd density like Switch-CNN [6], allowing for better adaptability to varying conditions. Designing effective mechanisms for dynamically adjusting network architecture can be complex and may not always lead to improved performance.

Detection-based approaches treat crowd counting as a detection problem, where individual people in the crowd are detected, localized, and counted. This typically involves using object detection models such as Faster R-CNN [7] or YOLO [8] to identify people within the image. The count is then determined based on the detected objects.

Contextual Information approaches incorporate contextual information, such as scene context or the relationships between people, to improve crowd counting accuracy. Context-aware models consider not only individual people but also their interactions within the crowd. CAN module proposed in [9] is built using this approach. Incorporating contextual information can add complexity to the model and may require additional computational resources.

The density Map Regression approach involves training a convolution neural network (CNN) to directly predict a density map from an input image (see Fig. 1). The density map assigns a density value to each pixel in the image. These density values are added up to get the total number of people in the scene. This method is effective for handling varying crowd densities within an image. Requires fine-grained annotations in the form of density maps, which can be labor-intensive and costly to create. Multi-column CNN approach which is introduced in MCNN [10] uses multiple parallel columns within the network to capture different aspects and scales of information in the crowd image. These columns may be specialized for different crowd densities or conditions. The final count is obtained by aggregating the outputs of these columns. The disadvantage of MCNN is increasing computational complexity due to multiple columns, making it more resource-intensive.

The application of Vision transformers (ViTs) to the crowd-counting problem was not widely explored. The CCTrans module proposed in [11] uses a pyramid transformer as a module backbone to extract global features from the input image. Another approach to enhance crowd counting problem results is to integrate the background/foreground mask into the network for more accurate density regression. Mask-aware networks for crowd counting [12] proposed this approach, they introduce mask integration into network architecture somehow like to Top-down feedback for crowd-counting convolutional neural network [13] which is a multi-layered CNN.

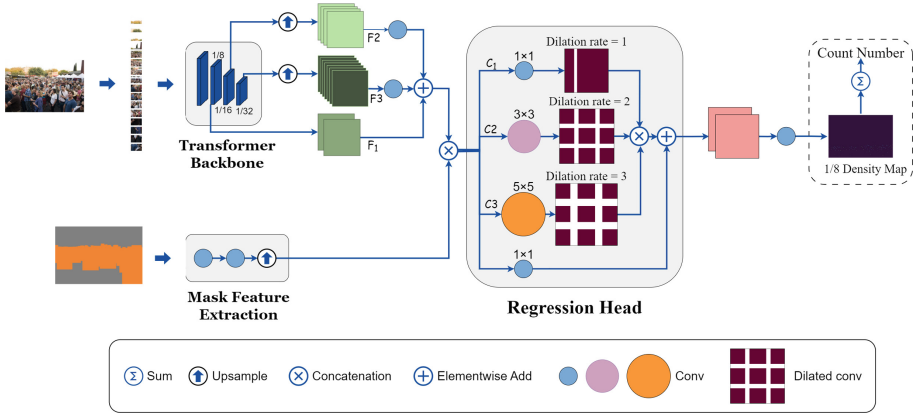
Our analysis of the related work revealed that there is a huge advantage to using transformer-based models. However, mask integration is still under investigation. Thus, we decided to bridge the gap by bringing the idea of mask integration into the successful transformer-based models for crowd counting.

### 3 Mask-Aware Transformer (M-Trans)

This section starts with an illustration of our proposed model’s overall architecture, followed by a detailed introduction to each component.

#### 3.1 M-Trans Overview

Our proposed Model integrates information of an image background/foreground mask to the CCTrans model to get better crowd counting results. As we can see in Fig. 2, we use the input image along with its foreground/background mask. The input image will go through a feature extraction branch using a CCTrans module. The mask will go through feature extraction via convolution. The features are concatenated before a regression head and then a density estimation head will produce the final density map. In the following, we start by describing our backbone model (CCTrans) and then describe our mask feature extractor and its integration into the model. Finally, we describe multiple pre-trained model aggregations using ensemble.



**Fig. 2.** The architecture of the proposed Mask integration model. A 1D sequence is created from the input image, and the output is then fed into the transformer backbone. Pyramid vision transformer [2] is used to extract global features through various downsampling stages. For feature aggregation, the outputs from each stage are transformed into two-dimensional feature maps. The corresponding image mask is fed to several convolutional layers followed by an upsampler to extract the mask feature map. The concatenation of the pyramid feature aggregation and mask feature map is used as mask integrated feature map which is fed to a simple regression head with a multi-scale receptive field which is used to regress the final results.



### 3.2 Backbone Module

We use the CCTrans network as the backbone of our work. First, fixed-size image patches are created from the input images. Subsequently, the output is compressed into a 1D vector sequence. Afterward, we extract global features from the sequence using a pyramid transformer backbone. Then each stage’s 1D sequence is transformed into two-dimensional feature maps and upsampled to the same resolution. These feature maps are then subjected to an elementwise addition. Lastly, the density map is regressed using a straightforward regression head with multi-scale receptive fields. The loss functions of fully supervised and weakly supervised methods are constructed using the final density map and the sum of all of its pixel values, respectively.

**From 2D Image to 1D Sequence.** Before entering the transformer, images must be converted into 1D sequences by separating them into  $\frac{H}{K} \times \frac{W}{K}$  image patches, each of which is  $K \times K \times 3$  in size, with H and W standing for image height and width, respectively, and K for crop size.

**Pyramid Feature Aggregation.** The transformer-based backbone takes the 1D sequence. The Pyramid Vision Transformer employs multiple steps of down-sampling to obtain global context. Each stage’s outcomes are transformed into two-dimensional maps. Feature maps from top layers still lack detailed information that can not be reconstructed by up-sampling. The crowd counting method struggles to accurately identify crowd locations due to high-level features that cannot differentiate between objects. A feature pyramid is constructed to aggregate the semantic information from high-level layers with detailed information from low-level layers. Up-sampling the feature maps at every stage to the input image’s  $\frac{1}{8}$  size, as is often done in most work [14]. Additionally, this resolution facilitates a fair comparison with alternative approaches.

**Regression Head.** Inspired by [15] and [16], dilated Convolution layers with varying dilation rates are stacked in parallel to create a multi-scale Dilated Convolution (MDC) block. It contains three columns (C1, C2, C3) and a shortcut path. A single convolutional layer and a dilated convolutional layer form each column. To accommodate the crowd-counting scenes with plenty of small-scale objects, the associated kernel sizes and dilation rates are made as small as possible. After every convolutional layer is a batch normalization (BN) layer and a ReLU activation function. To use multi-scale features, the output feature maps are concatenated from each column and added via a shortcut path. Finally, the density map is regressed using a  $1 \times 1$  convolution layer.

**Loss Function Design.** The design of loss functions is founded on a well-known loss from [28], which is created by adding up the weights of the following losses: total variation (TV), Optimal Transport (OT) loss, and counting loss. The loss

function for a predicted density map ( $D_P$ ) and its ground truth ( $D_G$ ) is defined as:

$$L_{dm} = L_1(P, G) + \lambda_1 L_{OT} + \lambda_2 L_{TV}(D_P, D_G) \quad (1)$$

where  $P$  denotes the crowd count of PD and  $G$  denotes the crowd count of GD.  $\lambda_1$  and  $\lambda_2$  are the loss coefficients to control the importance of loss values.  $L_1$  denotes counting loss which is defined as the absolute difference between  $P$  and  $G$ .  $L_{OT}$  and  $L_{TV}$  equations defined in [28]. The model benefits from OT loss since it has a good fitting capacity in order to reduce the gap in distribution between the predicted and the ground truth density map. But because this method is not very good at simulating sparse crowds, [28] also employs an additional TV loss for stabilization. Total variation loss makes use of Ground-truth’s original head annotations, which aren’t smooth enough to provide accurate human representation. Crowds have a bigger scale, especially in some sparse situations, and it is unrealistic to describe a person by a pixel. [28] uses the mean square error L2 to regularise the difference between the smoothed annotation maps and the prediction to solve this problem. By using the adaptive Gaussian kernels, the smooth feature maps are generated [14]. The total loss is defined as:

$$L_d = L_1(P, G) + \lambda_1 L_{OT} + \lambda_2 L_2(D_P, D_G) \quad (2)$$

We have set  $\lambda_1$  and  $\lambda_2$  in our experiment to 0.1 and 0.01 correspondingly.

### 3.3 Mask-Aware Approach

Motivated by [12], There are several methods to include mask prediction data in the density map estimate process as a whole. We implement two of those solutions.

**Mask Generation.** All predicted masks in our work are generated using the EncoderDecoder model from MMSegmentation. It’s a multi-class segmentation model that consists of a Data preprocessor, Backbone, Decode head, and Auxiliary head. The data preprocessor is the part that copies data to the target device and preprocesses the data into the model input format. The Backbone is the part that transforms an image into feature maps. ResNet v1c is used as a backbone to our encoder-decoder. The decode head is the part that transforms the feature maps into a segmentation mask. PSPHead is used as a decode head model. The auxiliary head is a component that transforms the feature maps into segmentation masks. FCNHead is used as an auxiliary head [15].

**Mask Multiplication.** The first solution is motivated by [12] but with some little bit changes. [12] propose elementwise multiplying the density map generated from the density estimation network with the ground-truth mask in the training stage and in the testing stage the density map is multiplied by the predicted mask from the segmentation model [15]. Our proposed approach is

to elementwise multiply the image by the mask before feeding it to the density estimation network. We also use ground-truth masks in training stage and the predicted masks in testing stage.

**Mask Feature Extractor Network.** The second approach, which is motivated by [12], involves mapping the mask using many convolutional layers to create a feature map, which can then be combined with image features to estimate density. More details for mask features integration are shown in Fig. 2. The architecture of the mask features extractor is C(512, 256, 3)-C(256, 256, 3). If the used dataset has a ground truth mask that can be used in the training stage otherwise predicted masks from MMSegmentation are used. Predicted masks always are used at the testing stage.

### 3.4 Greedy Ensemble

The main idea of the model ensemble is to choose a group of models that achieve the best accuracy on the held-out validation set rather than a single fine-tuned model. We combine multiple models fine-tuned independently.

**Model Ensemble.** combines the models by aggregating the predictions of several base models. One popular type of ensemble is the greedy ensemble which operates by iterative adding models to the ensemble based on their performance on the held-out validation set. The selected aggregation is averaging the predictions of the multiple models. Only models that improve the accuracy will added to the ensemble.

**Table 1.** The mathematical equations for the ensemble.  $f(x, \Theta)$  is considered a neural network with input data  $x$  and parameters  $\Theta_i$

Method	Method Equation
Ensemble	$\frac{1}{k} \sum_{i=1}^k f(x, \Theta_i)$
Greedy ensemble	check algorithm 1

## 4 Experiments

In this section, we describe our experiments. We begin by outlining the datasets that were used. Next, we go over our training parameters, hyper-parameters, and evaluation metrics. Finally, we present a comparison with the state-of-the-art, followed by a discussion of our experiments.

---

**Algorithm 1.** Greedy ensemble

---

**Input** potential ensemble models  $\{M_1, \dots, M_k\}$ **Procedure**

```

ensemble  $\leftarrow \{\}$ 
for  $i \leftarrow 1$  to  $k$  do
  if  $\text{MAE}(\text{average}(\text{count}(\textit{ensemble}) \cup \text{count}(M_i))) \leq$ 
     $\text{MAE}(\text{average}(\text{count}(\textit{ensemble})))$  then
     $\textit{ensemble} \leftarrow \textit{ensemble} \cup \{M_i\}$ 
  end if
end for
return ensemble

```

---

## 4.1 Datasets

**ShanghaiTech** dataset [10] is a large-scale crowd-counting dataset that contains 1,198 annotated images with a total of 330,165 people with the centers of their heads annotated. This dataset consists of two parts: Part A consists of 482 images separated into 300 images for the training set and 182 images for testing. Part A images represent indoor and outdoor scenes and it’s randomly crawled from the internet where the resolution of each image is greatly different. Part B consists of 716 images separated into 400 images for training and 316 images for testing. Part B images are taken from streets in Shanghai so all images represent outdoor scenes and the image resolutions of which are  $768 \times 1,024$ .

**UCF-QNRF** dataset [17] includes 1,535 images of different scenarios from the Internet with 1,251,642 annotations, among which it is divided into 1,201 images for training and 334 images for testing. The number of pedestrians in each image varies from 49 to 12,865. This dataset image represents indoor and outdoor scenes. Furthermore, the image resolutions are very large and its scale varied dramatically comparing other datasets. It is a challenging dataset for crowd counting due to the large number of people in the images and the diversity of scenes.

## 4.2 Training Setting and Hyper-Parameter

We did our experiments on Linux based machine with GPU model NVIDIA RTX 4090, memory 24 GB on average 3GB used, driver Version 545.23.06 and CUDA Version: 12.3. The transformer-based backbone is the official Twins-SVT-large model, which is pre-trained on the ImageNet 1k dataset [29]. For every experiment, we only use random horizontal flipping and random cropping as data augmentations, which strictly follows [28, 30] The crop size is 256 for all datasets. We use AdamW [31] with a batch size of 8 for both ShanghaiTech parts A and B and the batch size changed to 12 with the QNRF dataset. We tuned the learning rate and weight decay value to avoid over-fitting and we found

the best value of weight decay is 1.0E−04 and the best value of learning rate is 2.31E−05. The regularization value used is 12.

For the mask generation model, the initial weights loaded from pre-trained weights on the cityscapes dataset. We also fine-tune the model on the ShanghaiTech part A dataset to generate all dataset masks. ShanghaiTech part A separated into 240 images for training, 60 images for validation, and 182 images for testing.

### 4.3 Evaluation Metric

We use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE or MSE for short). They can be formulated as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N | predicted_i - groundTruth_i | \quad (3)$$

$$MSE = \sqrt{\frac{1}{N} \sum_{i=1}^N | predicted_i - groundTruth_i |^2} \quad (4)$$

### 4.4 Comparison with the State-of-the-Art

In Tables 2 and 3, we demonstrate the effectiveness of our suggested approach on several crowd-counting datasets.

**ShanghaiTech Dataset** [10] ShanghaiTech part A is a challenging dataset as it has a small training set and its images are randomly gathered from online sources where the resolution of each image is substantially different. As shown in Table 2 our mask integration model exceeds CCTrans by 19.57% in MAE and exceeds Mask-aware by 21.16% in MAE. The greedy ensemble exceeds CCTrans by 22.36% and exceeds FGNet by 3.29%. ShanghaiTech Part B Dataset all its images are taken from streets in Shanghai and its results are shown in Table 2. We expect that might be because the part B dataset is considered a low-size dataset (the average number of people in the image is small compared with other datasets). Our model performs better on medium and large sized datasets.

**UCF-QNRF Dataset** [17] When compared to other datasets, the scale of the images varied significantly and the resolutions were very high. As shown in Table 3 our mask integration model exceeds CCTrans by 4.23% in MAE. The greedy ensemble exceeds CCTrans by 10.7% in MAE and exceeds FGNet by 3.52%.

**Table 2.** The performance comparison on the ShanghaiTech dataset.

Method	Shanghai Part A		Shanghai Part B	
	MAE	MSE	MAE	MSE
MCNN [10] (2016)	110.2	173.2	26.4	41.3
Switch [6] (2017)	90.4	135.0	21.6	33.4
Scale-adaptive [5] (2018)	86.8	139.2	16.2	25.8
ic-CNN [18] (2018)	68.5	116.2	10.7	16.0
CSRNet [14] (2018)	68.2	115.0	10.6	16.0
SANet [19] (2018)	67.0	104.5	8.4	13.6
CAN [9] (2019)	62.3	100.0	7.8	12.2
S-DCNet [21] (2019)	58.3	95.0	6.7	10.7
Mask-aware [12] (2020)	65.7	107.8	11.7	16.4
LSC-CNN [20] (2021)	66.4	117.0	8.1	12.7
CCTrans [11] (2021)	64.4	95.4	7.0	11.5
M-SFANet+M-SegNet [22] (2021)	57.6	94.5	6.32	10.1
P2PNet [23] (2021)	52.7	85.1	6.25	<b>9.9</b>
LoViTCrowd [24] (2022)	54.8	80.9	8.6	13.8
DMCNet [25] (2023)	58.5	84.6	8.6	13.7
SRN [27] (2024)	53.4	84.4	-	-
FGENet [26] (2024)	51.7	85.0	<b>6.3</b>	10.5
Mask integration (Ours)	51.8	<b>78.7</b>	7.6	12.3
Greedy ensemble (Ours)	<b>50.0</b>	83.1	7.0	12.2

**Table 3.** The performance comparison on the UCF QNRF dataset.

Method	MAE	MSE
CAN [9] (2019)	107	183
S-DCNet [21] (2019)	104.4	176.1
LSC-CNN [20] (2021)	120.5	218.2
CCTrans [11] (2021)	92.1	158.9
M-SFANet+M-SegNet [22] (2021)	87.6	147.8
P2PNet [23] (2021)	85.3	154.5
LoViTCrowd [24] (2022)	87.0	141.9
DMCNet [25] (2023)	96.5	164.0
SRN [27] (2024)	92.3	164.2
FGENet [26] (2024)	85.2	158.8
Mask integration (Ours)	88.2	148.8
Greedy ensemble (Ours)	<b>82.2</b>	<b>135.4</b>

## 4.5 Discussion of the Proposed Approaches

In this paper, we apply a model ensemble to aggregate several pre-trained model predictions. We trained several mask integration models with different learning rates then we examined whether to feed partially or all of them to the greedy ensemble and we found significant enhancement in the results. The pre-trained models are sorted ascending based on their MAE values. When the order of the pre-trained models is changed then the greedy ensemble will get different results.

## 4.6 Ablation Analysis

**Various Approaches.** We performed an ablation study in the ShanghaiTech dataset part A investigation to examine how various approaches affected our findings. We specifically explored variations that mask multiplication and mask integration approaches, as well as with the no-mask module. Comparing the various methods, our results showed that integrating masks produced the best results.

**Table 4.** Mask-aware approaches to achieve the best performance in ShanghaiTech dataset part A.

Method	MAE	MSE
CCTrans weakly-supervised (No mask)	64.4	95.4
CCTrans fully-supervised (No mask)	52.3	84.9
Mask multiplication	53.3	85.8
Mask integration	51.8	<b>78.7</b>
Mask multiplication (Ensemble)	51.9	85.5
Mask integration (Ensemble)	<b>50.0</b>	83.1

**Mask Quality.** We also studied the effect of mask quality on the model’s performance. We examine the use ground-truth mask instead of a predicted mask also we examine the use predicted mask without fine-tuning the mask generation model. For ShanghaiTech part A dataset, we use the ensemble with 10 models, and the MAE when we use a ground-truth mask is 50.02 and the MAE while using a predicted mask without fine-tuning is 50.32 while the MAE when we used a predicted mask generated from a fine-tuned model on the same dataset is 50.19. For the ShanghaiTech part B dataset, we use the ensemble with 5 models, and the MAE when we use a ground-truth mask is 7.01 while the MAE when we use a predicted mask generated from a fine-tuned model on part A dataset is 7.0, and the MAE when we use predicted mask generated from a fine-tuned model on part B is 7.03. In summary, the mask quality doesn’t affect that much on the model’s performance.

**Number of Ensemble Models.** We performed an ablation study on the effect of the number of models inserted into the ensemble on the accuracy and the processing time. We did this experiment on the ShanghaiTech part A dataset and we found that starting from 10 models the ensemble MAE is enhanced by 2.14% over the best model in the inserted models to the ensemble and the enhancement we gained from the more 17 models is not large compared to the corresponding increase in the time. In the Table 5 we document different numbers of models and the corresponding MAE, average time per image, and the number of selected models by the greedy ensemble.

**Table 5.** Ensemble with different number of models. Models inserted into the ensemble with MAE range from 51.84 to 56.75

Number of models	MAE	Avg. time per image (sec)	Number of selected models
5	50.73	0.50	3
10	50.19	1.46	6
15	50.19	2.69	6
20	50.19	3.98	6
25	50.19	5.26	7
27	50.00	5.93	8

**Scalability.** To check the model’s scalability and possibility of using its practical application. We observe the total training time for the M-trans on different datasets and also the average time for testing path (segmentation - M-trans - ensemble) for a single image. Table 6 and Table 7 show values for our experiments.

**Table 6.** The Average number of training epochs, Epoch time, and the average total training time.

Dataset	Number of epochs	Epoch time(sec)	Total training time(hour)
Shanghai part A	302	22.405	1.9
Shanghai part B	726	38.5	7.8
UCF-QNRF	200	226	12.6



**Table 7.** The average time per image of segmentation, M-trans testing, and ensemble time. The total time per image is averaging segmentation time + ensemble time.

Dataset	Number of models	Segmentation time(sec)	M-trans testing time(sec)	Ensemble time(sec)	Total time(sec)
Shanghai part A	10	0.019	0.075	1.46	1.479
Shanghai part B	5	0.024	0.089	0.594	0.618
UCF-QNRF	5	0.1047	1.0267	5.8526	5.957

## 5 Conclusion

In this study, we have proposed a novel mask-aware transformer-based architecture (M-Trans) designed to address the challenges of crowd counting by integrating mask information into a transformer-based model. We apply two different approaches to this integration: 1. Mask multiplication. 2. Mask feature extractor network. Also, we improve the performance by applying a greedy ensemble strategy. Our work contributes to advancing the state-of-the-art in crowd-counting performance on different datasets.

## References

1. K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. Loy, and D. Lin, "MMDetection: Open MMLab Detection Toolbox and Benchmark", [arXiv:1906.07155](https://arxiv.org/abs/1906.07155)
2. W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo and L. Shao, "Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions", IEEE/CVF International Conference on Computer Vision (ICCV), 2021
3. A. Chan and N. Vasconcelos, "Bayesian Poisson regression for crowd counting", IEEE 12th international conference on computer vision, 2009
4. C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks", IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2015
5. L. Zhang, M. Shi and Q. Chen, "Crowd Counting via Scale-Adaptive Convolutional Neural Network". IEEE Winter Conference on Applications of Computer Vision (WACV), 2018
6. D. Sam, S. Surya and R. Babu, "Switching Convolutional Neural Network for Crowd Counting". IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017
7. S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". Advances in Neural Information Processing Systems 28 (NIPS 2015)
8. C. Liu, Y. Tao, J. Liang, K. Li and Y. Chen, "Object Detection Based on YOLO Network". IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), 2018
9. W. Liu, M. Salzmann and P. Fua, "Context-Aware Crowd Counting". IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019

10. Y. Zhang, D. Zhou, S. Chen, S. Gao and Y. Ma, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016
11. Y. Tian, X. Chu, and H. Wang, "CCtrans: Simplifying and improving crowd counting with transformer", [arXiv:2109.14483](https://arxiv.org/abs/2109.14483), 2021
12. S. Jiang, X.Lu, Y. Lei and L. Liu, "Mask-Aware Networks for Crowd Counting", IEEE Transactions on Circuits and Systems for Video Technology, 2020
13. D. Sam and R. Babu, "Top-Down Feedback for Crowd Counting Convolutional Neural Network", arXiv preprint [arXiv:1807.08881](https://arxiv.org/abs/1807.08881), 2018
14. Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018
15. L. -C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs" IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018
16. S. Liu, D. Huan,g and Y. Wang, "Receptive Field Block Net for Accurate and Fast Object Detection", Proceedings of the European Conference on Computer Vision (ECCV), 2018
17. H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds", Proceedings of the European Conference on Computer Vision (ECCV), 2018
18. V. Ranjan, H. Le and M. Hoai, "Iterative Crowd Counting", The European Conference on Computer Vision (ECCV), 2018
19. X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale Aggregation Network for Accurate and Efficient Crowd Counting", The European Conference on Computer Vision (ECCV), 2018
20. D. Sam, S. Peri, M. Sundararaman, A. Kamath and R. Babu, "Locate, Size, and Count: Accurately Resolving People in Dense Crowds via Detection", Pattern Analysis and Machine Intelligence, 2021
21. H. Xiong, H. Lu, C. Liu, L. Liu, Z. Cao, and C. Shen, "From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer," IEEE/CVF International Conference on Computer Vision (ICCV), 2019
22. P. Thanasutives, K. Fukui, M. Numao and B. Kijirikul, "Encoder-Decoder Based Convolutional Neural Networks with Multi-Scale-Aware Modules for Crowd Counting," International Conference on Pattern Recognition (ICPR), 2021
23. Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Wu, "Rethinking Counting and Localization in Crowds: A Purely Point-Based Framework", IEEE/CVF International Conference on Computer Vision (ICCV), 2021
24. N. H. Tran, T. D. Huy, S. T. M. Duong, P. Nguyen, D. H. Hung, C. D. Tr. Nguyen, T. Bui, and S. Q.H. Truong, "Improving Local Features with Relevant Spatial Information by Vision Transformer for Crowd Counting", British Machine Vision Conference (BMVC), 2022
25. M. Wang, H. Cai, Y. Dai, and M. Gong, "Dynamic Mixture of Counter Network for Location-Agnostic Crowd Counting", IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2023
26. H. Ma, L. Zhang, and X. Wei, "FGNet: Fine-Grained Extraction Network for Congested Crowd Counting", [arXiv:2401.01208](https://arxiv.org/abs/2401.01208), 2024

27. Z. Xiong, L. Chai, W. Liu, Y. Liu, S. Ren, and S. He, "Glance To Count: Learning To Rank With Anchors for Weakly-Supervised Crowd Counting", IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2024
28. Wang, B., Liu, H., Samaras, D., Hoai, M.: Distribution matching for crowd counting. Neural Information Processing Systems, NeurIPS (2020)
29. J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database", IEEE Conference on Computer Vision and Pattern Recognition, 2009
30. Z. Ma, X. Wei, X. Hong and Y. Gong, "Bayesian Loss for Crowd Count Estimation With Point Supervision", International Conference on Computer Vision (ICCV), 2019
31. I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization", International Conference on Learning Representations, 2019



# Unsupervised Real-Time Two-Stage Place Proposal Generation from a Moving Camera Video

H. Işıl Bozma<sup>(✉)</sup> 

Intelligent Systems Laboratory, Electrical Electronics Engineering, Boğaziçi University, Bebek,  
34342 Istanbul, Turkey  
bozma@bogazici.edu.tr  
<http://isl.bogazici.edu.tr>

**Abstract.** A video as acquired by a moving camera contains visual data pertaining to different places (i.e. a particular room, a corridor, etc.) that have been traversed through. Place proposal generation refers to delineating the correct place-related boundaries in the incoming video and encoding the respective visual data. This is an important problem since the resulting representation can be used for video-based place analysis. To this end, we propose a novel two-stage unsupervised place proposal generation framework that works real-time on unlabeled videos as acquired by a moving camera. First, each distinct place is delineated based on the continuous iterative partitioning of the incoming frames - considering their informativeness, coherency and plenitude. Following, “canonical scenes” within each generated place proposal are identified based on the hierarchical clustering of the respective frames. This enables larger or cluttered places that have multiple scenes to have better representations. Experimental results on benchmark data as well as real-time data demonstrate superior video place analysis performance as compared to a baseline approach.

**Keywords:** Video segmentation · Vision for robotics · Place recognition · Place learning · Video processing.

## 1 Introduction

This paper is focused on generating place proposals from RGB video as it is being acquired by a moving camera. Each place proposal refers to a specific spatial region similar to human’s concept of a ‘place’ and is then labeled accordingly such as ‘X’s kitchen’ or ‘Y Park entrance [19]. It is known that even in outdoors without any clear perceptual or physical boundaries, such a decomposition of space is done. In this paper, we also adopt this definition of a place. As such, a place proposal is defined by a collection of appearances sharing common perceptual signatures or physical boundaries. If place proposals can be generated in an identity-independent manner - namely no place-specific a priori knowledge being used - then they can be used as a basis for unsupervised video-based place related reasoning. For example, if the three places as shown

TUBITAK grant EEEAG-118E857 and ROYAL CB SBB 2019K12-149250.

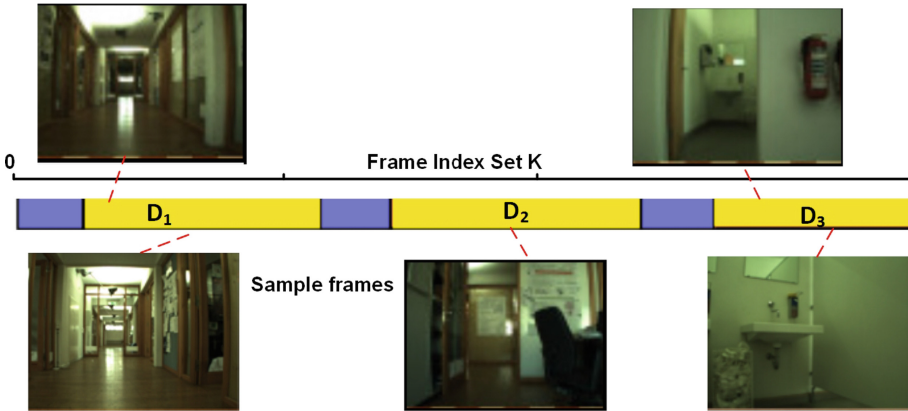
© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

A. Antonopoulos et al. (Eds.): ICPR 2024, LNCS 15318, pp. 31–45, 2025.

[https://doi.org/10.1007/978-3-031-78456-9\\_3](https://doi.org/10.1007/978-3-031-78456-9_3)

in Fig. 1 could be generated as place proposals from the incoming RGB video, learning these places or recognizing them in future visits would both become possible. Many human-machine interaction tasks would benefit from such a human-like detection of places.

This problem is related to the video scene segmentation problem - a fundamental step used for video summarization and browsing [6]. They both aim to partition the frames' sequence based on their temporal coherence. However, differing from video scene segmentation problems, there is only one camera, processing needs to be real-time and while changes between consecutive frames tend to be gradual, each place can contain several scenes depending on the camera's viewpoint. For example, larger or cluttered places that are partially visible upon entry turn out to be challenging - as appearances tend to vary significantly depending on camera's viewpoint. In such cases, the corresponding canonical scenes associated with each generated place proposal need to be identified separately. Hence, reliable partitioning becomes difficult. Furthermore, the resulting representation must also enable reliable recognition in incoming videos when moving through these places in future.



**Fig. 1.** Place proposal generation as the video is being acquired through traversing 3 places. First, each place needs to be delineated as indicated by yellow indexed frames. (Blue regions correspond to transition regions.) Next, if a delineated place (i.e places 1 and 3) has multiple canonical scenes, these need to be identified as well.

We propose to address this problem through introducing a novel two-stage place proposal generation method. In the first stage, each distinct place is delineated based on the continuous iterative partitioning of the incoming frames - considering their informativeness, coherency and plenitude based on previous work [12]. In the second stage, "canonical scenes" within the delineated place are determined using hierarchical clustering of the respective visual data. The level with the maximal gain in the resulting hierarchy is then used to determine the canonical scenes. Our approach is unsupervised since it does not require any place-specific a priori knowledge. It is also applicable in real-time since the processing of the first stage operates on each incoming video frame.

Finally, the proposed multi-scene representation provides an efficient mechanism for identifying the visually different parts of each place proposal. As such, semantic place analysis performance can improve significantly as demonstrated in the experimental results with benchmark data sets.

The paper is organized as follows. First, related work is covered in Section 2. The proposed two-stage place proposal generation framework is presented in Section 3. We then discuss how the place proposals can be used for video-based place analysis in Section 4. In Section 5, the video datasets and the experimental protocol are presented. A qualitative evaluation demonstrates the resulting performance on these datasets followed by its usage for video-based place analysis that shows the applicative potential of our approach as well as its superiority in comparison to the baseline approach. The paper concludes with a brief summary where we give some perspectives for future work.

## 2 Related Literature

The generation of place proposals is related with two areas: video scene segmentation and place detection in robot vision. We review work in both areas in this section.

Video scene segmentation (VSS) model the task from a global view of the frame. Interestingly, this problem is found to be important event for video object segmentation [28]. The proposed methods differ depending on whether they are category-aware or not. Category-aware approaches such as action spotting split the video into smaller segments based on frame parsing and tracking of actions [15]. Alternatively, class agnostic approaches assume the video to be structured in a specific way - namely a sequence of frames that can be divided into ‘shots’ (coherent sequence of frames) and ‘scenes’ (a sequence of shots) in terms of the granularity of semantics [16]. Generally, shots are separated by one of the several motion picture effects such as cuts, fade, dissolve or camera motion such as rotating or zooming. Existing VSS algorithms exploit the temporal relationships between consecutive shots and group several consecutive shots into a single scene that can be conceptualized as a sub-story occurring within a particular environment . Temporal relationships are commonly established based on visual features such as SURF [1] or as determined from a deep learning network [4, 5, 9]. Alternatively, in graph-based methods, instead, shots are arranged in a graph representation and then clustered by partitioning the graph based on shot similarity [2]. Boundary detection is performed by analyzing the dissimilarity of successive frames where high dissimilarity indicates the boundary [35]. Although these approaches can easily detect abrupt shot changes such as hard cuts, gradual shot changes that spread over the number of frames are relatively hard to detect.

Place detection considers a similar problem in robot and vehicle vision. Similar to VSS, this requires partitioning the incoming video stream into semantically meaningful sets where each corresponds to one particular place. However, this problem differs from VSS problems in three aspects: i) All frames are acquired by a single camera with changes between consecutive frames being gradual; ii) The processing must be real-time; iii) The resulting representation must also enable reliable recognition in videos acquired via passing through the same places. In one group of approaches, the consistency of the frame data is tracked with discontinuities signaling transitions among

places. Tracking is done mostly using local features such as SIFT or SURF descriptors [10, 14, 25]. Alternatively, global descriptors such as images [18], optical flow [21], census transform [34], histograms [11] or hybrid descriptors such as bag-of-words [20, 22, 32] and bubble descriptors [12] have also been used. There are also some work in which temporal nature is not considered and the problem is posed as a clustering problem [17]. The methods in the second group consider the scene contents at a higher-level and hence require frame parsing. For example, detection is based on identifying passages, doors or room structures [3, 31, 33]. As these methods primarily focus on transition regions, places are detected on the basis of being separated by transition regions. This may be problematic if transition regions are not obvious. Alternatively, place contents based on “proto-objects” (segments or blobs of uniform visual properties such as color or disparity) are used to define a detected place [7].

In all of these methods, each place proposal is then either represented by a single key frame or a single classifier based on the corresponding frames. As recognition is typically based on matching the information obtained from the current set of frames to the previously learned knowledge of places, the performance can deteriorate if there are large variations in the visual appearances within a detected place. In this paper, we introduce a two-stage unsupervised place proposal generation method in order to address this problem. Differing from previous work, our method generates the place proposals with their canonical scenes.

### 3 Two-Stage Place Proposal Generation

Consider the sequence of frames  $F_k$ ,  $k \in \mathcal{K}$  in an incoming video where  $\mathcal{K}$  denotes the frames’ index set. Let each frame  $F_k$  be encoded by a descriptor  $I_k$ . The descriptor can be formed using one or several of the representations that have been developed - as long as they are able to reliably represent the visual data. Here, we use bubble descriptors [8]. We prefer to use this representation due to its shown comparative advantages to other representations such as preserving the relative  $S^2$  geometry of visual features, being rotationally invariant and incorporating any number of observations. However, the proposed approach is in no way dependent on this particular choice and thus can be used with other kinds of descriptors.

#### 3.1 First Stage: Place Boundary Delineation

The first stage continually determines where a place starts and ends in the incoming RGB video. Each delineated place  $m \in \mathcal{D}$  is defined by  $D_m \subset \mathcal{K}$  set and  $\mathcal{D} = \{1, \dots, m^*\}$  denotes the index set of place proposals. The delineation is defined by the continuous iterative partitioning of the incoming frames as defined by the as presented in [12]. For completeness, a brief summary is presented here. The interested reader is referred to [12] for details.

The partitioning process has three states: delineation start, delineation in progress and delineation termination. Each incoming frame is first checked for its informativeness. Informativeness is related to semantic content of the respective frame. For example, in cases of low illumination or camera being very close to an object, the associated

data will not be informative and needs to be ignored. As such, it is measured by a binary valued function  $\varsigma(F_k) \in \{0, 1\}$  that depends on a priori determined threshold values. Following, its coherency is computed. This is measured by a binary valued function  $\kappa(F_k) \in \{0, 1\}$  that is defined on the space of descriptors encoding each frame. While data from consecutive frames in one place tend to be coherent, data from consecutive frames while in transition between two different places will be incoherent.

If a new delineation is started with index  $k$ , its extent  $\mathcal{Q}^*(k) \subset \mathcal{K}$  is determined by iteratively partitioning the consecutive frame indexes that are both informative and coherent. The iteration is defined as:

$$\mathcal{Q}^{i+1}(k) := \left( \bigcup_{j \in \mathcal{Q}^i(k)} \mathcal{Q}^1(j) \right) \text{ where}$$

$$\mathcal{Q}^1(k) \triangleq \{k+1 \mid \varsigma(F_{k+1}) = 1 \text{ and } \kappa(F_{k+1}) = 1\}$$

According to this definition, each  $(i+1)^{st}$  neighbor of frame with index  $k$  is informative and coherent wrt to some  $i^{th}$  neighbor of starting frame index  $k$ . The iterative process continues until uninformativeness or incoherency is detected. Let  $i^*$  be the corresponding index. In this case, a temporal window  $T^1(k)$  is started:

$$T^1(k) \triangleq \{\chi_1(k)\}$$

where  $\chi_1(k)$  denotes the index of uninformative or incoherent frame closest to  $F_k$ . A temporal window contains indexes that are uninformative or incoherent. It may correspond to temporary lapses in the delineated place or signal transitions between two different delineations. As new frames are received, uninformative frames or incoherent frames extend the temporal window:

$$T^{i+1}(k) := T^i(k) \bigcup_{j \in T^i(k)} \{k' \leq \chi_2(j)\}$$

where  $\chi_2(j)$  to be the smallest index that is at most  $\tau_n$  distant to index  $j$  while still being either not informative or incoherent: The incoherency extension threshold  $\tau_n$  defines the number of succeeding indexes that will be checked. If there is at least one uninformative or incoherent frame in the next  $\tau_n$  indexes, then the temporal window is extended to include the respective index. Thus, each  $T^i(k)$  th temporal window contains uninformative or incoherent frame indexes to some  $i^{th}$  neighbor of frame with index  $k$ . Let  $i^*$  be the corresponding frame index when the temporal window is terminated:

$$T^*(k) := T^{i^*}(k)$$

Once a temporal window terminates, the extent of  $T^*(k)$  as compared to temporal window extent parameter  $\tau_w$  is used to decide how to use this knowledge. A short extent indicates sensing problems. In this case, the associated frames are simply ignored and delineation continues. On the other hand, a long extent signals transition regions which suggests that the regions before and after the transition region need to be detected as



two different places. In this case, current delineation ends and a new delineation starts. Each delineated place is checked for plentitude - namely whether its extent is sufficiently long or not. Only delineated places with sufficiently long extents are then considered as detected places. Each detected place  $D_m$  is associated with corresponding set of descriptors  $I_l, l \in D_m$ . The mean descriptor  $\bar{I}_m$  is defined as:

$$\bar{I}_m = \frac{1}{|D_m|} \sum_{l \in D_m} I_l \quad (1)$$

### 3.2 Second Stage: Finding Canonical Scenes

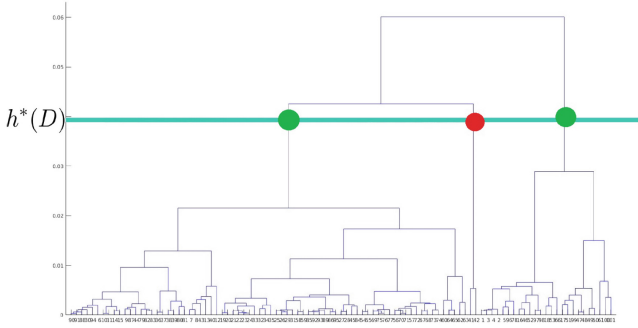
In the second stage, canonical scenes are determined within each delineated place  $D_m$ . Let  $n_s(m)$  be the number of canonical scenes. Note that if the delineated place  $D_m$  is small and unobstructed, appearances from different viewpoints will not vary significantly and  $n_s(m) = 1$ . Canonical scenes are found using the hierarchical clustering method SLINK [29]. In this method, an hierarchy is built in an incrementally bottom-up manner through the nested sequence of clusters of the frames (as indexed by  $D_m$  that are associated with the detected place. Let  $E(D_m)$  denote the set of equivalence relations on  $D_m$  and  $\zeta : \mathbb{R}^{\geq 0} \rightarrow E(D_m)$  be the clustering function. Each cluster is associated with an height  $h$  as measured by a similarity metric - namely  $\zeta(h + \delta h) = \zeta(h) \forall \delta h \approx 0$ . As the height  $h$  increases, clusters get larger - namely  $0 \leq h \leq h' \rightarrow \zeta(h) \subseteq \zeta(h')$ . Finally, the top level is a single cluster - namely  $\exists h^r$  s.t.  $\zeta(h^r) = D_m \times D_m$ . The resulting nested sequence of partitions can be shown to be equivalent to tree hierarchy consisting of  $n_L$  levels. Each level  $l = 0, \dots, n_L$  is associated with a height  $h_l \in (0, h^r]$ . The height  $h_0 = 0$  corresponds to the terminal nodes while  $h_{n_L} = h^r$  corresponds to the root node.

Once a place is detected, canonical scenes are detected automatically by finding the level  $h^*(D_m) \in (0, h^r)$  with the maximal height increment  $\delta h_l = h_{l+1} - h_l$  - namely  $h^*(D_m) \in \arg \max_l \delta h_l$ . Outlier subsets are pruned by only considering only those clusters that have cardinality greater than a preset threshold  $\tau_d$ . Each remaining cluster  $D_{mj} \subseteq D_m, j = 1, \dots, n_s$  with sufficient cardinality - namely  $|D_{mj}| > \tau_d$  then corresponds to one canonical scene in the detected place.

A sample case of finding canonical scenes is as shown in Fig. 2a. Here, 101 frames have been delineated in the first stage of proposed approach. The height  $h^*(D)$  with maximal increment contains 3 clusters with scenes as shown in Fig. 2a-2b. The cluster with low cardinality is pruned out so that two canonical scenes are detected. Two remarks are noteworthy: First, canonical scenes are based only on the similarity of frame data. Hence, frames associated with each canonical scene set are not necessarily temporally related. Second, when a large number of frames that have gradually decreasing visual overlap exist in a generated place proposal (ie movement along a corridor), the hierarchy typically ends up having two balanced clusters.

## 4 Video-Based Place Analysis

Place proposals can be used for video-based place analysis. In particular, we focus on learning places so that they can be recognized in videos obtained by traversing these



(a) Finding the canonical scenes within a delineated place consisting of 101 frames. In the resulting hierarchy, the level with maximal height increment  $h^*(D)$  designates 3 potential canonical scenes as shown by the circles. As one (red circle) does not have sufficient number of frames, the place is finally associated with 2 canonical scenes (green circles).



(b) 1<sup>st</sup> canonical scene



(c) 2<sup>nd</sup> canonical scene



(d) Scene with low cardinality

**Fig. 2.** Finding canonical scenes of a place proposal.

places in future. In this section, we present one approach to this as presented in [13]. However, other approaches including deep learning based approaches are also possible.

We first consider a place memory that enables the storage and retrieval of learned places as defined by the index set  $\mathcal{P}$ . We consider a hierarchical organization as defined by a nested sequence of partitions  $\mathcal{P}$ . The partition at the top level is the whole set  $\mathcal{P}$ . All inner nodes correspond to particular subsets  $\mathcal{P}(N) \subset \mathcal{P}$  while each  $N$  of terminal nodes corresponds to a distinct place  $p(N) \in \mathcal{P}$ . Such an organization enables associating an incoming place proposal  $D_m$  with the learned place knowledge efficiently through traversing down the memory hierarchy. Traversal is done level by level until either it reaches the terminal level or check condition is not satisfied. At each level, the similarity of each canonical scene with the learned places encoded by the children nodes  $N^\downarrow$  is evaluated and the node  $N^*$  with the maximum similarity is determined based on the discriminant function  $d_N(D_m)$ . It also ensures that the similarity is sufficiently high by checking against a preset recognition threshold  $\tau_r$ .

$$N^* \in \arg \max_{N \in N^\downarrow} d_N(D_m) \text{ subject to } d_N(D_m) \geq \tau_r \quad (2)$$

The discriminant function  $d_N$  considers each canonical scene one-by-one. In levels up to the final terminal level, it is defined as:

$$d_N(D_m) = \sum_{j=1}^{n_s} \frac{|D_j|}{|D_m|} (\rho_j(\bar{I}_N) + \eta_j(N)) \quad (3)$$

The term  $\rho_j(\bar{I}_j) \geq 0$  is the Pearson correlation that evaluates how well overall the frames of the canonical scene  $D_{mj}$  as encoded by the descriptor  $\bar{I}_{mj}$  match those associated with the current node  $N$  as encoded by the descriptor  $\bar{I}_N$  where:

$$\bar{I}_{mj} = \frac{1}{|D_{mj}|} \sum_{k \in D_{mj}} I_k \text{ and } \bar{I}_N = \frac{1}{|\mathcal{P}(N)|} \sum_{p \in \mathcal{P}(N)} I_p$$

The second term  $\eta_{jm}(N) \geq 0$  computes the overall SVM support for the canonical scene  $mj$  considering each frame associated with canonical scene  $D_{mj}$ :

$$\eta_{mj}(N) = \frac{1}{|D_{mj}|} \sum_{k \in D_{mj}} \nu_N(I_k) \text{ where } \nu_N(j) = \begin{cases} 1 & \text{if } \psi_N(I_k) > 0, \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

In the final terminal level, for each considered terminal node  $N$ , the discriminant function is defined by considering the  $n_s(p(N))$  canonical scenes associated with it:

$$d_N(D_m) = \sum_{j=1}^{n_s} \arg \max_{l \in \{1, \dots, n_s(p(N))\}} \frac{|D_j|}{|D_m|} (\rho_j(\bar{I}_{ml}) + \eta_l(D_{mj})) \quad (5)$$

where  $I_{ml}$  is the mean descriptor associated with the  $l$ -th canonical scene. This process is repeated until either a terminal node is reached or the check condition of Eq. 2 is not satisfied. In the former case, the place is recognized to be the place associated with the terminal node. In this case, the robot also updates its memory via incorporating the new knowledge appropriately. The recognition threshold  $\tau_r$  is a designating factor in the trade-off between precision and recall. As  $\tau_r$  increases, precision increases and recall decreases.

In case of no recognition, place learning is invoked in order to add the place proposal  $D_m$  into the place memory as a new place  $p$ . Learning a place consists of three steps: i) Modifying the place memory hierarchy incrementally in order to accommodate the new place proposal  $D_m$ ; ii) Updating the one-SVM discriminant functions  $\psi_N$  at the changed nodes  $N$  of the hierarchy associated with the place memory. This requires learning the cost functions  $\psi_N$  that are associated with the nodes of the place memory using one-class SVM [27]. The  $\psi_N$  functions are used in defining the discriminant functions  $d_N$  that are used in recognition. Thus, one-SVM discriminant functions  $\psi_N$  of the nodes associated with the changed parts of the hierarchy are relearned. iii) Associating the  $n_s(m)$  canonical scenes with the newly added place  $D_m$ .

## 5 Experiments

To evaluate our proposed approach, we carry out comprehensive experiments on benchmark datasets - including comparative studies with a baseline approach. Each incoming

**Table 1.** Place Proposal Generation Performance

(a) Recall Rates				(b) Precision Rates				(c) Canonical Scenes' Statistics			
Video	Cloudy	Sunny	Night Evening	Video	Cloudy	Sunny	Night Evening	Video	# Places	Mean $n_s$	Variance of $n_s$
Fr	0.68	0.65	0.71	Fr	0.79	0.63	0.71	Fr	24	1.21	0.26
Lj	0.28	0.41	0.29	Lj	0.47	0.54	0.38	Lj	25	1.28	0.46
Sa	0.53	0.65	0.71	Sa	0.67	0.81	0.67	Sa	16	1.31	0.23
NC	-	0.6	0.5	NC	-	0.46	0.42	NC	11	1.08	0.08
								Sy	12	1.50	0.25

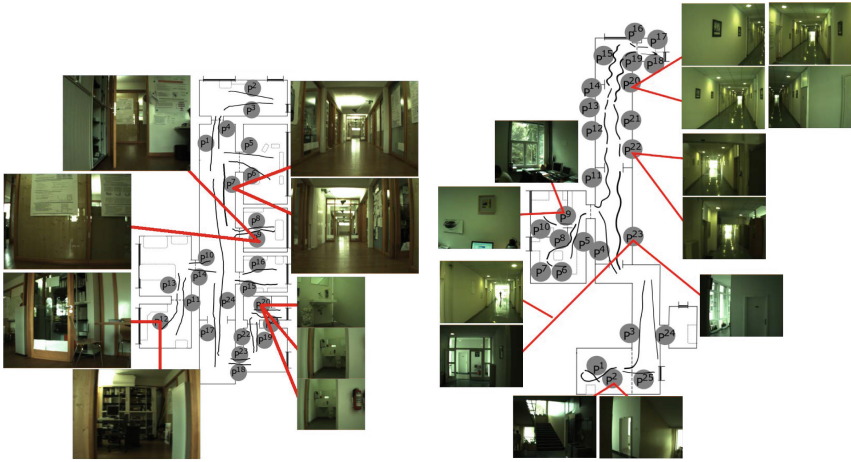
frame is encoded with a  $d = 600$ -dimensional descriptor [8]. Canonical scenes are detected with  $\tau_d = 10$  as determined experimentally. The per frame processing time depends on the extent of the detected place as expected or as place memory grows larger. Altogether, real-time performance of the whole system is around 2-3 frames per second with a standard I5 CPU processor.

## 5.1 Datasets

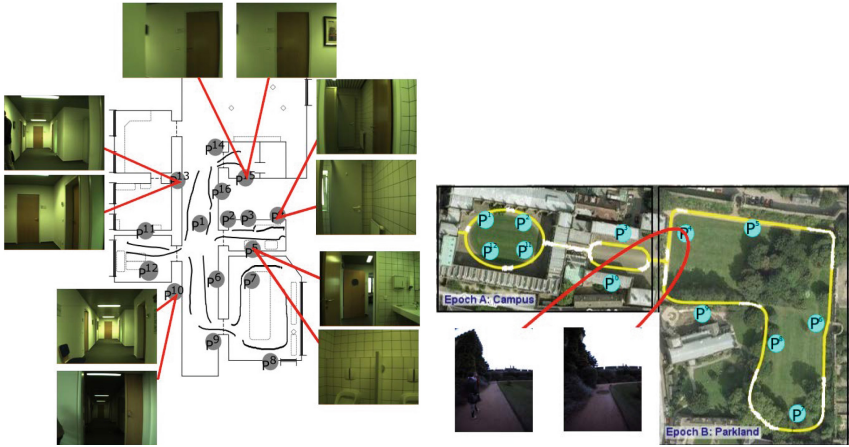
Datasets are RGB videos from a variety of different routes: i) Benchmark COLD dataset [23]; ii) New College (NC) dataset [30] and iii) SYNTHIA (SYN) dataset [26]. The COLD dataset consists of indoor videos taken from three different routes - Freiburg (Fr), Ljubljana (Lj) and Saarbrucken (Sa) routes under cloudy conditions. A typical video contains around 1000-2000 frames. The NC dataset is recorded with a perspective camera along an outdoor route of approximately two kilometers coverage with around 5000 frames per video. SYNTHIA dataset is a synthetic dataset from a car traveling along a highway in spring time and . In each case, the ground truth of detected places is obtained from the accompanying route and map data while that of canonical scenes is based on visually inspecting whether they correspond to distinct characterizing scenes in the detected place or not - as there is no ground truth provided. In each case, there are multiple videos along the same route, but taken at different times and varying wrt to illumination conditions. As such, datasets such as MovieNet [24] that are typically used in VSS evaluations are not suitable in our case.

## 5.2 Ablation Tests

The goodness of the generated place proposals is evaluated in comparison to the ground truth, which is obtained manually by considering the route coordinates and map data provided as well final visual checking. Note that there are small variations along each route as well. Moreover, there exists frames in the night/evening dataset where the illumination conditions is impossible to reverse, e.g no light source . A detected place  $D_m$  is determined to be correct if their IoU values with respect to the ground truth  $G_m \subset \mathcal{K}$  is at least 50% - namely  $\frac{|D_m \cap G|}{|G|} \geq 0.5$ . Furthermore, they are constrained to have compatible extents. This is checked as  $\frac{|D|}{|G|} \leq \tau^+$ .  $\tau^+$  is the upper bound of extent ratio and is set to  $\tau^+ = 2$ . Thus, the extent of a detected place can be at most twice



(a) In Fr video, places 7,9 and 12 have 2 canonical scenes while place 20 has 3. (b) In Lj video, places 2,9,22 and 23 have 2 canonical scenes while place 20 has 4.



(c) In Sa video, places 4,5,10,13 and 15 all have 2 canonical scenes. (d) In NC video, only place 4 has 2 canonical scenes.

**Fig. 3.** Places and canonical scenes. For places  $m$  with  $n_s(m) \geq 2$ , a sample image associated with each canonical scene is as shown.

of its ground truth. Hence, place proposals having an extremely long extents are not considered to be correct.

For sample COLD videos, the resulting place proposals are as shown in Fig. 3a-Fig. 3c. There are 24 place proposals as seen in Fig. 3a. Places 7, 9 and 12 are found to contain 2 canonical scenes while place 20 is found to contain by 3 canonical scenes. It should be remarked that some of the places that are traversed in opposite directions, the collected appearances have gradually decreasing visual overlap and the resulting clustering hierarchy turns out to be balanced - as is the case for place 7 in the Fr video.

Hence, the place is associated with two canonical scenes depending on the navigation direction. In the Lj video, places 2, 9, 22 and 23 are found to consist of 2 canonical scenes while place 20 consists of 4 canonical scenes as seen in Fig. 3b. In the Sa video, places 4,5,10,13 and 15 are all found to have 2 canonical scenes as seen in Fig. 3c. In the NC video, most of the places have open space in general so that these consist of a single canonical scene as seen in Fig. 3d. In the SYN videos, there are 8 places detected - with half of them consisting of 2 canonical scenes. Based on how each place is viewed and the openness of space, we expected the number of canonical scenes within each place proposal generated to be lower for outdoors places as is the case here. In all, it is observed that the canonical scenes correspond to distinct characterizing views in the detected place.

The number of places detected in each video as well the average number of canonical scenes and their variance are given in Table 1c. It is observed that the mean number of canonical scenes in each video varies between 1 and 2 which implies in this data set that most of the detected places are relatively small and do not contain major obstructions. As the size of a place increases (ie a corridor) or the clutter of a room increases (ie lab), the number of canonical scenes also tends to increase. Let it be also remarked abrupt camera movements also lead to canonical scenes as is the case in place 15 in the Sa video or due to dynamic scene entities (a walking person) as is the case in place 4 in the NC video.

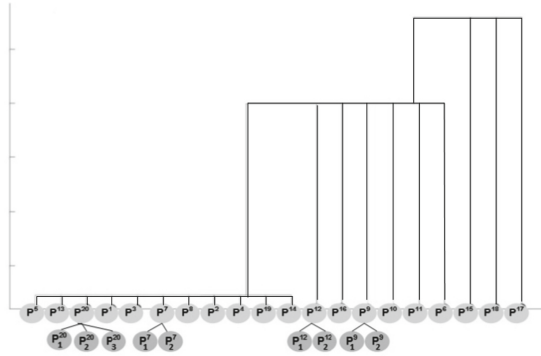


Fig. 4. Fr video - Learned places with canonical scenes.

### 5.3 Video-Based Place Analysis - Comparative Performance

Next, we compare the proposed approach with a baseline approach<sup>1</sup> in which only first level processing is applied - hence there are no canonical scenes [13]. The comparison is done with respect to the resulting video-based place analysis performance on two

<sup>1</sup> For the comparison of the baseline model with other approaches, we kindly refer the interested readers to the respective references [13].

**Table 2.** Precision and recall rates in  $2^{nd}$  videos.

(a) Fr video					(b) Lj video				
$\tau_r$ \ Method	Baseline		Proposed		$\tau_r$ \ Method	Baseline		Proposed	
	Precision	Recall	Precision	Recall		Precision	Recall	Precision	Recall
1	0.64	0.37	0.78	0.37	1	0.37	0.50	0.50	0.64
1.2	0.67	0.32	0.78	0.37	1.2	0.39	0.50	0.50	0.57
1.4	0.75	0.32	0.83	0.26	1.4	0.38	0.43	0.53	0.57
1.6	0.80	0.21	0.83	0.26	1.6	0.43	0.43	0.57	0.57
1.8	1.00	0.21	1.00	0.21	1.8	0.45	0.36	0.62	0.57

(c) Sa video					(d) New College video				
$\tau_r$ \ Method	Baseline		Proposed		$\tau_r$ \ Method	Baseline		Proposed	
	Precision	Recall	Precision	Recall		Precision	Recall	Precision	Recall
1	0.27	0.23	0.50	0.54	1	0.55	0.60	0.44	0.40
1.2	0.33	0.23	0.50	0.46	1.2	0.44	0.40	0.50	0.40
1.4	0.33	0.23	0.55	0.46	1.4	0.50	0.30	0.50	0.30
1.6	0.43	0.23	0.50	0.31	1.6	0.60	0.30	0.60	0.30
1.8	0.40	0.15	0.50	0.23	1.8	1.00	0.10	0.67	0.20

(d) SYN - Highway video				
$\tau_r$ \ Method	Baseline		Proposed	
	Precision	Recall	Precision	Recall
1	0.57	0.5	0.71	0.68
1.2	0.67	0.5	0.71	0.68
1.4	0.60	0.38	0.80	0.50
1.6	0.75	0.38	0.75	0.38
1.8	1	0.13	1	0.25

videos along each of the routes, but obtained at different times. The first video is used for learning the detected places while the second video is used to evaluate the recognition performance. As discussed, learned places are stored in a place memory. For example, place memory corresponding to the Fr video is as shown in the Fig. 4. Note that the detected place of Fig. 2 (with two canonical scenes) has been learned as place 7 in this memory. This memory is then used in the recognition experiments. Recall and precision rates are presented in Table 2a-2c. The recognition threshold  $\tau_r$  is varied between 1 to 1.8. It is observed that indoors recognition performance is significantly enhanced with an average of 15% improvement for both precision and recall rates in comparison to the baseline approach. For example, while place 10 in Sa video cannot be recognized in the baseline method, this is not the case with the proposed approach. With the 2 canonical scenes, it becomes possible to recognize in the second video. The highest improvement happens in the Sa video. This is expected as it has the highest mean  $n_s$  which implies that a higher proportion of the places are represented by multiple canonical scenes. Interestingly, such improvement is not observed with the places in the NC video. This is expected since most of the places are characterized by one canonical scene. In summary,

the representation of canonical scenes enables a better summary of each place proposal and thus lead to improved scene analysis performance.

**Table 3.** Real-time video-based place analysis from a moving camera.

(a) Sample frame.



(b) Precision and recall rates.

	Baseline		Proposed	
	First Video	Second Video	First Video	Second Video
Recall %	50	46	100	54
Precision %	100	75	100	87.5

## 5.4 Real-Time Video Processing

We also consider real-time video processing from a moving camera that follows approximately a 100 meter route. The same route is traversed twice - at different times. The processing is done real-time during each movement. In the first time traversal, there are 12 place proposals generated. Sample frames from some of these places are shown in Fig. 3a. Actually, 2 of these places are revisits. Interestingly, with the proposed approach, both are recognized. This is in contrast to the baseline approach in which only one place is recognized. In the second video as obtained from another traversal of the same route, there are 13 place proposals. With the baseline approach, there are 6 true recognitions and 2 false ones. This results in 75% precision with 46% recall rates as shown in Table 3b. The performance is considerably better with the proposed approach. In this case, there are 7 true recognitions and 1 false one. Thus, recall increases to 54% while precision goes up 87.5%.

## 6 Conclusion

We present an unsupervised two-stage approach for generating place proposals reliably from unlabeled videos as acquired by a moving camera. First, each distinct place is delineated based on the continuous iterative partitioning of the incoming frames - considering their informativeness, coherency and plenitude. Following, canonical scenes in each detected place are determined based on the hierarchical clustering of the respective set of frames. The generated place proposals can then be used for video-based place analysis. Experimental results demonstrate place recognition performance to improve considerably when canonical scenes associated with each detected place are used. With indoors datasets, there is a boost of precision rates up to 23% and of recall rates up to 21% as compared to the baseline approach. Outdoor performance is comparable with the baseline approach since places tend to be less obstructed. We are currently working on integrating this model with semantic video segmentation.



**Acknowledgments.** The author acknowledges the contributions of Hakan Karaoguz and Berkan Höke in the experimental evaluations.

## References

1. Apostolidis, E., Mezaris, V.: Fast shot segmentation combining global and local visual descriptors. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6583–6587 (2014). <https://doi.org/10.1109/ICASSP.2014.6854873>
2. Baraldi, L., Grana, C., Cucchiara, R.: Shot and scene detection via hierarchical clustering for re-using broadcast video. In: Azzopardi, G., Petkov, N. (eds.) Computer Analysis of Images and Patterns. pp. 801–811 (2015)
3. Bormann, R., Jordan, F., Li, W., Hampp, J., Hägele, M.: Room segmentation: Survey, implementation, and analysis. In: IEEE Int'l Conf. on Rob. Aut. pp. 1019–1026. IEEE (2016)
4. Chen, S., Nie, X., Fan, D., Zhang, D., Bhat, V., Hamid, R.: Shot contrastive self-supervised learning for scene boundary detection. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9791–9800 (2021). <https://doi.org/10.1109/CVPR46437.2021.00967>
5. Chen, Z., Wang, X., Wang, J.e.a.: Csmb-vss: video scene segmentation with cosine similarity matrixn. *Multimed Tools Appl.* (2024)
6. Del Fabro, M., Böszörményi, L.: State-of-the-art and future challenges in video scene detection: a survey. *Multimedia Syst.* **19**, 427–454 (2013)
7. Demir, M., Bozma, H.I.: Automated place detection based on coherent segments. In: 2018 IEEE 12th International Conference on Semantic Computing (ICSC). pp. 71–76 (2018). <https://doi.org/10.1109/ICSC.2018.00019>
8. Erkent, O., Bozma, H.I.: Bubble Space and Place Representation in Topological Maps. *The Int. J. of Rob. Res.* **32**(6), 671–688 (2013)
9. Esteve Brotons, M.J., Carmona Blanco, J., Lucendo, F.J., García-Rodríguez, J.: Video scene segmentation based on triplet loss ranking. In: Rojas, I., Joya, G., Catala, A. (eds.) *Advances in Computational Intelligence*, pp. 302–315. Springer Nature Switzerland, Cham (2023)
10. Fraundorfer, F., Engels, C., Nister, D.: Topological mapping, localization and navigation using image collections. In: IEEE/RSJ Int. Conf. on Intel. Rob. and Sys. pp. 3872–3877 (2007)
11. Jang, J.W., Oh, I.K.: Performance evaluation of scene change detection algorithms. In: Fifth Asia-Pacific Conf. on Communication. vol. 2, pp. 841–844 vol.2 (1999)
12. Karaoguz, H., Bozma, H.I.: Reliable topological place detection in bubble space. In: IEEE Int. Conf. on Rob. Aut. pp. 697–702 (2014)
13. Karaoguz, H., Bozma, H.I.: An integrated model of autonomous topological spatial cognition. *Auton. Robot.* **40**(8), 1379–1402 (2016)
14. Korrapati, H., Mezouar, Y.: Vision-based sparse topological mapping. *Rob. and Auto. Systems* **62**(9), 1259–1270 (2014)
15. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: Boundary sensitive network for temporal action proposal generation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *Computer Vision - ECCV 2018*, pp. 3–21. Springer International Publishing, Cham (2018)
16. Lin, T., Zhang, H.J.: Automatic video scene extraction by shot grouping. In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. vol. 4, pp. 39–42 vol.4 (2000). <https://doi.org/10.1109/ICPR.2000.902860>
17. Liu, M., Colas, F., Pomerleau, F., Siegwart, R.: A Markov semi-supervised clustering approach and its application in topological map extraction. In: IEEE/RSJ Int. Conf. on Intel. Rob. and Sys. pp. 4743–4748 (2012)

18. Matsumoto, Y., Inaba, M., Inoue, H.: Visual navigation using view-sequenced route representation. In: IEEE Int. Conf. on Rob. Aut. pp. 83 – 88 (1996)
19. Miller, S.: Space and Sense. Psychology Press (2008)
20. Murphy, L., Sibley, G.: Incremental unsupervised topological place discovery. In: IEEE Int. Conf. Robot. Aut. pp. 1312 – 1318 (June 2014)
21. Nourani-Vatani, N., Borges, P.V.K., Roberts, J.M., Srinivasan, M.V.: On the use of optical flow for scene change detection and description. *J. of Intel. & Rob. Sys.* **74**(3–4), 817–846 (2014)
22. Paul, R., Feldman, D., Rus, D., Newman, P.: Visual precis generation using coresets. In: IEEE Int’l Conf. Rob. and Aut. pp. 1304–1311 (2014)
23. Pronobis, A., Caputo, B.: COLD: The COSY localization database. *The Int’l J. of Rob. Res.* **28**(5), 588–594 (2009)
24. Qingqiu, H., Yu, X., Anyi, R.: Movienet. <https://movienet.github.io/> (2020)
25. Ranganathan, A.: PLISS: detecting and labeling places using online change-point detection. In: *Rob.: Science and Systems* (2010)
26. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016)
27. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. *Neural Comput.* **13**(7), 1443–1471 (2001)
28. Sellami, A., Tabbone, S.: Video semantic segmentation using deep multi-view representation learning. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. pp. 1–7 (2021). <https://doi.org/10.1109/ICPR48806.2021.9413239>
29. Sibson, R.: SLINK: an optimally efficient algorithm for the single-link cluster method. *Comput. J.* **16**(1), 30–34 (1973)
30. Smith, M., Baldwin, I., Churchill, W., Paul, R., Newman, P.: The New College vision and laser data set. *The Int. J. Robot. Res.* **28**(5), 595–599 (2009)
31. Tapus, A., Siegwart, R.: Incremental robot mapping with fingerprints of places. In: *IEEE/RSJ Int’l Conf. IROS*. pp. 2429–2434 (2005)
32. Tomoya, M., Kanji, T.: Change detection under global viewpoint uncertainty. *arXiv preprint arXiv:1703.00552* (2017)
33. Topp, E.A., Christensen, H.I.: Detecting structural ambiguities and transitions during a guided tour. In: IEEE Int. Conf. Rob. Aut. pp. 2564–2570 (2008)
34. Wu, J., Rehg, J.M.: Centrist: A visual descriptor for scene categorization. *IEEE Trans. PAMI* **33**(8), 1489–1501 (2011)
35. Zhou, T., Porikli, F., Crandall, D.J., Van Gool, L., Wang, W.: A survey on deep learning technique for video segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(6), 7099–7122 (2023). <https://doi.org/10.1109/TPAMI.2022.3225573>



# Severity of Flood Damage Estimation from Aerial Scenery

Tarakeswara Rao Landa<sup>(✉)</sup> and Tushar Sandhan

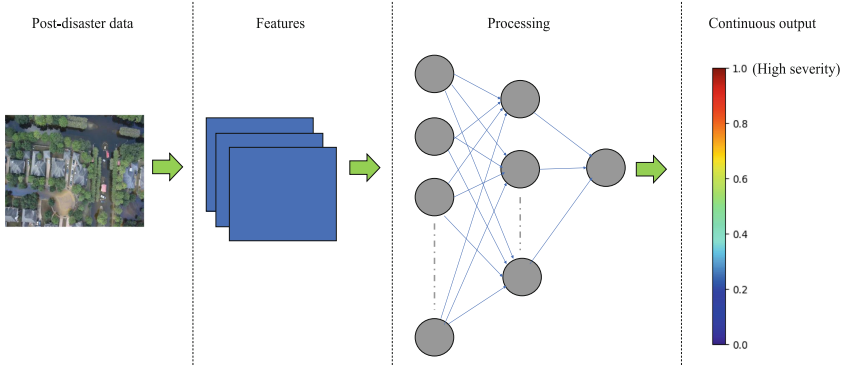
Perception and Intelligence Lab, Electrical Department, Indian Institute of Technology Kanpur, Kanpur, India  
{ltrao22,sandhan}@iitk.ac.in

**Abstract.** Accurate assessment of the flood damage and its severity estimation is essential for effective disaster management and related reconstruction works. In our study, we propose a novel approach for estimating the severity of flood damage from aerial scene images. We introduce a fusion network architecture that leverages both RGB and generated pseudo thermal image modalities. Our approach is based on training a CNN head as U-Net model to perform semantic segmentation of images from flood scenes. We show that the feature maps extracted from multimodal data, helps to improve accuracy even though one of the modalities is pseudo generated via CycleGAN. These feature maps are fed into a custom fully connected network for regression, predicting the severity level of flood damage. Our use of CycleGAN to generate thermal images from RGB images, providing additional input modalities for our network. Our approach significantly outperforms baseline methods, showcasing the effectiveness of leveraging multiple modalities for flood damage severity estimation. The results from our regression network show that our fused network outperforms conventional approaches. Our method achieves 0.053 MAE and 0.008 MSE, indicating a substantial enhancement of performance compared to baseline methods. These results show the importance of multimodal fusion via pseudo modality generation, which also offers valuable insights in flood damage assessment.

**Keywords:** Flood damage assessment · Semantic segmentation · Thermal image generation · Regression Analysis · Disaster management

## 1 Introduction

Natural disasters, worsened by climate change, pose significant threats to lives and infrastructure globally. Accurate and timely assessment of disaster damage is essential for efficient emergency response and recovery operations. Aerial imagery, particularly from unmanned aerial vehicles (UAVs), provides high-resolution data essential for assessing disaster impacts. Semantic segmentation is one deep learning technique that has shown promise in recent years for assessing disaster damage from aerial data.



**Fig. 1.** Our goal is to use exclusively post-disaster data to create a continuous number that represents the severity of flooding. The first column shows post-disaster data, which is the input used in our model. The U-Net segmentation network’s extracted features are shown in the second column. The regression network (third column) will use these features to create a continuous severity level (fourth column).

While existing research has focused on segmentation [1–3], classification [1, 3], and visual question answering [3] on aerial flood scene images, there is a lack of work addressing the flood damage severity level present in an image. Most authors have not explored this aspect, which is crucial for effective disaster response and management. Our work aims to fill this gap by focusing on semantic segmentation of aerial flood scene images and the ability to distinguish between different levels of flood severity using both RGB and pseudo-thermal modalities.

A dataset of flood scene images annotated with flood severity levels, a feature extraction technique to extract reliable features from post-disaster image data, and a regression network to condense these features into a single continuous value indicating flood severity are needed to meet these requirements. This idea is demonstrated in Fig. 1. The availability of publicly accessible datasets with satellite or ground-level imagery of hurricane-affected areas labeled with continuous values is limited. In order to overcome this difficulty, we labeled a portion of the FloodNet [3] dataset, indicating the level of flood damage severity in each instance.

In this study, we propose a novel approach for estimating flood severity on aerial flood scene images using deep learning techniques. We leverage the FloodNet [3] dataset, which offers high-resolution UAV imagery captured after Hurricane Harvey, along with detailed pixel-level annotations for various classes including flooded areas, buildings, roads, and more. Our approach involves training a U-Net [4] model for semantic segmentation to extract feature maps, followed by pseudo thermal modality generation from unpaired CycleGAN and finally a custom fully connected network for severity estimation based on these joint feature maps.

By combining these networks and utilizing both RGB and thermal imagery, we demonstrate significant improvements in flood severity estimation accuracy. Our approach not only enhances the understanding of flood-affected areas but also provides valuable insights for disaster response teams to effectively manage operations during emergencies.

Our work has made the following primary contributions:

1. A unique method that combines semantic segmentation and regression approaches to estimate the severity of flood damage on aerial scenes.
2. A demonstration of significant improvements in flood severity estimation accuracy by combining features from pseudo thermal and RGB images.
3. The use of a CycleGAN [5] to generate pseudo thermal images from RGB images, thereby providing latent feature dynamic information to basic segmentation network providing additional input modalities for our network.
4. Annotated a portion of the FloodNet [3] dataset, indicating the level of flood damage severity in each instance, to train a regression network for severity estimation.

The format of this paper is as follows: An overview of related work on image regression and semantic segmentation in the context of disaster damage assessment is given in Sect. 2. The methodology for assessing flood damage severity, including the model architecture, training process, and data preparation, is covered comprehensively in Sect. 3. In Sect. 4, we compare our technique with current methods and give experimental data and performance metrics. The ablation studies are discussed in detail in Sect. 5. The work is finally concluded in Sect. 6 with a discussion of potential future methods for this field of study.

## 2 Related Work

Damage assessment and detection are well-researched topics that have been studied in a variety of study fields. In the field of computer vision, techniques like segmentation and classification are essential for creating systems that efficiently analyze damage. Image regression, another crucial technique, predicts continuous values from image datasets. In disaster estimation, these methods, particularly semantic segmentation and image regression, play pivotal roles in accurately assessing damage severity and predicting outcomes.

### 2.1 Semantic segmentation

A crucial task in computer vision is called semantic segmentation, which is giving a label to every pixel in an image so that those pixels that have the same label are associated with the same object class. In recent years, it has been increasingly used in disaster estimation. Several methods were proposed by researchers for qualitative as well as quantitative improvement of semantic segmentation for various tasks. The Fully Convolutional Network [6] is a groundbreaking effort

that was followed by several cutting-edge models to handle semantic segmentation.

The PSPNet [7] is a novel approach that harnesses the power of spatial pyramid pooling to assimilate contextual information from various scales, thereby setting new standards in the realm of semantic segmentation. On a different note, DeepLab [8] merges the strengths of deep convolutional networks, atrous convolution, and fully connected conditional random fields. This fusion results in a robust framework for semantic image segmentation with a high degree of precision. U-Net [4] and SegNet [9], while both employing an encoder-decoder architecture for semantic segmentation, have distinct features. U-Net stands out with its symmetric pathways between the encoder and decoder, augmented by skip connections. SegNet, however, takes a unique approach by leveraging the pooling indices from the encoder phase to guide the upsampling during the decoder phase.

In the work of Doshi et al. [10], they leverage the power of semantic segmentation, specifically the Residual Inception Skip Network method proposed by Doshi [11], to analyze satellite images. Their goal is to discern alterations in the architecture of various human-made structures. This approach aids in pinpointing areas that would be most affected by natural disasters. Sahil et al. [1] presented a semi-supervised method for semantic segmentation and classification to address the challenges of damage assessment with limited labeled data. In [12], Rudner et al. combine multisensor, multitemporal, and multiresolution satellite images and present a unique method called Multi3Net for quick segmentation of flooded buildings. Rahnemoonfar et al. [13] introduced a hybrid network integrating densely connected Convolutional Neural Network (CNN) and Recurrent Neural Network for precise semantic segmentation of object boundaries in flooded UAV aerial images.

Using a UAV system and deep learning, Yang et al. [14] developed a flood detection method that achieved high accuracy in identifying flooded areas in a UAV dataset of flood-affected areas. Based on the Mask R-CNN deep learning model, Pi et al. suggested in [15] a disaster damage detection and semantic segmentation strategy employing UAV imagery. The suggested method demonstrated remarkable accuracy in identifying and classifying damaged objects when tested on two real-world disaster datasets. Gupta et al. [16] introduced RescueNet, a unified model designed for end-to-end training to simultaneously segment buildings and assess damage levels in post-disaster scenarios using satellite imagery.

Asad et al. [2] applied a Transformer-based approach to perform semantic segmentation on UAV images to assess damage caused by natural disasters. Safavi et al. [17] conducted a comparative analysis of real-time semantic segmentation networks applied to UAV imagery during flood conditions.

## 2.2 Image regression

Regression tasks are applied for image datasets in applications that require predicting a set of continuous values from the image. Image regression finds applica-

tions in various fields such as predicting human age, house prices, and estimating disasters.

In [18], Pereira et al. use social media photos to distinguish between images that provide clear proof of a flood and images that estimate the flood’s severity. The goal of the authors in [19] is to improve post-disaster management by identifying several hurricane categories using the estimation of tropical cyclone intensity. They use wind speed data from the HURDAT2 database and infrared satellite imaging data to estimate hurricane intensity using an enhanced deep CNN model.

Papatheofanous et al. [20] introduced an image regression module using image processing and CNNs to estimate solar irradiance, addressing Photovoltaic(PV) power production variability. Their sun localization-based method shows potential for real-time control in smart PV parks. Nia et al. [21] presented a deep learning model for building damage assessment using post-disaster data, with three neural networks for feature extraction and a regressor for severity estimation.

### 3 Methodology

We propose a novel approach for estimating flood damage severity on aerial flood scene images, leveraging a combination of semantic segmentation and regression techniques.

#### 3.1 Data preprocessing

**Semantic segmentation.** We used the Floodnet [3] dataset to train the U-Net [4] for the semantic segmentation task. It is a dataset collected using small UAVs after Hurricane Harvey. This dataset’s images have very high spatial resolution. The dataset contains 2,343 images paired with their respective masks, annotated with 9 classes: Road flooded, Road non-flooded, Grass, Vehicle, Tree, Water, Building flooded, Building non-flooded, and Pool. There are three sets of images in the dataset: 481 images for validation, 422 images for testing, and 1,440 images for training. We reduced the images to 416 x 320 dimensions during training due to memory constraints. The model was trained for 120 epochs with a batch size of 16.

We initiated the training phase by employing the Adam optimizer, with an initial learning rate (LR) of 0.001, and applied the reduced LR on the Plateau condition, which lowered the LR by 0.1 if the validation loss did not improve after 40 epochs. To save the optimal model weights depending on the validation loss, we also used the Model Checkpoint callback. We employed mIoU as the evaluation metric for the semantic segmentation and categorical cross-entropy as the loss function.

**Pseudo-thermal data generation.** We utilized the Cycle Generative Adversarial Network (CycleGAN) [5] to generate pseudo thermal images from the RGB images. CycleGAN is a method known for its capability in unpaired image-to-image translation tasks. Its objective is to acquire the knowledge of a mapping function  $G : U \rightarrow V$  between an output image in a target domain (thermal images,  $V$ ) and an input image from a source domain (RGB images,  $U$ ) without the need for paired training examples. This approach was chosen due to its effectiveness in translating images between different domains, which is particularly useful in our scenario where paired training examples are not available. The idea behind this pseudo-thermal image generation is to transfer the knowledge from pseudo-generation task to U-Net segmenter for efficient latent feature extraction from original RGB images.

To train the CycleGAN, we utilized RGB images from multiple datasets [3, 22, 23], and thermal images from other datasets [22–26]. The use of multiple datasets was motivated by the diversity and richness of the data they provided, improving the trained model’s ability to perform well on new, unseen data. Using a 0.0002 learning rate and a batch size of 1, the CycleGAN was trained across 50 epochs. To balance the trade-off between training speed and model performance, these parameters were selected.

The CycleGAN utilizes cycle consistency, adversarial, and identity losses. The cycle consistency loss makes sure that when an image is translated from one domain to another and back again, the original image is retained. The generated images are guaranteed to resemble real photos in the target domain due to the adversarial loss. The color distribution between the input and output images is preserved by the identity loss.

The definitions of identity loss ( $\mathcal{L}_{\text{idt}}$ ), adversarial loss ( $\mathcal{L}_{\text{GAN}}$ ), and cycle consistency loss ( $\mathcal{L}_{\text{cyc}}$ ) are as follows:

$$\begin{aligned} \mathcal{L}_{\text{idt}}(G, F) &= \mathbb{E}_{v \sim p_{\text{data}}(v)} [\|F(v) - v\|_1] + \mathbb{E}_{u \sim p_{\text{data}}(u)} [\|G(u) - u\|_1] \\ \mathcal{L}_{\text{GAN}}(G, D_V, U, V) &= \mathbb{E}_{u \sim p_{\text{data}}(u)} [\log(1 - D_V(G(u)))] + \mathbb{E}_{v \sim p_{\text{data}}(v)} [\log D_V(v)] \\ \mathcal{L}_{\text{cyc}}(G, F) &= \mathbb{E}_{v \sim p_{\text{data}}(v)} [\|G(F(v)) - v\|_1] + \mathbb{E}_{u \sim p_{\text{data}}(u)} [\|F(G(u)) - u\|_1] \end{aligned} \quad (1)$$

CycleGAN aims at minimizing the following loss function:

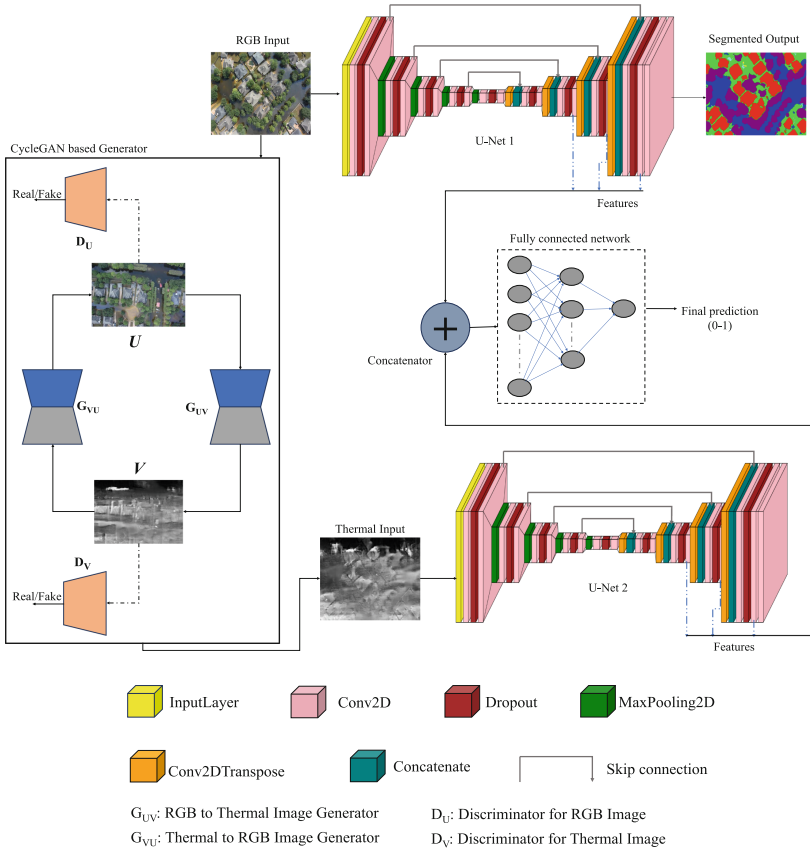
$$\begin{aligned} \mathcal{L}(G, F, D_U, D_V) &= \mathcal{L}_{\text{GAN}}(F, D_U, V, U) + \mathcal{L}_{\text{GAN}}(G, D_V, U, V) \\ &\quad + \lambda \mathcal{L}_{\text{cyc}}(G, F) + \mu \mathcal{L}_{\text{idt}}(G, F) \end{aligned} \quad (2)$$

Here,  $G$  and  $F$  are the mapping functions,  $D_V$  and  $D_U$  are the discriminators for  $V$  and  $U$  respectively,  $V$  and  $U$  are images from the target and source domains,  $\lambda$  and  $\mu$  are hyperparameters.

These loss functions assist CycleGAN in developing a reliable relationship between the RGB and thermal image domains, ensuring the generated thermal images are realistic and maintain structural similarity to the input RGB images.



Subsequently, we employed the trained CycleGAN network to generate thermal images corresponding to the RGB images, providing us with a comprehensive dataset for further analysis.



**Fig. 2.** An illustration of the Flood Damage Severity Architecture.

**Regression.** We created a new dataset of size 310 for the regression task using the Floodnet [3] dataset. We manually labeled ground truth values for 310 images. Non-flooded images were labeled as 0, while flooded images were divided into four severity levels: 0.25, 0.5, 0.75, and 1, focusing on buildings, roads, and vehicles affected by the flood event. The dataset consists of 56 images for testing, 64 for validation, and 190 for training. We used Mean Absolute Error as the evaluation metric and Mean Square Error as the loss function for the regression model. The model had 50 epochs of training with a batch size of 16. The dataset can be accessed through this link: <https://github.com/Tarakes796/Flood-damage-severity-estimation-dataset>.

### 3.2 Model architecture

The suggested model framework comprises three primary elements: U-Net segmentation networks for feature extraction, CycleGAN for pseudo-thermal modality generation, and a fully connected regression network for flood severity prediction. Fig. 2 illustrates the model architecture of our model.

**U-Net Segmentation Networks.** The first U-Net segmentation network (U-Net 1) is fed with RGB images, and features are extracted from the decoding layers with dimensions (BS, 80, 104, 64), (BS, 160, 208, 32), and (BS, 320, 416, 16) for each layer, where BS is the batch size. Global average pooling is then applied to each set of feature maps, resulting in pooled feature maps with dimensions (BS, 64), (BS, 32), and (BS, 16) respectively. The segmented output images from U-Net 1 (trained on the FloodNet dataset) provide detailed spatial information about the flood scene. Next, the RGB images are fed into a CycleGAN-based generator to convert them into thermal images. By doing this step, the model becomes more generalizable across various imaging modalities. The generated pseudo-thermal images are then processed by the second U-Net segmentation network (U-Net 2) which is exactly same as U-Net 1 model, and thermal features were extracted from the decoding layers with similar dimensions as the RGB features. RGB features and thermal features were saved separately.

**Feature concatenation.** RGB features and thermal features are concatenated to create fused features. This step combines spatial and thermal information, enhancing the model’s understanding of flood severity factors. The concatenated features form a single feature vector for each image, resulting in a final feature vector of dimensions (BS, 224).

**Regression network.** The concatenated features are fed into a custom fully connected neural network for regression. This network has three dense layers with 32, 16, and 1 neuron(s) each; the output layer has a sigmoid activation function, while the hidden layers have ReLU activation functions. The regression network is trained using the concatenated features as input and Mean Squared Error (MSE) as the loss function:

$$\text{MSE} = \frac{1}{m} \sum_{q=1}^m (y_q - \hat{y}_q)^2 \quad (3)$$

where  $y_q$  is the  $q_{th}$  sample’s actual value,  $\hat{y}_q$  is the  $q_{th}$  sample’s predicted value, and  $m$  is the number of samples in the dataset. MSE is useful in regression tasks as it penalizes larger errors more heavily, making it suitable for measuring the difference between predicted and actual continuous values.

The sigmoid activation function in the output layer of the regression network is defined as  $\sigma(r) = 1/(1 + e^{-r})$ . It maps the network’s output to a value between 0 and 1, allowing for the prediction of continuous values within this

range. This activation function is well-suited for tasks where the output needs to be interpreted as a probability or a continuous value within a specific range.

This model architecture leverages both semantic and thermal information to enhance the prediction of flood severity levels in aerial flood scene images.

## 4 Experiments and Results

### 4.1 Evaluation Metrics

**Mean Intersection over Union.** For evaluating the accuracy of object detection or segmentation models, a commonly used statistic is the mean Intersection over Union (mIoU), which measures the degree of overlap between the regions of interest in the ground truth and the predictions. It is given as,

$$\text{mIoU} = \frac{1}{N} \sum_{j=1}^N \frac{TP_j}{FN_j + FP_j + TP_j} \quad (4)$$

where TP is true positives, FN is false negatives, FP is false positives, and N is the total number of classes.

**Mean Absolute Error** Regression model performance is commonly evaluated using the Mean Absolute Error (MAE), which computes the average absolute difference between values that are predicted and those that are actual. Better performance is indicated by a lower MAE, while full consistency between predicted and actual values is shown by a value of 0. The MAE formula is:

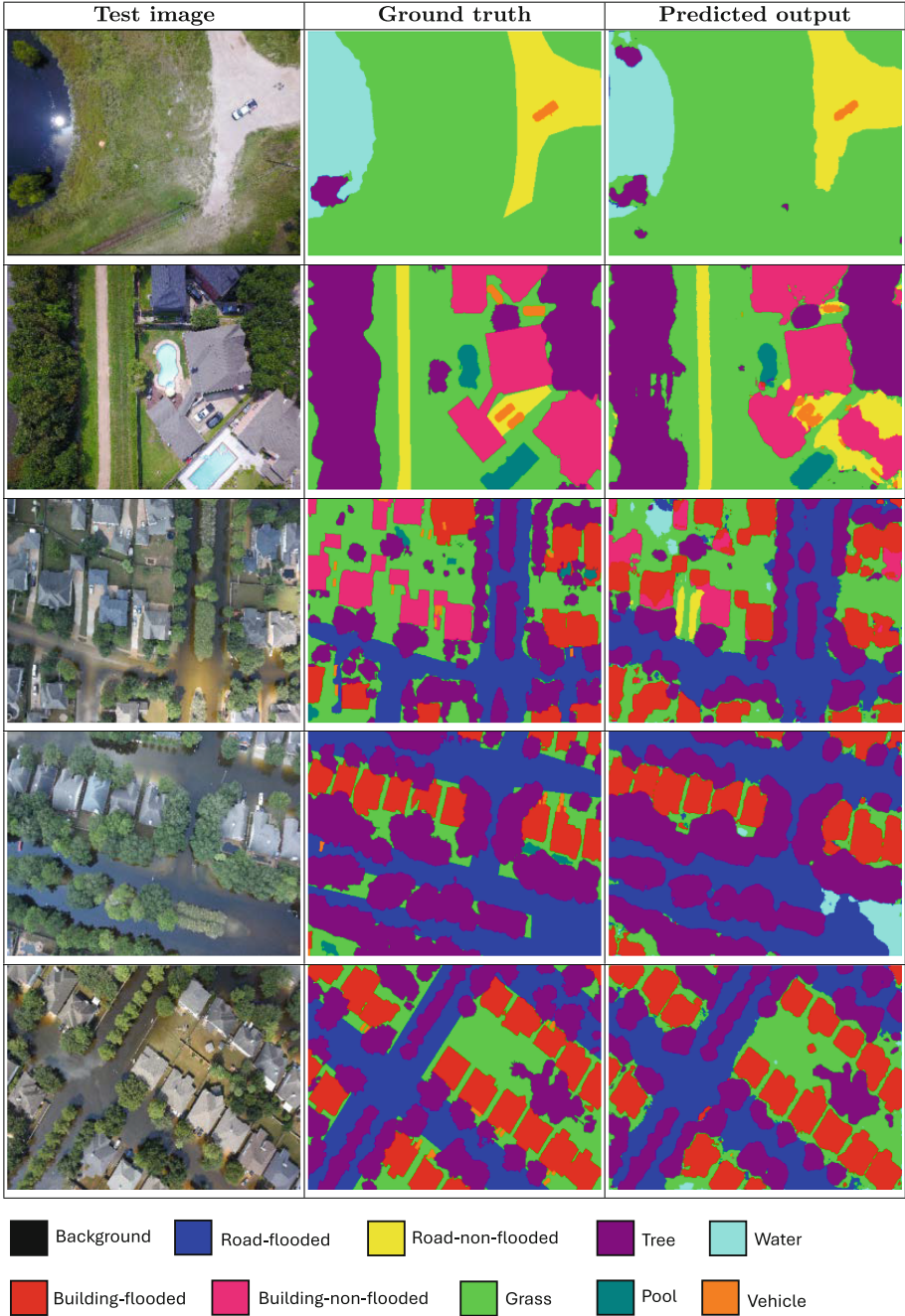
$$\text{MAE} = \frac{1}{m} \sum_{q=1}^m |y_q - \hat{y}_q| \quad (5)$$

where  $y_q$  is the  $q_{th}$  sample's actual value,  $\hat{y}_q$  is the  $q_{th}$  sample's predicted value, and  $m$  is the number of samples in the dataset.

### 4.2 Results on Test Set

**Semantic Segmentation Results.** After implementing the data processing and training settings described earlier, our semantic segmentation model achieved a mIoU score of 60.16% on the test dataset. This indicates the effectiveness of our approach in accurately segmenting the various classes in the Floodnet [3] dataset.

To visually demonstrate the performance of our semantic segmentation model, Fig. 3 showcases qualitative results. The figure illustrates how our model accurately identifies and segments different classes. These results visually confirm the model's capability to differentiate between different classes and accurately segment flood-related objects in high-resolution images.



**Fig. 3.** Semantic segmentation results of our re-trained U-Net base model and comparison with ground truths.

**Regression Model Results.** We adopted the model suggested by Nia et al. [21] as the reference model for our dataset, utilizing VGG16 [27] as the feature extractor to transmit features to the regression model. We assessed our model’s performance against several cutting-edge backbone models for feature extraction in the regression task. Table 1 presents a comparison of Mean Square Error (MSE) and Mean Absolute Error (MAE) between our proposed method and state-of-the-art methods, while Table 2 displays the comparison of the MSE and MAE for different backbone models.

**Table 1.** Comparison of MSE and MAE with State-of-the-Art Methods.

Model	MSE	MAE
Nia et al. [21]	0.023	0.095
<b>Ours</b>	<b>0.008</b>	<b>0.053</b>

**Table 2.** Comparison of MSE and MAE for different backbone models.

Backbone Model	MSE	MAE
ResNet50 [28]	0.010	0.059
MobileNetV3Small [29]	0.015	0.070
EfficientNetB0 [30]	0.014	0.063
DenseNet121 [31]	0.009	0.058
InceptionV3 [32]	0.125	0.205
Xception [33]	0.048	0.133
<b>Ours</b>	<b>0.008</b>	<b>0.053</b>

Our model achieved an impressive MSE of 0.008 and MAE of 0.053, outperforming all other models. Our fused network model demonstrates superior performance compared to both the baseline VGG16 [27] model (Nia et al. [21]) and other backbone models. Notably, the DenseNet121 [31] backbone performs well individually, but our fused approach using simple U-Net architecture still outperforms it. Several factors contribute to the superior performance of our model. One of the key aspects is the use of thermal images, which possess valuable distinguishing characteristics. Specifically, they highlight warm objects such as humans, animals, and hot vehicles, hot buildings which are typically the focus of attention in disaster management scenarios. These thermal features provide additional information that is not available in RGB images, thereby enhancing the model’s capability to estimate flood damage severity.

In our methodology, we used a U-Net model (U-Net 2) that is fed by the thermal images to extract these thermal features. This U-Net 2 acts as an efficiency booster for our overall network, leading to an increase in performance.

This highlights the effectiveness of combining features from multiple sources for more accurate regression in flood damage severity estimation. Fig. 4 shows the testing input samples and their corresponding predicted labels produced by our model.

## 5 Ablation studies

Our study on flood damage severity prediction using multimodal aerial imagery involved several key experiments and analyses. Here’s a structured overview of our work based on the experimental flow:

### 5.1 Experiment 1: RGB Images

Trained a U-Net [4] using the FloodNet [3] dataset for semantic segmentation. Created a dataset of 310 images for regression. Extracted feature maps for RGB images using the trained U-Net. Fed the RGB feature maps to a custom fully connected network for flood damage severity prediction. Evaluated performance using MSE and MAE.

### 5.2 Experiment 2: Pseudo-Thermal Images

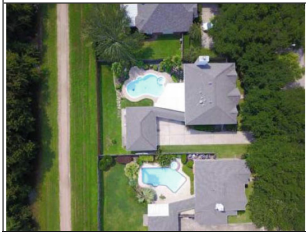




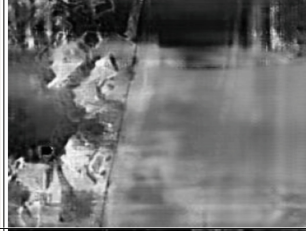
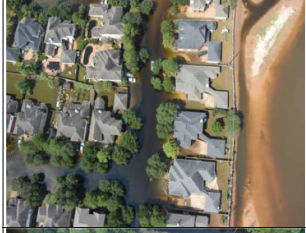



Generated pseudo-thermal images from RGB images using the CycleGAN [5]. Extracted feature maps for pseudo-thermal images using the trained U-Net [4]. Fed the pseudo-thermal feature maps to a custom fully connected network for flood damage severity prediction. Evaluated performance using MSE and MAE.

### 5.3 Experiment 3: Fused (RGB + Pseudo-Thermal) Images

Concatenated feature maps of RGB and pseudo-thermal images. Fed the fused feature maps to a custom fully connected network for flood damage severity prediction. Evaluated performance using MSE and MAE.

**Results and Analysis:** Table 3 contains the performance comparison of Mean Squared Error (MSE) and Mean Absolute Error (MAE) for flood damage severity prediction using different input modalities (RGB, Pseudo-thermal, and the fused (RGB + Pseudo-thermal)).

The fused results demonstrate the significance of our proposed model architecture in effectively leveraging multimodal information for improved flood damage severity prediction in aerial imagery. They also show the importance of combining different data modalities for more robust predictions in disaster management scenarios.

RGB Image	Thermal Image	True Label	Predicted Label
		0	0.01
		0.25	0.32
		0.5	0.57
		0.75	0.77
		1	0.93

**Fig. 4.** Illustration of testing input samples and predicted labels by our model. The first two columns represent the inputs to our model, the true label is displayed in the third column, and the estimated damage severity level is shown in the final column.

**Table 3.** Performance comparison of Mean Squared Error (MSE) and Mean Absolute Error (MAE) for flood damage severity prediction using different input modalities: RGB, Pseudo-thermal, and the fused (RGB + Pseudo-thermal) approach.

Input data	MSE	MAE
RGB	0.039	0.100
Pseudo-thermal	0.056	0.156
<b>Fused(RGB + Pseudo-thermal)</b>	<b>0.008</b>	<b>0.053</b>

## 6 Conclusion

In our work, we proposed a novel method for estimating flood damage severity on aerial flood scene images using a combination of semantic segmentation and regression techniques. Leveraging the FloodNet [3] dataset, we trained a U-Net [4] model for semantic segmentation, generated pseudo-thermal modality via CycleGAN [5], and a custom fully connected network for regression, achieving significant improvements in accuracy compared to existing methods.

Our semantic segmentation model attained a mIoU score of 60.16% on the test dataset, demonstrating its effectiveness in accurately segmenting flooded areas, buildings, roads, and other objects. Additionally, our regression model outperformed state-of-the-art methods, achieving an impressive MSE of 0.008 and a MAE 0.053.

By combining features from RGB and thermal images, our approach provides valuable insights into flood damage severity, aiding disaster response teams in managing operations during emergencies. Our work not only enhances the understanding of flood-affected areas but also demonstrates the potential of deep learning and fusion techniques in disaster damage assessment.

Future work could explore further improvements in accuracy by incorporating additional features or by refining the training process. Additionally, extending this approach to other types of natural disasters and integrating real-time data could enhance its applicability in disaster management scenarios.

**Acknowledgements.** This work is supported by IIT Kanpur C3iHub I Hub Foundation, a.k.a C3iHub under the aegis of DST.

## References

1. Sahil Khose, Abhiraj Tiwari, and Ankita Ghosh. Semi-supervised classification and segmentation on high resolution aerial images. arXiv preprint [arXiv:2105.08655](https://arxiv.org/abs/2105.08655), 2021
2. Muhammad Haroon Asad, Malik Muhammad Asim, Muhammad Naeem Muntaz Awan, and Muhammad Haroon Yousaf. Natural disaster damage assessment using semantic segmentation of uav imagery. In *2023 International Conference on Robotics and Automation in Industry (ICRAI)*, pages 1–7. IEEE, 2023



3. Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access*, 9:89644–89654, 2021
4. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 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*, pages 234–241. Springer, 2015
5. Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017
6. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015
7. Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017
8. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017
9. 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)
10. Jigar Doshi, Saikat Basu, and Guan Pang. From satellite imagery to disaster insights. arXiv preprint [arXiv:1812.07033](https://arxiv.org/abs/1812.07033), 2018
11. Jigar Doshi. Residual inception skip network for binary segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 216–219, 2018
12. Tim GJ Rudner, Marc Rußwurm, Jakub Fil, Ramona Pelich, Benjamin Bischke, Veronika Kopačková, and Piotr Biliński. Multi3net: segmenting flooded buildings via fusion of multiresolution, multisensor, and multitemporal satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 702–709, 2019
13. Maryam Rahnemoonfar, Robin Murphy, Marina Vicens Miquel, Dugan Dobbs, and Ashton Adams. Flooded area detection from uav images based on densely connected recurrent neural networks. In *IGARSS 2018-2018 IEEE international geoscience and remote sensing symposium*, pages 1788–1791. IEEE, 2018
14. Kaixin Yang, Sujie Zhang, Xinran Yang, Nan Wu, et al. Flood detection based on unmanned aerial vehicle system and deep learning. *Complexity*, 2022, 2022
15. Yalong Pi, Nipun D Nath, and Amir H Behzadan. Detection and semantic segmentation of disaster damage in uav footage. *Journal of Computing in Civil Engineering*, 35(2):04020063, 2021
16. Rohit Gupta and Mubarak Shah. Rescuenet: Joint building segmentation and damage assessment from satellite imagery. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4405–4411. IEEE, 2021
17. Safavi, F., Rahnemoonfar, M.: Comparative study of real-time semantic segmentation networks in aerial images during flooding events. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **16**, 15–31 (2022)

18. Pereira, J., Monteiro, J., Silva, J., Estima, J., Martins, B.: Assessing flood severity from crowdsourced social media photos with deep neural networks. *Multimedia Tools and Applications* **79**, 26197–26223 (2020)
19. Jayanthi Devaraj, Sumathi Ganesan, Rajvikram Madurai Elavarasan, and Umashankar Subramaniam. A novel deep learning based model for tropical intensity estimation and post-disaster management of hurricanes. *Applied Sciences*, 11(9):4129, 2021
20. Elissaios Alexios Papatheofanous, Vasileios Kalekis, Georgios Venitourakis, Filippos Tziolos, and Dionysios Reisis. Deep learning-based image regression for short-term solar irradiance forecasting on the edge. *Electronics*, 11(22):3794, 2022
21. Karoon Rashedi Nia and Greg Mori. Building damage assessment using deep learning and ground-level image data. In *2017 14th conference on computer and robot vision (CRV)*, pages 95–102. IEEE, 2017
22. Qishen Ha, Kohei Watanabe, Takumi Karasawa, Yoshitaka Ushiku, and Tatsuya Harada. Mfnet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5108–5115. IEEE, 2017
23. Sun, Y., Zuo, W., Liu, M.: RTFNet: RGB-Thermal Fusion Network for Semantic Segmentation of Urban Scenes. *IEEE Robotics and Automation Letters* **4**(3), 2576–2583 (July2019)
24. Suo, J., Wang, T., Zhang, X., Chen, H., Zhou, W., Shi, W.: Hit-uav: A high-altitude infrared thermal dataset for unmanned aerial vehicle-based object detection. *Scientific Data* **10**(1), 227 (2023)
25. Haitao Xiong, Wenjie Cai, and Qiong Liu. Mcnet: Multi-level correction network for thermal image semantic segmentation of nighttime driving scene. *Infrared Physics & Technology*, page 103628, 2021
26. Li, C., Xia, W., Yan, Y., Luo, B., Tang, J.: Segmenting objects in day and night: Edge-conditioned cnn for thermal image semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems* **32**(7), 3069–3082 (2020)
27. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556), 2014
28. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016
29. Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019
30. Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019
31. Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017
32. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016
33. François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017



# Spatio-Temporal Attentive Fusion Unit for Effective Video Prediction

Binit Singh, Divij Singh, Rohan Kaushal, Sana Vishnu Karthikeya Reddy, Bandam Sai Jaswanth, and Pratik Chattopadhyay<sup>(✉)</sup>

Indian Institute of Technology (BHU), Varanasi, India  
{binit Singh.cse21, divij.singh.cse20, rohan.kaushal.cse20,  
svishnu.karthikeyar.cse21, bandam.saijaswanth.cse21,  
pratik.cse}@iitbhu.ac.in

**Abstract.** In recent years, Recurrent Neural Networks (RNNs) have been extensively used for video frame prediction. The use of attention in recurrent networks helps in better memory utilization and to date, only a few recent video frame prediction models have used attention to capture long-term motion information. Still, these methods fail to preserve the structural consistency resulting in blurry output and fading out of smaller objects. We propose a new RNN-based spatio-temporal prediction unit with attention termed Spatio-Temporal Attentive Fusion Unit (STAFU) that combines temporal motion information and spatial appearance information respectively through a temporal attention unit and a spatial attention unit to preserve long-term sequence information at high resolution. The outputs from the two attention units are next aggregated through a hybrid aggregation unit with a wide receptive field for both the spatial and temporal features, which causes high-quality video prediction. The above units are embedded within a GAN framework that is trained in an end-to-end fashion. Our approach has been evaluated on three public datasets, namely Moving-MNIST, KTH-Action, and ETHZ, and interesting results have been obtained. A comparative study with other recent models shows that, on average, our model performs better and more consistently than the others in terms of the different metrics, namely MSE, MAE, SSIM, PSNR, and LPIPS.

**Keywords:** Spatio-Temporal Attentive Fusion Unit · Recurrent Model · Hybrid Aggregation · Video Frame Prediction

## 1 Introduction

Video frame prediction deals with the prediction of future frames in a video given a few previous frames. The topic has gained increasing attention in recent years due to its wide range of applications such as robotics, autonomous vehicles, person recognition, and surveillance. For instance, in robotics video prediction can be used to plan future trajectories and avoid obstacles whereas in autonomous vehicles video prediction can aid in predicting the behavior of other vehicles on

the road and prevent collisions. In surveillance, video prediction can be used to anticipate potential threats and provide early warning signals. Video prediction can help gain insights into the underlying structure and dynamics of a video sequence through which the complex interactions between different objects and elements in the video can be better understood, which is useful for tasks such as object tracking and recognition.

Recurrent Deep learning-based models for video prediction and related applications [25–27] have achieved impressive results in recent years. However, due to a smaller temporal receptive field, RNN units cannot retain long-term inter-frame motion information. To mitigate this problem, a few approaches use 3D Convolutional Neural Networks (CNNs) to widen the temporal receptive field, e.g., [26]. However, the increase in kernel size leads to high computational load. Further, a few methods enhance the conventional Mean Squared Error (MSE) loss to generate future frames, while others use Generative Adversarial Network (GAN) based methods, e.g., [11], [9] for better-generalized predictions with sharper visual quality. Most existing methods suffer from heavy computational overhead. The work in [3] uses an attention module that pays different levels of attention to the past temporal states, effectively widening the temporal receptive field while keeping a low computation load. However, it fails to generalize well in case of small inter-frame changes and overfits the training data due to mainly MSE loss. Existing methods of video prediction are either computationally intensive or fail to make long-term predictions effectively. In this paper, we improve upon the existing approaches in these aspects and make the following contributions:

- We propose a new GAN-based framework to train a video frame prediction model with dynamic adversarial loss for sharper frame reconstruction. Our GAN-based framework leverages dynamic adversarial loss to enhance frame sharpness, reducing blurriness over extended sequences. This results in more accurate and visually coherent long-term predictions compared to existing methods.
- We develop a new spatial attention module to capture localized attention between the highest motion areas in the current and previous frames.
- We develop a new hybrid aggregation module for an effective aggregation of the spatial and temporal states using two fusion strategies.

## 2 Related Work

The approach by Barbaeizadeh et al. [2] first explores the applicability of stochasticity in video prediction. In [5], a latent variable is constructed at every time step from observing frames with the help of deterministic estimation, followed by maximizing the likelihood of prior distributions. Villegas et al. [24] used a convolutional LSTM to minimize inductive bias without reducing the network capacity. Lee and Zhang [11] enclosed an adversarial loss inside a stochastic framework to maintain the naturalness of future frames. Shrivastava et al. [19]

proposed a model using the Gaussian Process to learn prior frames by conserving a probability distribution over observed frames. Akan et al. [1] further inherited the latent variables as static and developed an explicit motion model that remembers observed information’s static and dynamic values by maintaining appearance and motion distributions.

The use of memory cells enhances the ability of Deep Learning modules to anticipate future observations from historical events. In [15], Michalski et al. developed a pyramid structure of recurrent networks by stacking multiple gated autoencoders to represent the syntax of time series, which showed better generalization results compared to standard RNNs. Later in the same year, Srivastava et al. [20] defined an LSTM-based encoder-decoder predictive model that encodes the extracted percepts of video sequences into a fixed embedded representation and learns to decode probable future frames from that encoding. The ConvLSTM [17], an improved version of FC-LSTM [20], adapted the convolutional structures in the different state layers forming an end-to-end encoding structure. This work was further extended to develop TrajectoryGRU [18] which can learn the location-variant structure of recurrent cells with the help of Gated Recurrent Units. Another work, namely [8], adopted the hierarchical architecture of neural networks with 2D RNNs to avert the compounding errors in the pixel-level recursive prediction, producing high-level pixel spaces.

The above-mentioned approaches focus on retaining the motion information (inter-frame dependencies), neglecting spatial intra-frame information preservation like appearance features. In [27] by Wang et al., intra-frame temporal and spatial features are captured through a unified memory cell. Wang et al. further extended this work [25] by maintaining a quick alternative route for gradient flows, thereby capturing long-range information. Later, Wang et al. [26] integrated a 3D Convolution unit with Recurrent networks while maintaining a gate-controlled unit, named E3D-LSTM, that showed better information preservation for both short-term and long-term features. Another improved version of [27] is presented in [28] that retains long-range features through the introduction of a decouple loss in ST-LSTM cells. Moreover, to improve the local and global dynamics of long-range videos, SA-CLSTM [14] was proposed by Lin et al. by coupling a Self Attention Mechanism layer with standard Convolutional LSTM. Lee et al. [13] extended the bottleneck of typical RNNs in long temporal dependencies by introducing a Long Term Context (LTC) memory. To capture the inter-frame motion information of video prediction, a Motion Aware Unit (MAU) [4] was proposed where attention and fusion modules are used to learn an attention map between the present and historic frames. An Augmented Motion Information aggregates the information accumulated at the attention map and forwards it to the fusion module for final prediction. An improved version of MAU [4], i.e., STAU was introduced in [3] where not only the spatial information can supervise the temporal information in the temporal domain, but the temporal information can also supervise the spatial information in the spatial domain. The work in [23] combines spatial and temporal downsampling to effectively predict abstract representations such as human poses or locations over long

time horizons, while still maintaining a competitive performance for video frame prediction. In another work, namely [30], an end-to-end trainable two-stream video prediction framework is introduced termed Motion-Matrix-based Video Prediction (MMVP) to perform video prediction seamlessly while maintaining appearance consistency across the frames. A new recurrent cell termed SwinLSTM is presented in [22] that replaces the convolutional structure in ConvLSTM with the self-attention mechanism of SwinTransformer.

After extensive study of the previous works, it seems most of these are unable to extract useful information from the limited information field and suffer from over-fitting towards recent information only. Although the work in [4] can effectively extract past information and fuse this motion information with the current spatial appearance information using attention mechanisms, it fails to maintain structural consistency and the smaller motion entities tend to gradually get eliminated from the predictions. The approach in [3] is also unable to extract useful information from spatio-temporal modules at a high resolution. In this work, we consider solving the above problem by integrating attention mechanisms within an RNN framework to address this issue. The STAFU aims to enhance the memory capabilities of RNNs by employing a temporal attention module to capture the long-term motion information through a temporal attention unit that broadens the temporal receptive field without increasing computational load, a spatial attention module to preserve the structural consistency and finer details by focusing on high-motion areas within frames through a spatial attention unit and a hybrid aggregation module that fuses the spatial and temporal features to maintain pixel consistency and prevent the fading away of smaller objects. By embedding these units within a GAN framework, the proposed method seeks to leverage the strengths of RNNs while mitigating their limitations, resulting in improved video frame prediction quality.

### 3 Proposed Method

We aim to develop a Generative Adversarial Network (GAN)-based frame predictor that utilizes information from previous consecutive  $t$  frames, denoted by  $F_1, F_2, \dots, F_t$ , to predict the  $(t + 1)^{th}$  frame (which is either missing or occluded), denoted by  $\widehat{F}_{t+1}$ . Let  $\mathcal{P}$  denote the batch size at a particular instant of time while training the model, and  $\theta$  denote the model trainable parameters. If  $\widehat{F}_{t+1}^b$  and  $F_{t+1}^b$  respectively represent the predicted and ground truth for the  $(t + 1)^{th}$  frame corresponding to the  $b^{th}$  pattern of the batch, then after the batch training phase, the optimized network free-parameters  $\theta^*$  are obtained as  $\theta^* = \arg \min_{\theta} \sum_{b=1}^{\mathcal{P}} \|F_{t+1}^b - \widehat{F}_{t+1}^b\|^2$  through back-propagation based on ADAM optimizer.  $\theta^*$  gets fine-tuned as the batch training process continues and the final optimal set of values for  $\theta^*$  is obtained once training is done for all the batches. The overall framework of our proposed network is illustrated using Fig. 1. It follows a GAN-based architecture with a generator ( $\mathcal{G}$ ) and a discriminator ( $\mathcal{D}$ ). The generator takes as input a frame  $F_t$  and predicts the immediately succeeding frame  $F_{t+1}$ , which we denote as  $\widehat{F}_{t+1}$ . This prediction is carried out by encoding

the frame  $F_t$  through an Encoder that consists of a set of convolutional layers and fusing it with a set of previous spatial and temporal states using a recurrent network to obtain a prediction about the next frame in an encoded form, and finally using a Decoder to transform the predicted encoding back to the image space. The recurrent network used here consists of  $L$  layers of *STAFU* that effectively fuse the spatial and temporal memory states by preserving structural as well as long-term motion information. The skip connections between the convolutional and corresponding de-convolutional layers help to recover fine-grained structural details in the prediction. The model framework also consists of a discriminator network, where the output frame from the generator network is fed along with the ground-truth frame separately. The discriminator network thus produces two encoded outputs that are further used for adversarial loss computation.

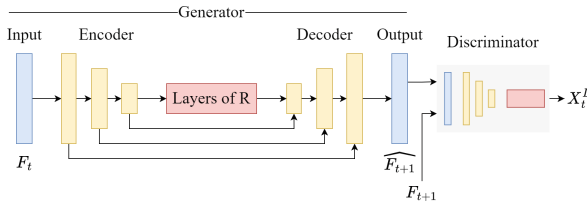


Fig. 1. Overall framework of the proposed GAN-based architecture

### 3.1 Generator

The generator network consists of an encoder network  $E$ , a decoder network  $D$  (both of which are based on CNNs) and a set of  $L$  Spatio-Temporal Attentive Fusion Units (*STAFUs*) ( $R^1, R^2, \dots, R^L$ ), as shown in Fig. 2. A spatial and temporal memory is maintained and updated using the outputs from *STAFU* (to be discussed in Section 3.1) at each layer at every time step by various gating mechanisms. The encoder part of the generator takes as input one frame at a time (say, frame  $F_t$ ) and produces an encoded spatial state of the frame denoted

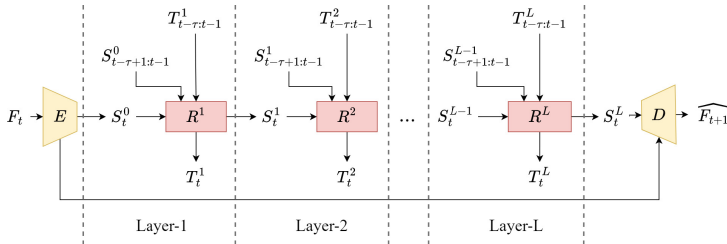
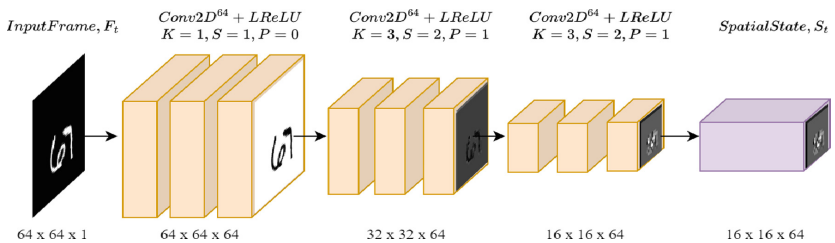


Fig. 2. Overall framework of the proposed generator module

by  $S_t^0$  at the initial layer (i.e., Layer-1 with reference to Fig. 2). This spatial state is given as input to a recurrent unit denoted as  $R^1$  in the figure, along with the previous  $\tau$  spatial and temporal states to compute self-attention, denoted as  $S_{t-\tau+1:t-1}$  and  $T_{t-\tau:t-1}$ , respectively. The outputs from  $R^1$  are matrices  $S_t^1$  and  $T_t^1$ , out of which  $T_t^1$  is updated and appended in the temporal memory to be used for the next time step, i.e., while predicting  $F_{t+2}$  from  $F_{t+1}$ . On the other hand,  $S_t^1$  is used by the *STAFU* in the next layer (denoted as  $R^2$  in the figure) and also appended to the spatial memory. The contents of the spatial memory corresponding to the previous  $\tau$  time steps of the preceding layer (denoted by  $S_{t-\tau+1:t-1}^1$  in Fig. 2) along with the previous  $\tau$  temporal states of the current layer available in the temporal memory (denoted by  $T_{t-\tau:t-1}^2$  in Fig. 2) are also input to the  $R^2$  unit. Similar to  $R^1$ , the *STAFU*  $R^2$  also generates two outputs, namely  $T_t^2$  which is saved in temporal memory and  $S_t^2$  which is used as input to *STAFU*  $R^3$  and also saved in the spatial memory for further use. The spatial and temporal states for all the  $L$  layers of *STAFUs* are computed similarly.

**Encoder** The encoder network is a fully convolutional network that accepts one frame at a time (say,  $F_t$ ) and outputs an encoded spatial state (say,  $S_t$ ). Let us assume that the frame  $F_t$  with dimensions  $(H, W, C)$  is input to the encoder and the encoded spatial state  $S_t$  output by this network has dimensions  $(H', W', C')$ . Here,  $C$  and  $C'$  denote the number of input and output channels respectively, whereas  $H$  and  $W$  correspond to the image height and width before the encoding step, and  $H'$  and  $W'$  denote the image width after the encoding step. Usually  $H' < H$ ,  $W' < W$ , and  $C' > C$ . The layer-wise architecture of the encoder is shown in Fig. 3. This figure also depicts the layer-wise transformation of an input mono-channel frame of dimensions  $64 \times 64 \times 1$  to the final encoded spatial state of dimensions  $16 \times 16 \times 64$ . The symbols  $K$ ,  $S$ , and  $P$  in the figure respectively denote Kernel size, stride, and padding.

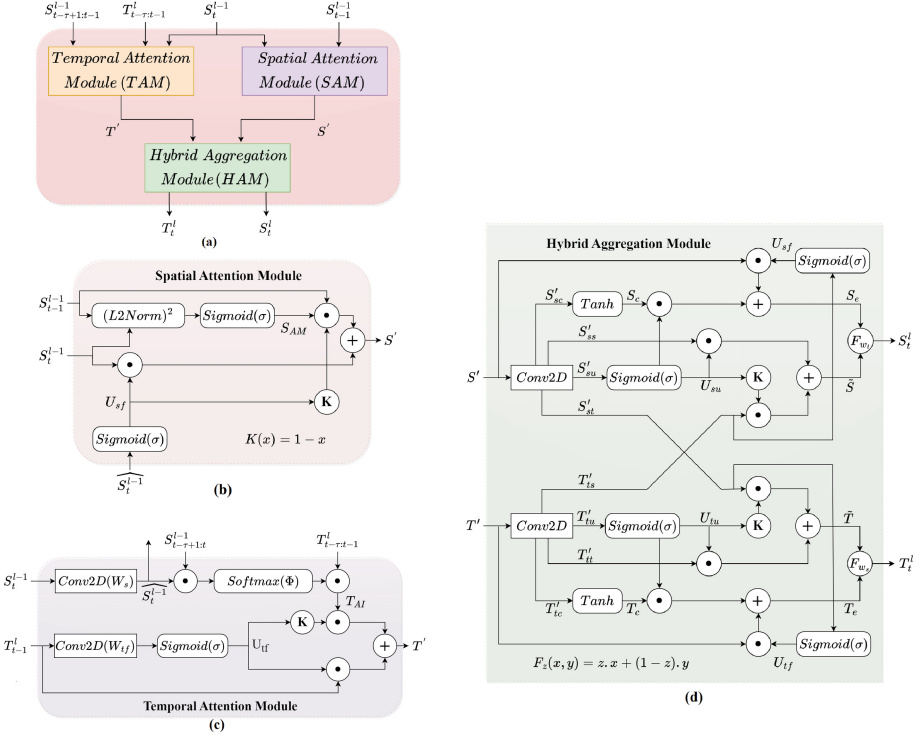


**Fig. 3. Graphic model of the encoder unit** displaying all the Convolution layers along with their parameters used for feature extraction and frame compression. The negative slope coefficient for LeakyReLU is set to 0.2.

**Spatio-Temporal Aware Unit** As explained using Fig. 2, the generator consists of  $L$  layers of our proposed *STAFU*. Each of these units has an identical



architecture and consists of (i) a spatial attention module, (ii) a temporal attention module, and (iii) a novel hybrid aggregator that combines information from the two attention modules, as shown in Figs. 4(a)-(d).



**Fig. 4.** (a) Spatio-Temporal Attentive Fusion Unit (*STAFU*) showing the abstract recurrent unit architecture, (b) Spatial Attention Module, (c) Temporal Attention Module, (d) Hybrid Aggregation Module for combining enriched spatial and temporal information to produce the spatial and temporal states at the current layer for the current time step

Spatial Attention Unit: The architecture of the spatial attention unit of the proposed *STAFU* is shown in Fig. 4(b). At a particular layer  $l$ , the spatial state  $S_t^{l-1}$  from the encoder (or, from the previous layer) is input to this unit within the *STAFU*  $R^l$  along with a spatial state from the previous time step denoted by  $S_{t-1}^{l-1}$ . At each time step, the squared Euclidean distance between the current and previous spatial states is computed which is further passed through sigmoidal activation ( $\sigma$ ) to generate a spatial attention map  $S_{AM}$  as follows:

$$S_{AM} = \sigma(\|S_t^{l-1} - S_{t-1}^{l-1}\|_2^2). \quad (1)$$

$S_{AM}$  is next aggregated with the previous spatial state  $S_{t-1}^{l-1}$  and current spatial state  $S_t^{l-1}$  using a spatial fusion gate  $U_{sf}$  to obtain a superior spatial state

representation  $S'$ , as explained using the following expression:

$$S' = U_{sf} \odot S_t^{l-1} + (1 - U_{sf}) \odot S_{AM} \odot S_{t-1}^{l-1}. \quad (2)$$

Here,  $U_{sf} = \sigma(\widehat{S_t^{l-1}})$ , where  $\sigma$  is the sigmoidal activation and  $\widehat{S_t^{l-1}} = W_{sf} * S_t^{l-1}$ ,  $W_{sf}$  being the weights of the convolutional kernel. The proposed spatial attention unit mainly focuses on specific parts of the frame that contain high-motion information. It also helps preserve smaller moving entities in the frame from fading away and maintains pixel consistency while concentrating on intrinsic intra-frame motion correlations.

**Temporal Attention Module:** The initial spatial state that is output from the encoder is input to this temporal attention module (TAM) which is another sub-module of the *STAFU*. The module has three more inputs which are initially zero matrices but are updated along the pipeline flow. These inputs are the temporal state from the previous time step and a limited set of previous  $\tau$  spatial and temporal states used for computing self-attention as observed in Fig. 4(c). These states are updated and controlled by the gating mechanisms. The module solves one of the most challenging tasks of broadening the temporal receptive field without increasing the kernel size for long-term information preservation and prediction. Let the set of previous temporal states from time step  $t-\tau$  to  $t-1$  for layer  $l$  be denoted by  $T_{t-\tau:t-1}^l$ . Further, let us use  $\tau$  to refer to the number of past temporal states,  $*$  and  $\odot$  to denote the convolution and Hadamard operator respectively, and  $T_{AI}$  to denote the attention information for long-term temporal information. Then,  $T_{AI}$  is computed as:  $T_{AI} = \sum_{i=1}^{\tau} \alpha_i T_{t-i}^l$ , where,  $\alpha_i$  is the attention score for the temporal state  $T_i$  and is computed as  $\alpha_i = \Phi(\sum_{j=1}^{\tau} S_{t-j+1}^{l-1} \odot \widehat{S_t^{l-1}})$ . Here,  $\Phi$  denotes the softmax function and  $\widehat{S_t^{l-1}} = W_s * S_t^{l-1}$ .  $T_{AI}$  is accumulated with the preceding temporal state using a temporal fusion gate  $U_{tf}$  to obtain a superior temporal state  $T'$ , given by:

$$T' = U_{tf} \odot T_{t-1}^l + (1 - U_{tf}) \odot T_{AI}, \text{ where } U_{tf} = \sigma(W_{tf} * T_{t-1}^l). \quad (3)$$

**Hybrid Aggregation Module:** This module effectively fuses the enhanced spatial and temporal states, namely  $S'$  and  $T'$  obtained respectively by applying (2) and (3), thereby exploiting the broadened receptive field. Unlike the existing method [4] which only fuses the motion information from the aggregated temporal state and the appearance information from the spatial state, we employ an improved aggregation technique to fuse general appearance, motion information as well as spatial structural changes to preserve better information related to the motion of every moving entity in the frame with different levels of attention. The final spatial output from this module has a high level of spatial information that the discriminator network may effectively utilize to produce a much sharper output. Our proposed hybrid aggregation module is schematically represented using Fig. 4(d). With reference to the figure,  $S'$  and  $T'$  are fed into convolution layers where each input is convoluted with four sets of kernels of the same dimensions. One of the convolution operations is represented as  $S'_{sc} = S' * W_{sc}$ . Similarly,  $S'_{ss}$ ,  $S'_{su}$ ,  $S'_{st}$  and  $T'_{tc}$ ,  $T'_{tt}$ ,  $T'_{tu}$ ,  $T'_{ts}$  are generated by convolving with the respective

kernels namely  $W_{ss}$ ,  $W_{su}$ ,  $W_{st}$  and  $W_{tc}$ ,  $W_{tt}$ ,  $W_{tu}$ ,  $W_{ts}$  weights.  $W_{su}$  and  $W_{tu}$  are activated using a sigmoidal function to create update gates for respective spatial and temporal states represented in (4):

$$U_{su} = \sigma(S'_{su}) \text{ and } U_{tu} = \sigma(T'_{tu}), \quad (4)$$

which helps to control the information fusion ratios from  $S'_{ss}$ ,  $T'_{ts}$  and  $T'_{tt}$ ,  $S'_{st}$ . The respective fusions generate  $S_e$  and  $T_e$  which is the final output after the first level of aggregation. The fusion equations are given below (5):

$$\tilde{T} = U_{tu} \odot T'_{tt} + (1 - U_{tu}) \odot S'_{st} \text{ and } \tilde{S} = U_{su} \odot S'_{ss} + (1 - U_{su}) \odot T'_{ts}. \quad (5)$$

Here,  $\tilde{S}$  contains motion information fused into spatial information and  $\tilde{T}$  is an updated temporal state with attention information that is used for the next prediction.  $\tilde{S}$  can readily be decoded using the decoder to produce the next frame ( $F_{t+1}$ ). However, the pixel consistency and localized motion are not focused in these aggregation so we employ yet another aggregation technique which activates the  $S'_{sc}$  and  $T'_{tc}$  using the tanh function, as shown next:

$$T_c = \tanh(T'_{tc}) \text{ and } S_c = \tanh(S'_{sc}). \quad (6)$$

These intermediate outputs are controlled using a forget gate constructed using sigmoidal activation of  $T'_{ts}$  and  $S'_{st}$  (7) to generate estimated output states (denoted by  $S_e$  and  $T_e$  respectively) as described in (8),

$$U_{tf} = \sigma(S'_{st}) \text{ and } U_{sf} = \sigma(T'_{ts}). \quad (7)$$

$$T_e = U_{tf} \odot T' + T_c \odot U_{tu} \text{ and } S_e = U_{sf} \odot S' + S_c \odot U_{su}. \quad (8)$$

These estimations are merged with the corresponding outputs from aggregation-1 using an adaptive function with learnable parameters to finally generate the spatial and temporal state for the current layer and time step as shown next:

$$T_t^l = W_t \odot \tilde{T} + (1 - W_t) \odot T_e, \text{ and } S_t^l = W_s \odot \tilde{S} + (1 - W_s) \odot S_e. \quad (9)$$

**Decoder** The last layer of the *STAFU* produces the spatial state  $S_t^L$  that is input to the decoder network composed of transposed 2D CNNs to produce the predicted frame at the next time step  $\widehat{F_{t+1}}$  (refer to Fig. 1). The decoder architecture mirrors the encoder architecture, i.e., the transposed 2D convolutions are arranged in a way that exactly reflects the encoder network with double the number of filters such that a skip connection between each corresponding layer of encoder-decoder is possible. Mathematically,  $\widehat{F_{t+1}} = \text{Decoder}(S_t^L)$ . The decoder-generated frames and the ground truth are used to compute standard L1 loss and MSE loss denoted by  $L_{L1}$  and  $L_{MSE}$ , respectively.

### 3.2 Discriminator

The discriminator network that learns a function denoted by  $\mathcal{D}$ , is composed of the encoder and stacked layers of *STAFUs* (shown in Fig. 1). It takes any given frame ( $F_t$ ) and produces a concatenated spatial and temporal state information  $X_t^L$  which is used in adversarial loss function for comparing the real and generated spatial states (appearance) along with the temporal trend (motion). The following equations describe the generation of  $X_t^L$  from  $F_t$ :

$$S_t^L, T_t^L = \mathcal{D}(F_t) \text{ and } X_t^L = \text{Concat}(S_t^L, T_t^L). \quad (10)$$

Adversarial learning is employed to get sharper output frames. The adversarial loss function is given by:

$$\min_G \max_{\mathcal{D}} L_{Adv}(\mathcal{D}, G) = \mathbb{E}_{x \sim p_{true}(x)} [\log \mathcal{D}(x)] + \mathbb{E}_{z \sim p_{pred}(z)} [\log(1 - \mathcal{D}(G(z)))] \quad (11)$$

$$\text{where } x = F_t \text{ and } z = \begin{cases} F_{t-1}, \eta < \text{teacher forcing ratio} \\ \widehat{F_{t-1}}, \text{ otherwise.} \end{cases} \quad (12)$$

Here,  $F_{t-1}$  is the previous true frame and  $\widehat{F_{t-1}}$  is the previously generated frame. At the early stages of model training, the  $\beta$  value is kept much higher with a lower value of the  $\alpha$  parameter because the penalization for a blurry output is not required. The overall loss of the model is given by:

$$Loss_{model} = \alpha L_{Adv} + \beta L_{MSE} + \gamma L_{L1}, \quad (13)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are constants used to control the weightage of the different loss terms.

## 4 Experiments

All experiments have been carried out on a system having 96 GB RAM, an i9-18 core processor, and three GPUs: a Titan Xp with 12 GB RAM, 12 GB frame-buffer memory, and 256 MB BAR1 memory and two GeForce GTX1080Ti with 11 GB RAM, 11 GB frame-buffer memory, and 256 MB BAR1 memory. The following subsections describe the datasets used in the study and evaluation metrics, implementation details, experiments and analyses of results obtained, and also some cross-dataset experiments. Our STAFU has a total of 86M parameters. We use three datasets with sequential actions for evaluation.

- **Moving MNIST** [12]: This dataset contains two handwritten digits sampled from the static MNIST dataset that move with different velocities and bounce off the edges of the image. The digits pass through each other on collision. For the experiment, 7,000 sequences have been used for training and 3,000 sequences have been used for testing.

- **KTH Action** [16]: It contains six types of human actions: jogging, walking, running, waving, clapping, and boxing. These actions are performed by 25 people in four different scene settings. 5686 sequences from this data have been sampled for training and 2437 sequences are used for testing.
- **ETHZ** [6]: This pedestrian motion dataset is captured from a stereo rig mounted on a car, with a resolution of  $640 \times 480$ , and a frame rate of 13-14 FPS. The pedestrians’ sequences are cropped and resized to  $64 \times 64$ . For training and testing, 910 and 390 sequences are sampled from this data.

For each of the datasets, we consider 20 successive frames from each sequence, with each frame cropped (if required) and resized to dimensions  $64 \times 64$ . To determine an optimal value for  $\tau$ , we conduct an ablation study for varying values of  $\tau$  using a validation set and observe the MSE, MAE, and inference times for these different values. Corresponding results reported in Table 1 show that with an increase in  $\tau$ , both MSE and MAE get lowered implying the prediction accuracy increases. However, the inference time gradually gets higher. We consider  $\tau$  equal to 7 to be a good balance between prediction accuracy and inference time.

**Table 1.** Ablation study considering different values for  $\tau$  with MSE Loss only (10 frames  $\rightarrow$  10 frames) trained for 15 epochs

$\tau$	MSE↓	MAE↓	Inference Time↓
1	82.7	147.6	20.34
3	82.0	146.1	20.86
5	80.8	144.7	22.71
7	78.0	140.3	24.58
9	75.3	135.3	27.03

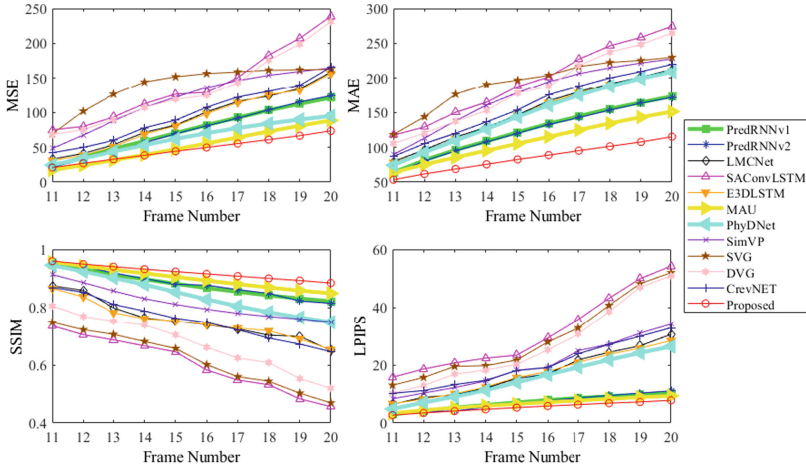
A comparative study is next carried out with several recent approaches, namely [3–5, 7, 10, 11, 13, 14, 22, 26–30]. For a fair comparison, each model is trained from scratch for 100 epochs using the same training set to predict a video frame from its previous 10 consecutive frames. Further, for each RNN-based prediction model, two recurrent layers have been used with 64 hidden units. The results for the compared methods are obtained using our experimental settings (i.e., training for 100 epochs and observing the metrics for future 10 predicted metrics) using the source code provided by the authors of the respective papers. The learning rate of our model has been set to  $2e^{-4}$  with a kernel size of  $5 \times 5$  and a stride of 1. Adam optimizer is employed for updating the model. A maximum batch size of 16 has been used to train all the models. During the first 70 epochs of training of our proposed model, the  $\alpha$  and  $\beta$  variables (refer to 13) are set to  $1e^{-6}$  and 1 and thereafter till the 100<sup>th</sup> epoch, these are interpolated with steps of 10x and 0.1x respectively till  $\alpha = 1e^{-3}$ . Table 2 shows comparative results for the trained models averaged over 10 frames for the Moving MNIST dataset and their inference times when run on the test data to

**Table 2.** Quantitative results of different models on the Moving MNIST dataset (10 frames  $\rightarrow$  10 frames). The scores are averaged over 10 frames.

Model	MSE $\downarrow$	MAE $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	Response Time (secs) $\downarrow$
PredRNN[27]	74.7	124.5	0.878	17.8	7.1	61.00
PredRNNv2[28]	73.7	122.7	0.881	18.0	7.0	51.60
E3D-LSTM[26]	89.6	149.7	0.752	17.7	16.9	30.50
SA-CLSTM[14]	139.6	195.7	0.603	14.9	31.3	37.40
LMCNet[13]	91.4	151.8	0.756	17.6	17.2	22.70
CrevNet[29]	98.4	159.5	0.755	16.5	19.9	24.10
PhyDNet[10]	63.4	147.9	0.843	18.8	15.4	21.40
SIM-VP[7]	119.3	174.4	0.814	15.6	19.9	17.60
MAU[4]	52.0	109.3	0.899	20.0	6.6	20.40
SVG[5]	75.7	197.0	0.859	18.3	7.5	20.60
STAU[3]	25.41	87.8	0.917	20.8	5.41	18.00
SAVP[11]	32.52	66.6	0.892	19.53	5.94	19.80
DVG[19]	139.5	192.4	0.618	14.8	29.1	34.00
SwinLSTM[22]	<b>17.7</b>	<b>55.5</b>	<b>0.955</b>	<b>38.9</b>	<b>3.2</b>	19.00
MMVP[30]	93.3	154.6	0.791	17.1	18.7	24.50
Proposed	46.9	84.9	0.920	21.7	5.4	<b>15.40</b>

predict the future 10 frames. The proposed model outperforms all the previous models, except [22] in all metric scores. The good performance is due to the high-quality spatiotemporal information extraction of earlier frames and effective fusion with the enhanced spatial state (appearance). The response time of our method is also the least among all the compared approaches, which verifies its computational effectiveness over the others.

The plots in Fig. 5 show per-frame metric plots for all metrics for all models in consideration. The y-axis denotes the respective metric and the x-axis shows the frame number. As observed from the plots, the metric scores worsen for longer predictions for each model. However, the worsening trend seems to be the least for the proposed model and it outperforms all the previous methods in all metrics. This is due to the effective attention mechanisms where only the most relevant information is retained from the past frames during the prediction. The comparative results for KTH Action and ETHZ datasets are shown in Table 3. In both cases, the proposed approach outperforms all the other compared methods. Unlike the observation from Table 2, for these datasets, our model performs better than SwinLSTM for most of the metrics. The superior performance of our model for RGB datasets (i.e., KTH Action and ETHZ) is mostly due to effectively capturing the richness of image features in the three channels of the RGB images by the proposed spatial and temporal attention modules in STAFU. For the ETHZ dataset, frame sequences of cropped pedestrians are taken and the frames are resized to  $64 \times 64$ . The dataset also contains static and dynamic



**Fig. 5.** Frame-wise metric plot for different metrics for per frame metric comparison for different methods on the Moving MNIST dataset

occlusion scenarios. The improved results for the proposed model are due to the effective fusion of the spatial pixel consistency with the temporal features, and further due to using the adversarial loss that makes the final prediction sharper. Fig. 6 shows a visual comparison of different frame prediction models using the KTH Action data. As observed from the image, the predictions of the proposed approach are not only in near-perfect synchronization with ground truth frames but also have maintained the perceptual quality. The superiority in the prediction quality of our model over the other compared models has been observed for the ETHZ data as well, but due to space constraints, we could not provide the comparative results here. Due to the effective utilization of the learned temporal dynamics, the proposed model can learn much faster and retain correct motion information for longer predictions. Further, the enhanced quality of the generated frames is due to the incorporation of spatial attention for pixel consistency and dynamic adversarial loss. It may be noted that among the datasets used in the study, KTH Action data has videos with simple pose variations and less background clutter. Hence, the obtained metrics are quite good for this dataset. Our proposed spatial and temporal attention modules retain better structural information for RGB videos than for binary videos. This is because, unlike RGB data, binary data lacks detailed gradient information that is necessary to capture complex structural information by our attention modules. Hence, the MSE and MAE for Moving MNIST data containing binary frames are to some extent high.

We also compare the results stated in the respective papers with the proposed method in Table 4 for the KTH dataset. Here, also we can see that the best results obtained using our STAFU are better than those reported in the compared approaches which is due to the effective fusion of spatial and temporal features.

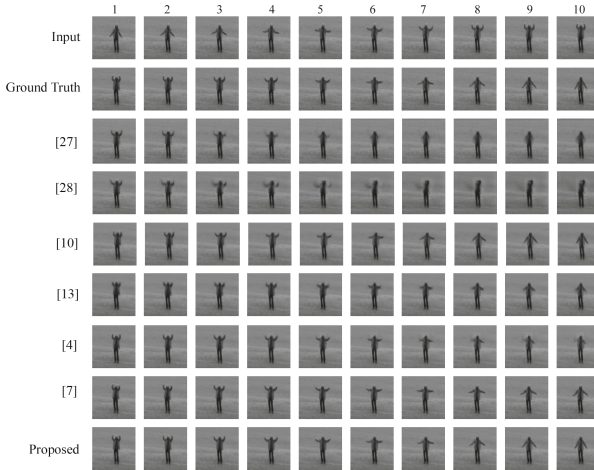
**Table 3.** Quantitative results of different models on the KTH Action and the ETHZ datasets (10 frames  $\rightarrow$  10 frames). Metrics are averaged over 10 frames.

Model	KTH Action					ETHZ				
	MSE↓	MAE↓	SSIM↑	PSNR↑	LPIPS↓	MSE↓	MAE↓	SSIM↑	PSNR↑	LPIPS↓
PredRNN[27]	8.04	78.8	0.948	27.0	6.39	40.9	245	0.613	20.00	38.3
PredRNNv2[28]	7.98	75.5	0.951	27.1	6.14	37.1	204	0.646	20.42	36.7
SA-CLSTM[14]	2.38	51.9	0.961	32.3	3.68	35.3	209	0.652	20.63	34.5
LMCNet[13]	1.71	43.7	0.960	33.7	3.62	43.9	252	0.542	19.69	38.8
CrevNet[29]	6.89	71.3	0.950	27.7	6.08	34.6	211	0.693	20.72	28.6
PhyDNet[10]	2.68	53.3	0.955	31.8	5.58	30.9	247	0.716	21.22	33.5
SIM-VP[7]	1.55	40.2	0.968	34.2	3.58	24.7	165	0.724	22.19	25.2
MAU[4]	1.00	39.1	0.977	36.1	3.47	20.9	168	0.763	22.90	24.1
SVG[5]	1.53	41.4	0.969	34.2	3.58	36.8	231	0.684	20.46	29.2
DVG[19]	1.55	33.65	0.970	34.1	3.59	27.8	204	0.724	21.67	27.0
SwinLSTM[22]	5.62	14.57	0.889	34.4	3.13	49.7	298	0.505	19.27	40.1
MMVP[30]	3.17	58.6	0.952	30.4	5.98	26.9	187	0.759	22.16	25.0
Proposed	<b>0.72</b>	<b>31.8</b>	<b>0.981</b>	<b>37.5</b>	<b>2.04</b>	<b>14.5</b>	<b>159</b>	<b>0.834</b>	<b>24.48</b>	<b>22.0</b>

The reasons behind the improved performance of STAFU when scaling from 10 $\rightarrow$ 10 to 10 $\rightarrow$ 20 on the KTH dataset are:

- *Effective Spatio-Temporal Attention Mechanisms:* The proposed method uses Spatio-Temporal Attentive Fusion Units (STAFU), which are designed to capture and integrate both spatial and temporal features effectively and combine these through a novel hybrid aggregation module. This dual attention mechanism ensures that the model maintains a high level of accuracy even when the prediction horizon is extended. Focusing on relevant spatial and temporal cues allows the model to better understand and predict future frames without losing coherence.
- *Robust Feature Representation:* The model’s architecture is robust to learning and representing features across different temporal scales. By efficiently encoding temporal dependencies and spatial structures, the model can generalize well even when the prediction task scales from 10 $\rightarrow$ 10 to 10 $\rightarrow$ 20. This robustness is particularly important in video prediction tasks where maintaining temporal consistency is crucial.
- *Attention to Temporal Dynamics:* The STAFU model’s emphasis on temporal attention allows it to capture long-term dependencies and motion patterns effectively. This capability is particularly beneficial when scaling the prediction horizon, as the model can leverage learned temporal dynamics to maintain performance over longer sequences.
- *Adaptation to Dataset Characteristics:* The KTH dataset, with its relatively simple and temporal human actions, suits the strengths of the STAFU model. The actions in KTH have consistent and predictable temporal patterns, which





**Fig. 6.** Comparison results for different methods on the KTH Action dataset

**Table 4.** Quantitative results of different models on the KTH dataset. (10 frames  $\rightarrow$  20 frames)

Model	SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$
PredRNN[27]	0.839	27.55	0.204
PredRNNv2[28]	0.838	28.37	0.139
SA-CLSTM[14]	0.712	23.58	0.231
LMCNet[13]	0.806	26.29	-
SIM-VP[7]	0.905	33.72	-
MMVP[30]	0.906	27.54	-
MSPred[23]	0.930	28.93	<b>0.032</b>
MOSO[21]	0.822	29.80	0.083
Proposed	<b>0.981</b>	<b>37.5</b>	0.204

**Table 5.** Ablation study to study the impact of the different loss terms during training using the MNIST data (10 frames $\rightarrow$ 10 frames)

Loss	MSE $\downarrow$	MAE $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$
MSE	84	144	0.84	17.3	12.8
L1	102	153	0.82	16.3	15.7
GAN	202	257	0.56	12.9	12.4
MSE+L1	91	143	0.83	16.9	12.5
GAN+L1	253	325	0.63	12.0	25.1
GAN+MSE	79	142	0.87	17.5	12.4
GAN+L1+MSE	<b>47</b>	<b>85</b>	<b>0.92</b>	<b>21.7</b>	<b>5.4</b>

the model’s spatio-temporal attention mechanisms can capture and predict accurately over extended horizons.

Other compared methods exhibit slight performance degradation possibly due to their sensitivity to the increased prediction horizon caused by a lack of effective mechanisms to capture long-term dependencies, leading to error accumulation and degradation in prediction accuracy.

Table 5 shows the model performance on training it with different loss settings. The adversarial loss is multiplied by  $1e^{-4}$  and L1 loss with 0.5 for all tests. The model was trained for 15 epochs on the Moving MNIST dataset with the same settings considered in our work. When the loss criterion is individually employed, we see that for the adversarial setting, the temporal and spatial dynamics are penalized for sharper frame generation which results in poor accuracy, while for MSE and L1 loss the accuracy is good. With the weighted com-

**Table 6.** Ablation study on the Moving MNIST dataset of different core submodules of the proposed model (10 frames→10 frames)

Modules	MSE↓	MAE↓	SSIM↑	PSNR↑	LPIPS↓
Temporal Attention	81.5	119	0.85	19.7	9.1
Spatial Attention	50.7	75	0.87	<b>21.7</b>	6.7
Our Hybrid Aggregator	<b>46.9</b>	<b>85</b>	<b>0.92</b>	<b>21.7</b>	<b>5.4</b>

bination of all three criteria can achieve the best results. The MSE loss leads to better learned motion dynamics and higher accuracy while adversarial loss along with L1 loss leads to better visual quality of output. To study the impact of the different components of the proposed model, we next conduct a similar training process by separately considering (i) the Temporal Attention module, (ii) the Spatial Attention module, and (iii) the proposed hybrid model. Corresponding results are reported in Table 6 using the Moving MNIST data. It can be seen from the results that the spatial and temporal attention modules alone are not effective enough for video prediction. However, the proposed hybrid aggregator improves over all the metrics compared to the individual attention modules, which emphasizes the superior frame prediction ability of STAFU.

## 5 Conclusions and Future Work

We propose a new recurrent architecture termed Spatio-Temporal Attentive Fusion Unit (*STAFU*) that combines spatial and temporal attention features from previous frames to predict future ones. In this work, the proposed STAFU is embedded within the generator of a GAN so that the spatio-temporal information can be exploited better through adversarial learning. The complete generator model typically comprises multiple STAFUs. The temporal attention mechanism in the STAFU is capable of perceiving long-term temporal information and effectively aggregating it with short-term dynamics. In contrast, the spatial attention module perceives the immediate distortion in the spatial states. The best estimated temporal state is next fused with the enhanced spatial state to generate the subsequent frame. Information skip has been employed between mirrored encoders and decoders to achieve high structural-quality outputs. Experimental results reveal that the proposed model achieves state-of-the-art performance on various occluded and occlusion-free spatiotemporal datasets. Experiments show that when trained for 100 epochs, our model always outperforms the other compared methods. Only in the case of Moving MNIST data, SwinLSTM has a slightly better performance than ours. In future, focus can be given to improving the performance of our model for videos with binary frames. A possible approach can be incorporating dilation at the convolutional layers of the Encoder to capture better structural features at different resolutions. Our model

has 86M parameters and in the future efforts may also be given to making our model lightweight through the application of knowledge distillation. Research on developing effective diffusion models for video frame prediction is another scope for future work.

**Acknowledgements.** The authors acknowledge SERB-DST, Government of India for supporting their work with a project grant (ref. no. CRG/2020/005465).

## References

1. Akan, A.K., Erdem, E., Erdem, A., Güney, F.: SLAMP: Stochastic Latent Appearance and Motion Prediction. In: Proc. of the IEEE/CVF ICCV. pp. 14728–14737 (2021)
2. Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R.H., Levine, S.: Stochastic Variational Video Prediction. arXiv preprint [arXiv:1710.11252](https://arxiv.org/abs/1710.11252) (2017)
3. Chang, Z., Zhang, X., Wang, S., Ma, S., Gao, W.: STAU: A SpatioTemporal-Aware Unit for Video Prediction and Beyond. arXiv (2022)
4. Chang, Z., Zhang, X., Wang, S., Ma, S., Ye, Y., Xiang, X., Gao, W.: MAU: A motion-aware unit for video prediction and beyond. In: Proc. of the Advances in NIPS (34). pp. 26950–26962 (2021)
5. Denton, E.L., Fergus, R.: Stochastic video generation with a learned prior. In: Proc. of the ICML. pp. 1174–1183 (2018)
6. Ess, A., Leibe, B., Van Gool, L.: Depth and Appearance for Mobile Scene Analysis. In: Proc. of the IEEE ICCV. pp. 1–8 (2007)
7. Gao, Z., Tan, C., Wu, L., Li, S.Z.: SimVP: Simpler Yet Better Video Prediction. In: Proc. of the IEEE/CVF CVPR. pp. 3170–3180 (2022)
8. Kim, T., Ahn, S., Bengio, Y.: Variational Temporal Abstraction. In: Proc. of the Advances in NIPS (32). pp. 11570–11579 (2019)
9. Kwon, Y.H., Park, M.G.: Predicting Future Frames Using Retrospective Cycle GAN. In: Proc. of the IEEE/CVF CVPR. pp. 1811–1820 (2019)
10. Le Guen, V., Thome, N.: Disentangling Physical Dynamics From Unknown Factors for Unsupervised Video Prediction. In: Proc. of the IEEE/CVF CVPR. pp. 11471–11481 (2020)
11. Lee, A., Zhang, R., Ebert, F., Abbeel, P., Finn, C., Levine, S.: Stochastic Adversarial Video Prediction. arXiv preprint [arXiv:1804.01523](https://arxiv.org/abs/1804.01523) (2018)
12. Lee, J., Lee, J., Lee, S., Yoon, S.: MSnet: Mutual Suppression Network for Disentangled Video Representations. CoRR [abs/1804.04810](https://arxiv.org/abs/1804.04810) (2018)
13. Lee, S., Kim, H.G., Choi, D.H., il Kim, H., Ro, Y.M.: Video Prediction Recalling Long-term Motion Context via Memory Alignment Learning. In: Proc. of the IEEE/CVF CVPR. pp. 3053–3062 (2021)
14. Lin, Z., Li, M., Zheng, Z., Cheng, Y., Yuan, C.: Self-Attention ConvLSTM for Spatiotemporal Prediction. In: Proc. of the AAAI Conf. on Artificial Intelligence. pp. 11531–11538 (2020)
15. Michalski, V., Memisevic, R., Konda, K.: Modeling Deep Temporal Dependencies with Recurrent Grammar Cells. In: Proc. of the Advances in NIPS (27). pp. 1925–1933 (2014)
16. Schudt, C., Laptev, I., Caputo, B.: Recognizing Human Actions: A Local SVM Approach. In: Proc. of the ICPR. pp. 32–36 Vol.3 (2004). <https://doi.org/10.1109/ICPR.2004.1334462>

17. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., WOO, W.C.: Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In: Proc. of the Advances in NIPS (28). pp. 802–810 (2015)
18. Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c.: Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model. In: Proc. of the Advances in NIPS (30). pp. 5617–5627 (2017)
19. Shrivastava, G., Shrivastava, A.: Diverse Video Generation using a Gaussian Process Trigger. In: Proc. of the ICLR (2021), [https://openreview.net/forum?id=Qm7R\\_SdqTpT](https://openreview.net/forum?id=Qm7R_SdqTpT)
20. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised Learning of Video Representations using LSTMs. In: Proc. of the ICML. pp. 843–852 (2015)
21. Sun, M., Wang, W., Zhu, X., Liu, J.: MOSO: Decomposing MOtion, Scene and Object for Video Prediction. In: Proc. of the IEEE/CVF CVPR. pp. 18727–18737 (2023)
22. Tang, S., Li, C., Zhang, P., Tang, R.: SwinLSTM: Improving Spatiotemporal Prediction Accuracy using Swin Transformer and LSTM. In: Proc. of the IEEE/CVF ICCV. pp. 13470–13479 (2023)
23. Villar-Corrales, A., Karapetyan, A., Boltres, A., Behnke, S.: MSPred: Video Prediction at Multiple Spatio-Temporal Scales with Hierarchical Recurrent Networks. arXiv preprint [arXiv:2203.09303](https://arxiv.org/abs/2203.09303) (2022)
24. Villegas, R., Pathak, A., Kannan, H., Erhan, D., Le, Q.V., Lee, H.: High Fidelity Video Prediction with Large Stochastic Recurrent Neural Networks. In: Proc. of the Advances in NIPS (32). pp. 81–91 (2019)
25. Wang, Y., Gao, Z., Long, M., Wang, J., Yu, P.S.: PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In: Proc. of the 35<sup>th</sup> ICML. pp. 5123–5132 (2018)
26. Wang, Y., Jiang, L., Yang, M.H., Li, L.J., Long, M., Fei-Fei, L.: Eidetic 3d LSTM: A model for video prediction and beyond. In: Proc. of the ICLR (2019)
27. Wang, Y., Long, M., Wang, J., Gao, Z., Yu, P.S.: PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs. In: Proc. of the Advances in NIPS (30). pp. 879–888 (2017)
28. Wang, Y., Wu, H., Zhang, J., Gao, Z., Wang, J., Philip, S.Y., Long, M.: PredRNN: A Recurrent Neural Network for Spatiotemporal Predictive Learning. IEEE Trans. on PAMI **45**(2), 2208–2225 (2022)
29. Yu, W., Lu, Y., Easterbrook, S., Fidler, S.: Efficient and Information-Preserving Future Frame Prediction and Beyond. In: Proc. of the Intl. Conf. on Learning Representations (2020)
30. Zhong, Y., Liang, L., Zharkov, I., Neumann, U.: MMVP: Motion-Matrix-Based Video Prediction. In: Proc. of the IEEE/CVF ICCV. pp. 4273–4283 (2023)



# Few-Shot View Synthesis Based on Geometric and Semantic Consistency

Mizuki Kojima<sup>(✉)</sup>, Rei Kawakami, and Masatoshi Okutomi

Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550, Japan  
kojima.m.ao@m.titech.ac.jp

**Abstract.** Neural Radiance Fields (NeRF) excels in generating realistic novel views of 3D scenes. However, generating these views based on few input views is challenging because of the insufficient data to recover the radiance field of the entire scene. To address this, we present a method for reconstructing NeRF from just three closely spaced input views, leveraging both geometric and semantic consistencies. Geometric consistency is ensured using a cost volume and variance evaluation across voxels, effectively reconstructing visible areas. For unseen areas, semantic consistency aligns semantic vectors between rendered and input images using pre-trained feature extractors. Combining these consistencies allows for precise reconstruction of both seen and unseen areas. Additionally, we enhance NeRF learning through entropy minimization for volume density regularization, black blending to eliminate floating artifacts, and relative learning-rate decay to facilitate learning of volume density. This multifaceted approach outperforms existing methods that rely on single consistency types, showing superior quantitative and qualitative results.

**Keywords:** 3D Reconstruction · Novel View Synthesis · NeRF · Few-shot

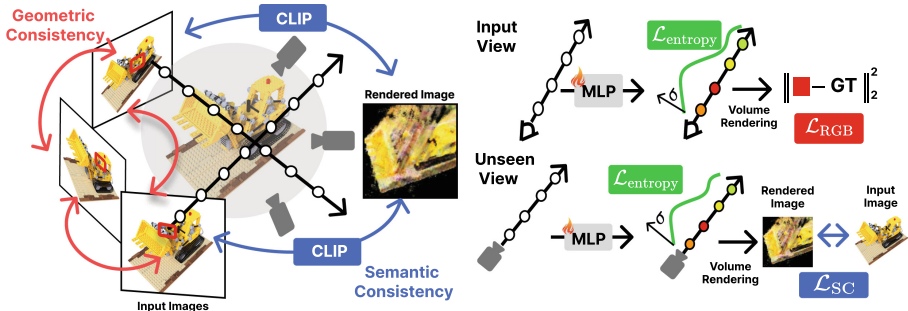
## 1 Introduction

3D scene reconstruction from multiple images is crucial for diverse applications such as navigation, robotics, architectural design, and entertainment. Powered by the progress of deep learning, neural 3D reconstruction, employing neural networks for 3D scene synthesis, has attracted attention because it can incorporate contextual information within a scene. Numerous studies are conducted to learn a good representation of 3D scenes; examples include estimating signed distance fields from a specific object [13,26], exploring representations through meshes [19,46], adopting methods to depict 3D spaces with discrete voxels [12,47], and representing neural light field [39]. Among them, Neural Radiance Fields (NeRF) [9] and gaussian splatting [7] stand out for generating novel

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78456-9\\_6](https://doi.org/10.1007/978-3-031-78456-9_6).

views of 3D scenes. NeRF uses neural network’s weights to implicitly capture a scene’s radiance field, while gaussian splatting employs multiple gaussians to model this field explicitly. Both methods are effective because they rely on a continuous function that simultaneously represents the structure and appearance of a scene. This allows for rendering from any camera position and orientation.



**Fig. 1. Overview:** Our method reconstructs a complete 3D scene from a few closely taken input images, enabling the generation of images from any viewpoint. High-accuracy restoration is achieved within the areas covered by the input images due to geometric consistency. Semantic consistency provides cues for recovering areas the input images does not depict. This collaboration ensures a comprehensive and seamless restoration, filling in the unseen spaces with coherent detail.

NeRF employs a differentiable neural network concerning its input. This inherent differentiability is critical in ensuring spatial coherence and supports the capability to produce high-resolution rendered images. However, there are several challenges associated with this method. Primarily, it demands a significant amount of training images for each scene reconstruction. Furthermore, it cannot reconstruct regions not present in the training data.

To relax the large number of images required for NeRF, geometric consistency inherent between images are utilized. PixelNeRF [6] and MVNeRF [1] incorporate geometry-aware features extracted from training images using pre-trained convolutional neural networks (CNNs). These features are added as input to NeRF, provide insights into the scene structure, guiding the network towards a more accurate reconstruction. Semantic consistency from images are also used to enhance 3D reconstruction. DietNeRF [3], for instance, leverages semantic embeddings extracted from images via CLIP [4] encoder to recover regions not present in the input images. A notable approach, DreamField [2] leverages semantic embeddings derived from captions utilizing the CLIP encoder for 3D reconstruction.

Even with geometric or semantic consistency, reconstructing an entire scene from just three closely spaced images is still difficult. As we will demonstrate in the experiments, semantic consistency alone is insufficient to reconstruct the

scene, while geometric consistency alone fails to restore the invisible areas. However, this problem setting is important, as it is highly realistic, with relevant applications such as three-camera smartphones and similar camera configurations for mobile robots.

In this paper, we propose a method that achieves the generation of novel viewpoint images from a full  $360^\circ$  range using only three nearby images, each reflecting a limited region of a nearby object. This is achieved by seamlessly integrating geometric and semantic consistencies. Each consistency does not contribute independently to separate regions; instead, their collaboration enhances 3D reconstruction across all areas. Fig. 1 provides a broad overview, while Fig. 2 offers a detailed explanation. Geometric consistency ensures precise recovery of regions illustrated by the three adjacent images. Semantic consistency provides essential cues for recovering areas this trio of images does not depict.

Specifically, we use a geometric feature vector from 3D-CNN that transfers knowledge from a cost volume derived from 2D-CNN image features. This ensures geometric fidelity as in MVSNeRF [1], and both networks produce key features for radiance field estimation. Semantically, we utilize CLIP encoder embeddings to reconstruct areas not captured by the three input images. Additionally, incorporating volume density regularization from InfoNeRF [31] eliminates artifacts and enhances object coherence.

Furthermore, we propose *black blending* and *relative learning-rate decay*. In black blending, the rendered image merges with a black background based on each pixel’s accumulated transmittance, effectively sharpening boundaries, removing artifacts, and improving coherent object estimation. A black background is essential because it does not interfere with the foreground color, preserving its semantic meaning. This has minimal negative impact when evaluated by the CLIP encoder, especially in which the presence of objects is ambiguous. With relative learning-rate decay, we reduce the learning rate of the MLP for color output compared to the MLP for density, leading to clearer images and accurate geometry estimation. This shows that in NeRF, learning color precedes learning geometry, quickly minimizing RGB loss and weakening the signals needed to estimate geometry.

In the experiments on the Realistic Synthetic NeRF dataset [9], we evaluated the quality of images ours and existing methods generated from new viewpoints using three metrics. We confirmed accuracy improvements of 3.44 in PSNR, 0.048 in SSIM, and a reduction of 0.124 in LPIPS compared to those that rely solely on single consistency.

Our contributions are summarized as follows:

1. A comprehensive method is proposed to reconstruct the radiance field of a scene *from only three closely-shot* images, *synergistically* combining geometric and semantic consistencies. The method also effectively incorporates entropy minimization, black blending, and relative learning-rate decay for performance improvement.

2. *Black blending* enhances rendered image quality and eliminates the negative impact of background color in regions where object presence is ambiguous, as demonstrated both numerically and visually.
3. *Relative learning-rate decay* is proposed to prioritize volume density over color, leading to improved geometric recovery as verified quantitatively.
4. Experiments on a NeRF dataset reveal performance improvements compared to relying solely on semantic or geometric consistency. The experiments also demonstrate the impact of each consistency as the synthesized viewpoint diverges from the input viewpoints, as well as the contribution of each loss for the quality of generated images.

## 2 Related Work

Various approaches are taken to enable 3D reconstruction with NeRF even with a small number of images. Below, we review methods that incorporate the benefits of large-scale pre-trained models into NeRF, methods that integrate geometric information, and various improvement techniques aimed at enhancing the accuracy of NeRF estimation.

**NeRF with Priors** Recent proposals have leveraged large pre-trained models as prior to distill a complete radiance field from just a few images. For instance, methods that improve the quality of NeRF by utilizing GANs [22] or incorporating the probability distribution of images learned by Diffusion Models (DMs) [24, 38, 49] have been introduced [10, 11, 17, 21, 25, 27, 30, 33, 35, 40, 43, 44, 50].

CLIP [4], trained on about 400 million pairs of captions and images through contrastive learning, can similarly be used as a large pre-trained model for prior knowledge. CLIPNeRF [14] uses semantic vectors extracted from CLIP, based on captions or reference images, to modify the color and shape of specific 3D objects. Blending-NeRF [20] proposes editing scenes to align with captions against pre-estimated radiance fields. DreamField [2] devises a method to estimate radiance fields that generate images aligned with text, using semantic vectors extracted from text by CLIP. DietNeRF extracts semantic vectors from input images to ensure that rendered images from new viewpoints have similar semantic vectors. As in those studies, we introduce semantic vectors from CLIP so that it aids recovery of unseen regions. However, estimating radiance fields with only semantic vectors is challenging.

**NeRF with Geometric Information** Many methods incorporate geometric information into NeRF to reduce the number of input images needed. Examples include using depth from images as geometric information and extracting geometric information through geometric procedures.

Various methods employ depth as geometric information in different ways. Techniques include using actual depth images, utilizing depth information estimated by pretrained models, and verifying the consistency between rendered images from estimated depth and input images. DSNeRF [29] uses real depth from input images, while SparseNeRF [18] and SCADE [36] use depth



images estimated by pretrained models for inferring radiance fields. DiffusioNeRF [27] leverages gradients of log-likelihood from RGBD image patches predicted by pretrained models. GeCoNeRF [32] and SinNeRF [16] check consistency through reprojecting depth information between reference and unseen viewpoints. Although RegNeRF [34] does not use depth information, it applies regularization to ensure minimal differences between adjacent depths.

Geometric procedures reduce the number of input images for estimating radiance fields by using geometric clues. PixelNeRF, GRF [5], and IBRNet [41] utilize features extracted from the pixels of input images corresponding to a specific 3D point as additional information. ReconFusion [44], similar to PixelNeRF, uses these extracted features as conditions for DM. SparseFusion [50] aggregates geometric information along epipolar lines corresponding to the rays being rendered. MVSNeRF acquires geometric information by creating a cost volume through plane sweep from input images. Using just geometric consistency makes it difficult to reconstruct the unknown regions. The seamless integration of geometric and semantic consistencies remains unexplored.

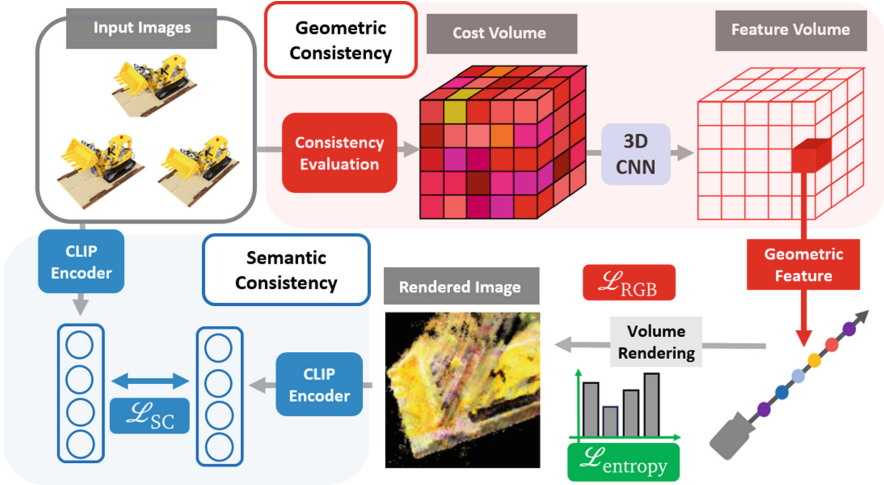
**NeRF with Various Improvement Techniques** Various methods are proposed to improve the accuracy of NeRF estimations. FreeNeRF [28] suppresses early catastrophic overfitting by learning from low to high-frequency components. Similarly, SimpleNeRF [37] reduces depth discontinuities by lowering the highest frequency of positional encoding and minimizing the impact of view-dependent radiance. RawNeRF [8] reduces surface ambiguity by encouraging a decrease in the second moment around the mean of the probability distribution (variance) of object surface locations along rays. Likewise, InfoNeRF [31] decreases entropy when an object is likely present along rays, considering a probability distribution of the objects' existence. The proposed black blending and relative learning-rate decay can be used concurrently with these techniques.

## 3 Method

### 3.1 Overview

We present a comprehensive method for efficiently estimating NeRF from three closely positioned images for novel view synthesis. The accurate geometry is obtained from input images by a NeRF equipped with stereo reconstruction techniques, while the radiance field of unseen regions are reconstructed by applying semantic consistency to them. We also incorporate an entropy minimization loss to reduce fog-like artifacts. In addition, we propose two novel techniques for estimating NeRF effectively: black blending and relative learning-rate decay.

The combination of geometric and semantic consistencies enables the reconstruction of the entire scene by complementing each other, rather than contributing independently. In areas slightly away from the input images, semantic consistency helps fill in the uncertain parts. Geometric consistency, reinforced by other consistencies, propagates from the visible area and provides clues for



**Fig. 2. Overview (detail):** Our method consists of three main components: (a) geometric consistency, (b) semantic consistency, and (c) ray entropy minimization loss. (a) Geometric consistency is achieved by incorporating geometric features into the input of the radiance estimating MLP, based on cost volumes derived from plane-sweep across multiple input images. (b) Semantic consistency involves comparing the similarity of semantic vectors obtained by feeding rendered and input images into the CLIP encoder to minimize their semantic difference. (c) Ray entropy minimization loss reduces the entropy of the probability distribution of volume density over rays, helping to eliminate floaters like white fog and enhancing a scene’s coherence.

accurate 3D reconstruction. Given that the overall geometry is highly unconstrained, entropy loss helps regularizing ambiguous geometry. As a result, the synergy of these consistencies contributes in all areas.

The overview of the method for a synergistic combination of those consistencies is shown in Fig. 2. Total loss  $\mathcal{L}_{total}$  is formulated as a linear combination of the standard loss  $\mathcal{L}_{RGB}$ , loss  $\mathcal{L}_{SC}$  ensuring geometric and semantic consistency, and ray entropy minimization loss  $\mathcal{L}_{entropy}$ :

$$\mathcal{L}_{total} = \mathcal{L}_{RGB} + \lambda_1 \mathcal{L}_{SC} + \lambda_2 \mathcal{L}_{entropy}. \quad (1)$$

Note that during training, RGB values for pixels are simultaneously estimated from rays corresponding to both the input image and unseen viewpoint image pixels. Ray entropy minimization loss is applied to both, whereas the RGB loss is applied only to the former and the semantic loss is applied only to the latter.

Moreover, we propose two novel techniques: black blending and relative learning-rate decay. These methods contribute to producing more precise and accurate estimations. The details of these proposals are in 3.3.

### 3.2 Geometric, Semantic, and Entropy Losses

**Geometric Consistency** 3D reconstruction is achieved by estimating the entire scene’s radiance field. This radiance field is represented by weight  $\Theta$  of a function  $f$ , which takes a scene’s position and direction as inputs and outputs the volume density and color at that point. In other words, for a given position  $x$  and direction  $d$ , it outputs the volume density  $\sigma$  and color  $c$ :

$$(\sigma, c) = f_{\Theta}(x, d, g, \{C_i^{\text{ref}}\}_{i=1}^3). \quad (2)$$

Unlike the basic NeRF [9], we follow MVSNerF [1] to add geometric feature  $g$  and the pixel values  $\{C_i^{\text{ref}}\}_{i=1}^3$  corresponding to a 3D point  $x$  in the scene from the input images. We detail the MVSNerF’s method for extracting geometric features in supplementary materials for readers.

NeRF aims to match the true pixel values of input images with the estimated pixel values via the radiance field. To decide the pixel RGB values, the estimated volume density and color at sampling points along the ray are integrated using volume rendering. For a ray, the estimated pixel value and the true pixel value lead to the loss function  $\mathcal{L}_{\text{RGB}}$ :

$$\mathcal{L}_{\text{RGB}} = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \|\hat{C}_r(\Theta) - C_r\|_2^2. \quad (3)$$

$\Theta$  are the parameter characterizing the function  $f$  representing the radiance field, updated through the estimation process.

**Semantic Consistency** As in other work [2, 3], in the quest to ensure semantic consistency between training and their rendered counterparts, we leverage CLIP, a pre-trained encoder, to formulate a loss function as described in the bottom row in Fig. 2. Given three semantically aligned input images,  $\{x_i\}_{i=1}^3$ , CLIP produces corresponding feature vectors,  $\{v_i\}_{i=1}^3$ . One is randomly chosen as  $v_{\text{GT}}$ . Simultaneously, images rendered during the training phase denoted as  $x_{\text{render}}$ , yield features termed  $v_{\text{render}}$ . The objective is to minimize the loss function  $\mathcal{L}_{\text{SC}}$  as:

$$\mathcal{L}_{\text{SC}} = -\langle v_{\text{GT}}, v_{\text{render}} \rangle. \quad (4)$$

This semantic loss thereby ensures that the rendered images retain the semantic essence of the originals.

**Ray Entropy Minimization** Ray entropy minimization loss is introduced in InfoNeRF [31], which ensures objects on a ray become coherent and their boundaries clear, also removing artifacts like fog. It specifically lowers the entropy of the probability distribution for volume density along a ray, based on the estimated volume density, resulting in objects that are more distinct and coherent. Our method incorporates this loss for further enhancing more accurate estimation of the neural radiance field. We describe it here briefly for self-completion.

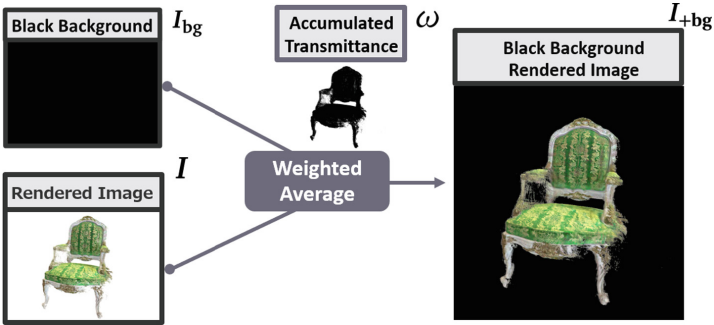
When denoting the probability distribution as  $p(x)$ , the entropy minimization loss  $\mathcal{L}_{\text{entropy}}$  is defined as follows:

$$\mathcal{L}_{\text{entropy}} = M \odot \mathbb{H}[p] = M \odot \left( - \sum_i p(x_i) \log p(x_i) \right). \quad (5)$$

$M$  is a mask set to 1 when an object is considered to exist and 0 otherwise.

### 3.3 Enhancing the Learning of NeRF

**Black Blending** Black blending, shown in Fig 3, averages the rendered image with a black background during training, weighted by the accumulated transmittance from volume density along the ray. The advantage of the black background is that it avoids any negative impact from the background color. In NeRF, the background color and the estimated RGB value are blended based on the object’s existence probability, which can cause problems in ambiguous areas. A black background, with all zero elements, prevents any adverse effects. As a result, this enhances object coherence and sharpens their outlines while clearing up floaters, which is especially effective for objects with ambiguous transparency. It applies only to rendered images during training for CLIP’s input.



**Fig. 3. Black Blending:** We propose a method that blends the rendered image with a black background, weighted by the accumulated transmittance. This technique eliminates floaters, sharpens object-scene boundaries, and ensures consistent object estimation. The accumulated transmittance, representing the likelihood of a ray encountering an object, is based on the volume density along the ray.

Specifically, Black blending combines a rendered image  $I$  with a black background  $I_{\text{bg}}$ , weighted by the accumulated transmittance  $\omega$  as a linear combination. Mathematically, it is expressed as:

$$I_{+\text{bg}} = \omega I_{\text{bg}} + (1 - \omega)I = (1 - \omega)I. \quad (6)$$

Here, the accumulated transmittance integrates the volume density information of sampling points along a ray, representing the probability of an object’s presence on the ray. This is calculated as:

$$\omega = \left\{ \sum_{i=1}^N (1 - \exp(\sigma_i \delta_i)) \right\}, \quad \text{where } \delta_i = t_{i+1} - t_i. \quad (7)$$

**Relative Learning-rate Decay** We propose relative learning-rate decay to promote accurate geometry estimation. Specifically, relative learning-rate decay slows down the learning rate of the MLP estimating color compared to the MLP estimating volume density. This technique increases the importance of accurately estimating volume density relative to color, thereby encouraging precise geometry estimation. By doing so, it helps reduce the risk of decreasing semantic loss based solely on the estimated color. We use two types of schedulers and optimizers: one for the weights up to the volume density and another for estimating the color. The learning rate for the color is lower than that for the volume density.

## 4 Experiments

### 4.1 Experimental Settings

**Data** We conducted a numerical evaluation using Realistic Synthetic NeRF dataset [9], which features diverse scenes such as a chair, hotdog, and ficus. The dataset has eight distinct scenes with 100 training images. We used three closely spaced images for training from there, in line with MVSNerF [1]. For testing, instead of the provided 200 test images, we used 100 validation images to avoid potential biases from the original test set’s viewpoints. The original test set’s bias, along with the locations of our test and training data, is visualized in the supplementary material’s “Positions of Train/Test Images”. This approach ensures clarity and unbiased research outcomes.

**Evaluation Metrics** We employed three evaluation metrics for our proposed method: peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [48], Learned Perceptual Image Patch Similarity (LPIPS) [45] following prior research. These metrics offer a comprehensive assessment of the images rendered by the respective methods.

**Backbone Architecture** Our model employs the same geometric consistency approach as MVSNerF, leveraging multi-view stereo principles to generate feature volume from three input images. The weights of 2D-CNN and 3D-CNN in this model remain fixed throughout the training process. 2D-CNN incorporates seven 2D convolutional layers for extracting image features. For the 3D-CNN, a U-net structure with ten 3D convolutional layers (seven for downsampling and three for upsampling) is employed to output a neural encoding volume within the reference view’s frustum. It is pre-trained on the DTU dataset [42]. We trained an MLP to estimate both radiance and volume density based on three types of inputs: a given position in 3D space, the extracted volume feature, and

pixel values of three input images. We did not include the view direction as an input to the MLP because view-dependent radiance complicates the estimation of the radiance field when the inputs are a small number of nearby images. This model incorporates a six-layer function, with positional encoding applied to the position vector at the preliminary stages.

**Implementation Details** The overall process follows Fig. 2. Our implementation is based on the MVSNerF codebase, using its feature extraction and RGB loss on input images. As MVSNerF’s training is based on ray sampling, we added the rendering of the images at novel views and semantic loss was added by feeding the rendered images into CLIP and measuring the semantic similarity to the inputs. At the volume rendering, the density along the ray is obtained for entropy loss in both input and novel views. For Black blending, we remove the term of blending with the background, as presented in Eq. (6). In Relative Learning-rate Decay, we use separate optimizers and schedulers with different learning rates for the neural network weights before and after volume density calculation.

**Additional Details** For our training, we generated images by uniformly sampling viewpoints from an upper hemisphere around the object. The training process, conducted using an RTX 6000 Ada, took approximately 4.5 hours for each scene. It consisted of 11 epochs, with each epoch having 1,884 training steps. The images created during training were 160 by 160 pixels in size and were later resized to 224 by 224 pixels through bicubic interpolation to comply with the requirements of the CLIP encoder. Our model follows the parameterization of MVSNerF. We used a two-stage stepwise sampling method for more refined sampling along rays, leading to better scene reconstruction. Additionally, we utilized Adam [15] with hyperparameters  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$ ,  $\epsilon = 10^{-7}$ , and employed the CosineAnnealingLR [23] as our learning rate scheduler. For the MLP that outputs volume density, we started with an initial learning rate of  $5.0 \times 10^{-4}$ . The learning rate for the MLP responsible for the color output was set to 0.1 of the volume density. We set the hyperparameters to  $\lambda_1 = 0.1$  and  $\lambda_2 = 0.001$ , following the approaches of DietNeRF [3] and InfoNeRF [31]. This setting ensures that the scale between each model’s loss and the RGB loss matches that of their respective prior art.

## 4.2 Comparison with Existing Methods

Table 1 and Fig. 5 demonstrate that our proposed method, possessing both geometric and semantic consistency, clearly outperforms methods with only one type of consistency, both quantitatively and qualitatively. We prefer readers to visit the Supplementary material for detailed qualitative evaluations and object-level scores for other objects.

Note that, MVSNerF is not fine-tuned but compared under zero-shot conditions. The reason is that fine-tuning with three nearby input images diminishes its generalization ability, making it less effective even near the input images, as carefully validated in the Supplementary material. Nevertheless, its zero-shot reconstruction ability with three adjacent inputs stabilizes our learning, and the

reconstruction accuracy in areas close to input images is obviously high as in Fig. 4. Therefore, we decided to use MVSNerF as our backbone.

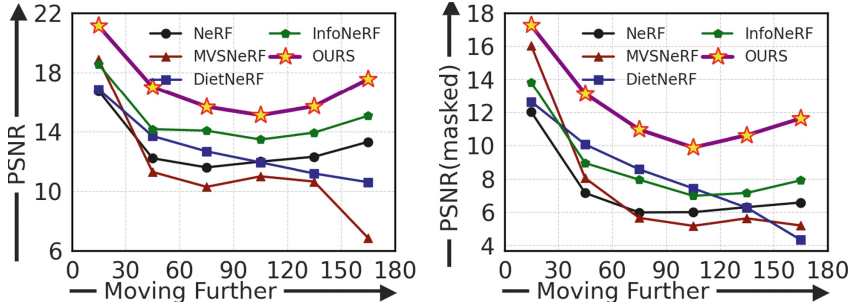
Fig. 4 demonstrates how evaluation scores change as the novel viewpoint moves away from the input image locations. Near these input locations, combining semantic with geometric consistency enables more detailed reconstruction than using geometric consistency alone. Even when moving further from input image locations, the fusion of semantic and geometric consistency allows for high-precision reconstruction. This is because the two types of consistency work together to compensate for each other’s weaknesses, providing a more comprehensive understanding of the scene. The geometric consistency ensures that the reconstruction adheres to the spatial constraints imposed by the input images, while the semantic consistency fills in the missing details and ensures that the reconstruction is coherent and meaningful. The two consistencies are also supported by the entropy regularization of the volume density.

The reason NeRF, neither using geometric nor semantic consistency, outperforms metrics like SSIM and LPIPS in Table 1 is considered to be because it outputs white, the background color in the NeRF Dataset, in areas completely away from the input image positions. RegNeRF [34] also mentions this issue as evaluation bias, discussing the relationship between evaluation scores and background color estimation. Following RegNeRF, by evaluating only the non-background elements, we removed the background-induced evaluation bias and conducted further analysis in Table 1. PSNR (masked) clearly shows NeRF’s significant susceptibility to this bias, and our proposed method outperforms prior arts even after eliminating this bias.

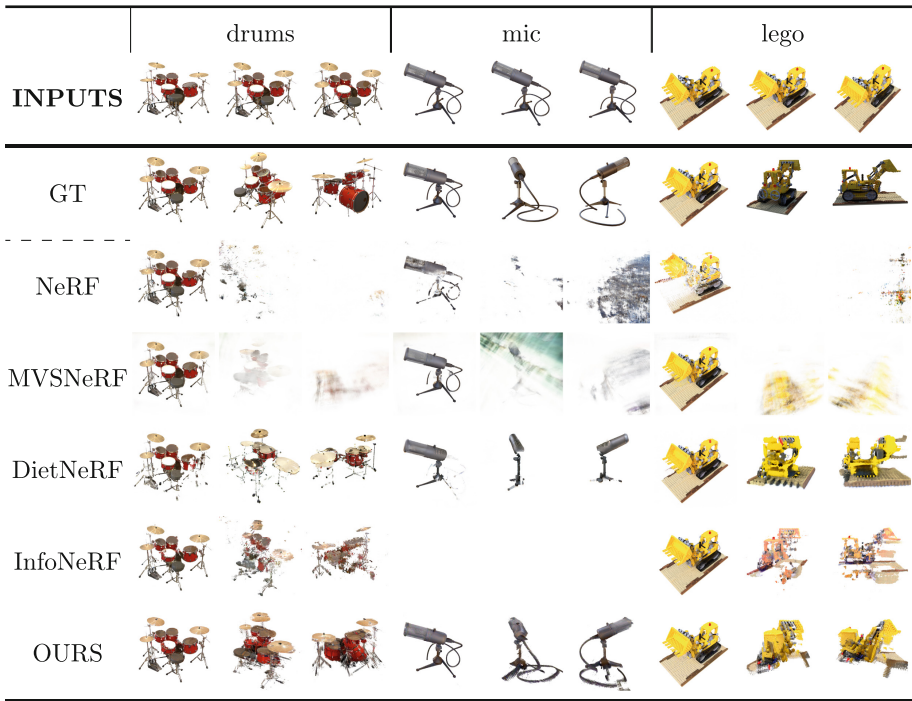
As shown in Fig. 5, the method using only geometric consistency can accurately reconstruct areas captured in the input images but fails to reconstruct unknown areas on the sides and back. The method employing only semantic consistency can utilize semantic coherence to reconstruct areas close to the input images, but it becomes clear that semantic consistency alone cannot reconstruct areas not reflected in the input images. However, our method enables complete 3D reconstruction and arbitrary viewpoint image generation through the complementary collaboration of semantic and geometric consistencies. In areas captured by the input images, it achieves precise reconstruction based on geometric consistency, while semantic consistency allows for the reconstruction of unknown areas by leveraging the information from geometric consistency.

### 4.3 Ablation Study

**Impact of Each Loss Term** Table 2 reveals the impact of each loss term individually. The combination of these losses significantly boosts all metrics. Entropy minimization loss greatly improves all metrics, while semantic loss slightly deteriorates SSIM but significantly improves PSNR and LPIPS. This improvement is thought to occur because entropy minimization loss encourages accurate geometry estimation, and semantic loss contributes to making rendered images more high-quality. DietNeRF [3] mentions that SSIM disagrees with human judgments



**Fig. 4. PSNR of novel views versus the angle of deviation from input images:** The figure demonstrates how utilizing geometric consistency near the input images enables detailed reconstruction, while in areas farther from the input images, semantic and geometric consistency work together. The evaluation divides the average angles formed by the input images and the evaluation image, based on the origin, into six intervals, calculating the mean within each.



**Fig. 5. Quantitative Evaluation:** Images were rendered with *roughly same, orthogonal, and inverse view directions* relative to the input views. The results highlight that our approach significantly outperforms by combining both geometric and semantic consistencies, compared to methods focusing on either alone.



**Table 1. Quantitative Evaluation:** We conducted a comprehensive assessment comparing the generated images to ground-truth images across comprehensive rendering orientations. Our proposed method consistently outperforms the other two methods which have either consistency in isolation across all evaluation metrics. PSNR (masked) calculates scores for non-background elements, specifically to exclude the evaluation bias RegNeRF [34] discusses.

Method	PSNR $\uparrow$	PSNR (masked) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
NeRF [9]	12.48	7.01	0.731	0.337
MVSNeRF [1]	11.63	7.35	0.731	0.343
DietNeRF [3]	13.11	8.96	0.699	0.344
InfoNeRF [31]	14.45	8.61	0.763	0.254
OURS	<b>16.55</b>	<b>12.01</b>	<b>0.779</b>	<b>0.219</b>

of similarity, indicating that incorporating semantic loss may lower SSIM outside of human perception.

**Table 2. Impact of Each Loss Term:** The contributions of semantic loss and entropy minimization loss are clearly identified. Each loss contributes to improvements, and their combination notably enhances accuracy in all metrics, especially in PSNR and LPIPS.

$\mathcal{L}_{\text{RGB}}$	$\mathcal{L}_{\text{SC}}$	$\mathcal{L}_{\text{entropy}}$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
✓			11.63	0.731	0.343
✓	✓		14.59	0.720	0.319
✓		✓	14.81	<b>0.770</b>	0.250
✓	✓	✓	<b>16.26</b>	0.762	<b>0.238</b>

**Impact of Black Blending** Table 3 and Fig. 6 reveal that the black blending outperforms when we clarify the effects of employing white, random (from DreamField) background in the rendered images during training. Its perceived advantage over the white background is thought to stem from its ability to discriminate white fog-like artifacts and backgrounds. Additionally, it seems to offer superiority over random backgrounds by providing accurate restoration signals without introducing noise to the CLIP semantic vectors. This approach also suggests that using a white background for rendered images during training, a common practice, negatively impacts learning.

**Impact of Relative Learning-rate Decay** Table 4 shows the changes in evaluation metrics when we adjust the relative learning speeds in our experiments to investigate how slowing down the learning speed of an MLP predicting color affects its effectiveness compared to an MLP estimating volume density. These results demonstrate that reducing the color MLP’s learning speed to 0.1 times

**Table 3. Impact of Black Blending:** Black blending outperforms other methods. A white background fails to differentiate between cloud-like floaters and the background, while a random background introduces noise to the semantic vectors from CLIP. Our method avoids these issues and suggests that using a white background for training, a common practice, negatively impacts learning.

Background	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
White	15.20	0.747	0.246
Random [2]	15.70	0.747	0.251
Black (OURS)	<b>16.26</b>	<b>0.762</b>	<b>0.238</b>



**Fig. 6. Effect of Black Blending:** Black blending clearly outperforms other methods, resulting in cleaner outcomes. It notably sharpens object boundaries and allows for a more accurate estimation of object shapes.

is most effective. The reason this method is considered effective is because prioritizing the learning speed of the MLP for estimating volume density over the MLP for predicting color promotes accurate geometry estimation.

## 5 Limitation

While our approach enables the reconstruction of an entire scene from three closely-spaced images, the final results still have room for improvement. Use of depth priors, either with patch-based [34] or image-based [18], may improve the reconstruction quality. Another limitation is that we have only evaluated

**Table 4. Impact of Relative Learning Rate Decay:** Lowering the MLP’s color learning rate compared to volume density boosts accuracy by focusing on volume density.

Relative Learning-rate	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
1.0	16.26	0.762	0.238
0.1	<b>16.55</b>	<b>0.779</b>	<b>0.219</b>
0.01	16.50	<b>0.779</b>	<b>0.219</b>
0.001	16.37	0.764	0.238

our method on a set of a single object. Further testing on multiple objects and real-world datasets is needed to fully assess its potential and generalizability. The rendering of CLIP is time-consuming and accelerating the computation is needed in practice. The difficulty of evaluating the generated image quality is an issue for the entire research field.

## 6 Conclusion

We have presented a method that, by complementarily coordinating semantic and geometric consistencies, can estimate a complete radiance field and generate images from arbitrary viewpoints, even in settings challenging for either consistency alone. We propose various techniques for improved performance and clarified their effectiveness one by one. The condition of using three close-shot images, similar to those used in iPhones, also demonstrates high versatility.

## References

1. A.Chen, Z.Xu, F.Zhao, X.Zhang, F.Xiang, J.Yu, H.Su: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: CVPR (2021)
2. A.Jain, B.Mildenhall, J.Barron, P.Abbeel, B.Poole: Zero-shot text-guided object generation with dream fields. In: CVPR (2022)
3. A.Jain, M.Tancik, P.Abbeel: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: CVPR (2021)
4. A.Radford, J.Kim, C.Hallacy, A.Ramesh, G.Goh, S.Agarwal, G.Sastry, A.Askell, P.Mishkin, J.Clark, G.Krueger, I.Sutskever: Learning transferable visual models from natural language supervision. In: ICML (2021)
5. A.Trevithick, B.Yang: Grf: Learning a general radiance field for 3d scene representation and rendering. In: ICCV (2021)
6. A.Yu, V.Ye, M.Tancik, A.Kanazawa: pixelnerf: Neural radiance fields from one or few images. In: CVPR (2021)
7. B.Kerbl, G.Kopanas, T.Leimkühler, G.Drettakis: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (2023)
8. B.Mildenhall, P.Hedman, R.Martin-Brualla, P.Srinivasan, J.Barron: NeRF in the dark: High dynamic range view synthesis from noisy raw images. In: CVPR (2022)
9. B.Mildenhall, P.Srinivasan, M.Tancik, J.Barron, R.Ramamoorthi, R.Ng: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
10. B.Poole, A.Jain, J.Barron, B.Mildenhall: Dreamfusion: Text-to-3d using 2d diffusion. In: ICLR (2023)
11. B.Roessle, N.Müller, L.Porzi, S.Bulò, P.Kontschieder, M.Nießner: Ganerf: Leveraging discriminators to optimize neural radiance fields. ACM Transactions on Graphics (2023)
12. C.Hsu, C.Chiu, C.Kuan: Fast single-view 3d object reconstruction with fine details through dilated downsample and multi-path upsample deep neural network. In: ICASSP (2020)
13. C.Jiang, A.Sud, A.Makadia, J.Huang, M.Nießner, T.Funkhouser: Local implicit grid representations for 3d scenes. In: CVPR (2020)
14. C.Wang, M.Chai, M.He, D.Chen, J.Liao: Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In: CVPR (2022)

15. D.Kingma, J.Ba: Adam: A method for stochastic optimization. In: ICLR (2014)
16. D.Xu, Y.Jiang, P.Wang, Z.Fan, H.Shi, Z.Wang: Sinnerf: Training neural radiance fields on complex scenes from a single image. In: ECCV (2022)
17. E.Chan, K.Nagano, M.Chan, A.Bergman, J.Park, A.Levy, M.Aittala, S.Mello, T.Karras, G.Wetzstein: GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In: ICCV (2023)
18. G.Wang, Z.Chen, C.Loy, Z.Liu: Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In: ICCV (2023)
19. H.Kato, Y.Ushiku, T.Harada: Neural 3d mesh renderer. In: CVPR (2018)
20. H.Song, S.Choi, H.Do, C.Lee, T.Kim: Blending-nerf: Text-driven localized editing in neural radiance fields. In: ICCV (2023)
21. H.Wang, X.Du, J.Li, R.Yeh, G.Shakhnarovich: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In: CVPR (2023)
22. I.Goodfellow, J.Pouget-Abadie, M.Mirza, B.Xu, D.Warde-Farley, S.Ozair, A.Courville, Y.Bengio: Generative adversarial nets. In: NeurIPS (2014)
23. I.Loshchilov, F.Hutter: Sgdr: Stochastic gradient descent with warm restarts. In: ICLR (2016)
24. J.Ho, A.Jain, P.Abbeel: Denoising diffusion probabilistic models. In: NeurIPS (2020)
25. J.Kwak, Y.Li, D.Yoon, D.Kim, D.Han, H.Ko: Injecting 3d perception of controllable nerf-gan into stylegan for editable portrait image synthesis. In: ECCV (2022)
26. J.Park, P.Florence, J.Straub, R.Newcombe, S.Lovegrove: Deepsdf: Learning continuous signed distance functions for shape representation. In: CVPR (2019)
27. J.Wynn, D.Turmukhambetov: DiffusioNeRF: Regularizing Neural Radiance Fields with Denoising Diffusion Models. In: CVPR (2023)
28. J.Yang, M.Pavone, Y.Wang: Freenerf: Improving few-shot neural rendering with free frequency regularization. In: CVPR (2023)
29. K.Deng, A.Liu, J.Zhu, D.Ramanan: Depth-supervised NeRF: Fewer views and faster training for free. In: CVPR (2022)
30. L.Melas-Kyriazi, C.Rupprecht, I.Laina, A.Vedaldi: Realfusion: 360° reconstruction of any object from a single image. In: CVPR (2023)
31. M.Kim, S.Seo, B.Han: Infonerf: Ray entropy minimization for few-shot neural volume rendering. In: CVPR (2022)
32. M.Kwak, J.Song, S.Kim: Geconerf: Few-shot neural radiance fields via geometric consistency. In: ICML (2023)
33. M.Niemeyer, A.Geiger: Giraffe: Representing scenes as compositional generative neural feature fields. In: CVPR (2021)
34. M.Niemeyer, J.Barron, B.Mildenhall, M.Sajjadi, A.Geiger, N.Radwan: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: CVPR (2022)
35. M.Son, J.Park, L.Guibas, G.Wetzstein: Singraf: Learning a 3d generative radiance field for a single scene. In: CVPR (2023)
36. M.Uy, R.Martin-Brualla, L.Guibas, K.Li: Scade: Nerfs from space carving with ambiguity-aware depth estimates. In: CVPR (2023)
37. N.Somraj, A.Karanayil, R.Soundararajan: SimpleNeRF: Regularizing sparse input neural radiance fields with simpler solutions. SIGGRAPH Asia (2023)
38. P.Dhariwal, A.Nichol: Diffusion models beat gans on image synthesis. In: NeurIPS (2021)
39. Q.Li, F.Multon, A.Boukhayma: Learning generalizable light field networks from few images. In: ICASSP (2023)

40. Q.Meng, A.Chen, H.Luo, M.Wu, H.Su, L.Xu, X.He, J.Yu: GNeRF: GAN-based Neural Radiance Field without Posed Camera. In: ICCV (2021)
41. Q.Wang, et al.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021)
42. R.Jensen, A.Dahl, G.Vogiatzis, E.Tola, H.Aanæs: Large scale multi-view stereopsis evaluation. In: CVPR (2014)
43. R.Liu, R.Wu, B.Hoorick, P.Tokmakov, S.Zakharov, C.Vondrick: Zero-1-to-3: Zero-shot one image to 3d object. In: ICCV (2023)
44. R.Wu, B.Mildenhall, P.Henzler, K.Park, R.Gao, D.Watson, P.Srinivasan, D.Verbin, J.Barron, B.Poole, A.Holynski: Reconfusion: 3d reconstruction with diffusion priors. In: CVPR (2024)
45. R.Zhang, P.Isola, A.Efros, E.Shechtman, O.Wang: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
46. T.Fujihashi, T.Koike-Akino, T.Watanabe: Soft 2d-to-3d delivery using deep graph neural networks for holographic-type communication. In: ICASSP (2023)
47. V.Sitzmann, J.Thies, F.Heide, M.Nießner, G.Wetzstein, M.zollhöfer: Deepvoxels: Learning persistent 3d feature embeddings. In: CVPR (2019)
48. V.Sitzmann, M.Zollhöfer, G.Wetzstein: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: NeurIPS (2019)
49. Y.Song, S.Ermon: Generative modeling by estimating gradients of the data distribution. In: NeurIPS (2019)
50. Z.Zhou, S.Tulsiani: Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In: CVPR (2023)



# Fast and Consistently Accurate Perspective-n-Line Pose Estimation

George Terzakis<sup>1</sup>  and Manolis Lourakis<sup>2</sup> 

<sup>1</sup> Institute of Communication and Computer Systems, 9 Ir. Polytechniou, Zografou, 157 73 Athens, Greece

`george.terzakis@iccs.gr`

<sup>2</sup> Institute of Computer Science Foundation for Research and Technology – Hellas, P.O. Box 1385, 711 10 Heraklion, Crete, Greece

`lourakis@ics.forth.gr`

**Abstract.** We present an approach for estimating the pose of a pinhole camera from a set of 3D lines and their corresponding line segment projections on a single image. The problem is formulated as a non-linear quadratic program on the elements of the rotation matrix in a manner that establishes direct correspondence to the Perspective-n-Point (PnP) problem. By leveraging this connection to the PnP, existing methodologies are reapplied to recover the camera rotation from a constrained quadratic program. Furthermore, a novel least squares formulation is proposed to estimate the translation of the camera. Detailed comparative experiments demonstrate that the proposed approach is robust to noise and outperforms established techniques in terms of accuracy.

**Keywords:** Perspective-n-Line · PnL · PnP · Pose Estimation · Quadratically Constrained Quadratic Program · SQPnL · Line Segment · Structure from Motion · SfM

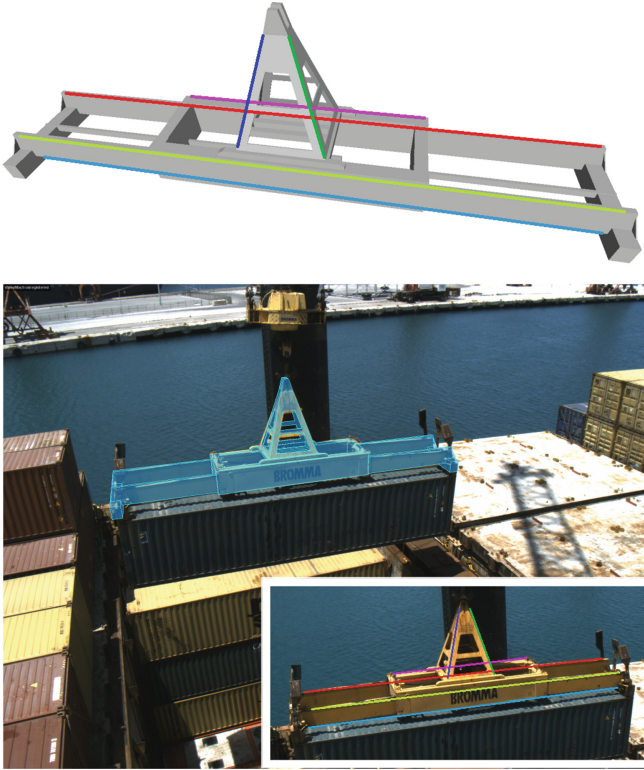
## 1 Introduction

Human-made environments typically abound with linear edges which give rise to straight line segments when imaged in 2D. Line segments can be accurately localized in images at a reasonable computational cost [9], offering robust structural cues by delineating a scene’s main elements. They are also moderately robust to noise, occlusions and illumination or viewpoint changes. Furthermore, they are often present on even poorly textured objects, for which techniques based on local patch detectors and descriptors such as [21], are not applicable. Hence, in certain structured environments with weak texture, straight line segments are the preferred type of visual features for tasks such as motion analysis [19], scene reconstruction [12, 17], visual SLAM [31] and object recognition [6].

This paper concerns the Perspective-n-Line (PnL) problem which given a set of 3D lines, aims to determine a calibrated camera’s pose (i.e., position and

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78456-9\\_7](https://doi.org/10.1007/978-3-031-78456-9_7).



**Fig. 1.** Top: 3D mesh model for a container crane spreader and six color-coded lines. Bottom: The spreader while lifting a container. Using manually specified model-to-image line correspondences delineated with identical colors in the bottom right inset image, the spreader’s pose was estimated with the proposed method; the mesh model was then rendered in cyan with the estimated pose and overlaid semi-transparently on the image with its edges highlighted; see also [19].

orientation) from the line projections in an image taken with that camera (cf. Fig. 1). A problem similar to PnL that has received considerably more attention is the Perspective-n-Point (PnP) problem [23,26]. PnP seeks to determine the camera pose that relates a set of 3D world points to their corresponding 2D image counterparts. Our work concerns a formulation of PnL as a non-linear quadratic program (NLQP) which we call SQPnL. It develops a cost that encodes information regarding the geometry of at least three 3D lines and their matching 2D image projections. Similarly to our SQPnP solver for PnP [26], this cost is optimized for the rotation by finding special feasible points and then conducting low-iteration local searches in their vicinity.

The rest of the paper is organized as follows. Section 2 presents an overview of relevant works from the literature. The novel aspects of the proposed method are outlined in Sect. 3 whereas its details are presented in Sect. 4. Experimental evaluation results are reported in Sect. 5 and a conclusion is in Sect. 6.

## 2 Related Work

It is known that when three line correspondences are available, the solution to the Perspective-3-Line (P3L) problem is not uniquely determined. This is due to the fact that the order of the general polynomial arising from P3L is 8 [5, 32], i.e. considerably higher compared to 4 which is the order pertaining to the analogous polynomial of the Perspective-3-Point (P3P) problem defined on points. As a result, the P3L incurs increased computational cost, is more sensitive to noise and admits more potential solutions. Dhome et al. [8] presented one of the earliest closed-form solutions for the P3L problem adopting a polynomial approach. Xu et al. [32] provided an analysis of special line configurations and concluded that the order of the P3L polynomial can be reduced depending on the 3D lines arrangement. An example of the latter is the case of three lines lying in a common plane that is addressed in [3]. A recent algebraic method to solve P3L is provided by Wang et al. in [29].

While geometric ambiguities can cause P3L to have up to 8 possible solutions, the PnL problem with four or more lines typically has a unique solution. Owing to this observation and the potential of data redundancy to increase accuracy and robustness, algorithms for the over-constrained case have attracted considerable interest. For example, Ansar and Daniilidis [2] employed a general procedure for linearizing quadratic systems to convert the polynomial system to a linear one in the elements of the rotation matrix. The method guarantees a solution for non-critical configurations of  $n \geq 4$  lines, yet it is very sensitive to image noise and does not scale well with the number of lines. The first globally optimal and non-iterative method for PnL was AlgLS, proposed by Mirzaei and Roumeliotis [22]. They decoupled rotation from translation by deriving a multivariate polynomial system for the rotation parameters that was solved with resultants. Due to the latter, the method is computationally expensive and returns a large number of candidate solutions. Furthermore, it suffers from singularities at orientations  $\pm\pi$  that stem from the Cayley rotation representation [27] employed.

Zhang et al. [35] developed RPnL, a solver that performs well on small-sized sets of lines but becomes less accurate and slow on larger ones. RPnL forms a suboptimal problem and operates by selecting a rotation axis to separate lines into triplets, then building a sixteenth order univariate cost function from P3L polynomials and finally retrieving the optimum among the local minima. Extending [35], Xu et al. [32] proposed ASPnL which is generally accurate, yet is computationally expensive for larger numbers of lines. They also explored the similarity between PnL and PnP and developed a series of linear PnL formulations of which LPnL-Bar\_LS that uses barycentric coordinates performs best.

Přibyl et al. [24] used the Plücker coordinates to represent 3D lines and derived a formulation of PnL as a homogeneous linear system that is solved with the DLT algorithm. Wang et al. [30] developed SRPnL by deriving a closed-form technique that involves solving a fifteenth order polynomial followed by root polishing with a single Gauss-Newton step. Yu et al. [34] proposed OPnL which employs the Cayley rotation representation and derives a cubic system



that is solved with Gröbner basis techniques. While the latter provide a powerful framework for solving polynomial systems, they typically construct linear solvers involving fairly large coefficient and action matrices that need considerable time for their evaluation and decomposition. This limitation can be a hindrance to applications in demand of real-time, timely results. More recently, Wang et al. [28] proposed yet another formulation which also employs the Cayley parameterization for rotation to derive a system of polynomial equations solved with the hidden variable method. For the sake of completeness, it is noted that iterative methods for PnL such as [14, 18] have also been proposed. However, their need for initialization combined with the possibility of converging to a local minimum, makes them less attractive in practice.

### 3 Contributions

This work introduces a novel methodology that employs, for the first time in PnL, existing machinery developed in [26] for the PnP problem. Similarly to its point-based analog, the proposed solver is non-minimal and has linear time complexity. Coupled with its consistent accuracy, these features lend it a great deal of practical utility, particularly when used in the context of locally optimized (LO) RANSAC [15] or combined with modern approaches to robust estimation such as graduated non-convexity [33] or adaptive kernels [4]. Our contributions are summarized in the following:

1. We put forward a novel method<sup>1</sup> for constructing a quadratic cost function in the elements of the rotation matrix. By doing so, the PnL problem is cast as a quadratically constrained quadratic program (QCQP), which can be solved by leveraging existing PnP algorithms. Furthermore, we present a linear method to recover the camera position from the estimated rotation.
2. Building upon the aforementioned methodology, we adapt the approach of SQPnP [26] to derive a novel PnL solver. We demonstrate experimentally that this solver is resilient to noise and exhibits consistent performance, estimating pose with higher accuracy compared to state-of-the-art PnL solvers.

### 4 Method

A 3D line can be represented by a parametrized collection of points,

$$\mathbf{L} := \{ \mathbf{P} + \lambda \mathbf{u} \in \mathbb{R}^3 : (\mathbf{P}, \mathbf{u}) \in \mathbb{R}^3 \times \mathcal{S}^2, \lambda \in \mathbb{R} \}, \quad (1)$$

where  $\mathcal{S}^2$  is the unit sphere in the 3D space  $\mathbb{R}^3$ ,  $\mathbf{P}$  is an arbitrary point on the 3D line and  $\mathbf{u}$  is its direction vector. With the representation of Eq. (1) in place,

---

<sup>1</sup> Some of the constituents have been previously employed, particularly the quadratic cost in [32].

we will henceforth refer to 3D lines either by name, or by means of the tuple  $(\mathbf{P}, \mathbf{u})$ . A line  $\mathbf{l}$  in a 2D plane is represented with its Hesse normal form, i.e.

$$\mathbf{l} := \{\mathbf{x} \in \mathbb{R}^2 : \mathbf{n}^T \mathbf{x} + c = 0, \mathbf{n} \in \mathcal{S}^1, c \in \mathbb{R}\}, \quad (2)$$

where  $\mathcal{S}^1$  is the unit circle,  $\mathbf{n}$  is the unit-norm normal of the line and  $c$  a real constant. Clearly, a 2D line in a plane embedded in 3D space is also a 3D line and can therefore have the following tuple representation,

$$\mathbf{l} = \left( \begin{bmatrix} -c\mathbf{n} \\ 1 \end{bmatrix}, M\mathbf{n} \right), \quad M = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad (3)$$

where  $\mathbf{n}^T \mathbf{x} + c = 0$  is the Hesse line equation defined in (2) above.

Conversely, for a 3D line  $\mathbf{L}$  that does not pass through the origin<sup>2</sup>, we may obtain its projection,  $\text{proj}(\mathbf{L})$ , on the camera plane at  $Z = 1$  as a 3D line tuple (subsequently converted into a 2D line tuple),

$$\text{proj}(\mathbf{L}) = \begin{cases} \left( \begin{array}{l} \left( \frac{(\mathbf{1}_z^T \hat{\mathbf{P}})\hat{\mathbf{P}} + (\mathbf{1}_z^T \hat{\mathbf{u}})\mathbf{u}}{(\mathbf{1}_z^T \hat{\mathbf{P}})^2 + (\mathbf{1}_z^T \hat{\mathbf{u}})^2}, \frac{-(\mathbf{1}_z^T \mathbf{u})\hat{\mathbf{P}} + (\mathbf{1}_z^T \hat{\mathbf{P}})\mathbf{u}}{\sqrt{(\mathbf{1}_z^T \hat{\mathbf{P}})^2 + (\mathbf{1}_z^T \hat{\mathbf{u}})^2}} \right) \\ (\mathbf{P} + \mathbf{1}_z, \mathbf{u}) \end{array} \right) & (\mathbf{1}_z^T \hat{\mathbf{P}})^2 + (\mathbf{1}_z^T \hat{\mathbf{u}})^2 \neq 0, \\ & \text{otherwise} \end{cases}, \quad (4)$$

where  $\mathbf{1}_z = [0 \ 0 \ 1]^T$  and  $\hat{\mathbf{P}}$  is the (unit) bearing vector of the nearest point on  $\mathbf{L}$  to the origin,

$$\hat{\mathbf{P}} = \frac{(\mathbf{I}_3 - \mathbf{u}\mathbf{u}^T)\mathbf{P}}{\sqrt{\mathbf{P}^T(\mathbf{I}_3 - \mathbf{u}\mathbf{u}^T)\mathbf{P}}}. \quad (5)$$

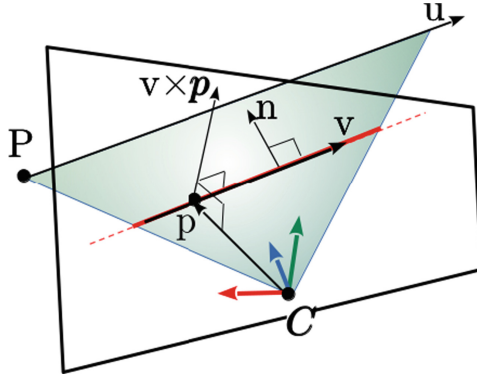
Derivations of Eqs. (3), (4), and (5) are provided in the supplementary material.

#### 4.1 A Quadratic Program in the Elements of the Rotation Matrix

The PnL is intrinsically related to its point counterpart, the PnP. It has been shown that it yields similar problem formulations to those obtained from the PnP, e.g., [32]. We describe next such an approach which decouples the unknown orientation from the translation and yields the well-known in the PnP literature quadratically constrained quadratic program (QCQP) on the rotation matrix.

Consider a camera orientation matrix  $\mathbf{R} \in \mathcal{SO}(3)$  and position  $\mathbf{C}$  with respect to a world frame. To clarify conventions, a world point  $\mathbf{P}_w$  is assumed to transform to the local camera frame as  $\mathbf{P}_c = \mathbf{R}^T(\mathbf{P}_w - \mathbf{C})$ . Let  $\mathbf{L} = (\mathbf{P}, \mathbf{u})$  be a 3D line in the world frame and  $\mathbf{l} = (\mathbf{p}, \mathbf{v})$  be the corresponding projected line on the  $Z = 1$  plane in the local camera frame where  $\mathbf{p}$  and  $\mathbf{v}$  are given as per Eq. (2). Figure 2 illustrates the geometry of the projection,  $\mathbf{l}$ , of  $\mathbf{L}$  onto the  $Z = 1$  plane of the local camera frame, via the plane induced by  $\mathbf{L}$  and  $\mathbf{C}$ .

<sup>2</sup> The projections of 3D lines that pass through the origin are points and therefore not applicable to the PnL setup.



**Fig. 2.** The projection of a 3D line,  $L = (P, u)$ , as a line  $l = (p, v)$  in the camera with local image plane  $Z = 1$  positioned at  $C$  in the world; the nearest point on the line to  $C$  is  $p$  and the normal of the plane is given by the cross product of  $p$  with the normal of the line on the plane,  $n$ .

In an ideal, noise-free setup, the 3D line  $L$  will lie in the plane defined by the camera center  $C$  and the line  $l$ . Vectors  $p$  and  $v$  of Eq. (3), i.e.,

$$p = \begin{bmatrix} -cn \\ 1 \end{bmatrix}, \quad v = Mn, \quad (6)$$

comprise an orthogonal basis for this plane in the camera frame. Thus, the normal of the plane defined by  $L$  and  $C$  should be the cross-product of  $v$  and  $p$ . This cross product, transformed by  $R$  to the world frame, should in turn be orthogonal to the direction,  $u$ , of  $L$ . Thus, the following should hold

$$u^T R(v \times p) = 0. \quad (7)$$

The orthogonality constraint of Eq. (7) can be used to devise a quadratic cost function in the elements of  $R$ . Consider  $n$  3D lines  $L_1, \dots, L_n$  and their corresponding projections,  $l_1, \dots, l_n$  on the  $Z = 1$  plane in the local camera frame. We may construct a cost function  $\mathcal{C}$ , that penalizes the average deviation from orthogonality as

$$\mathcal{C} = \sum_{i=1}^n \left( u_i^T R \left( \frac{v_i \times p_i}{\|v_i \times p_i\|} \right) \right)^2, \quad (8)$$

where  $p_i$  is the nearest point from  $l_i$  to  $C$ ,  $v_i$  the direction of  $l_i$  and  $\|\cdot\|$  is the Euclidean norm. Note that the normalization of the cross-product  $v_i \times p_i$  is necessary to eliminate arbitrary scaling in the  $i$ -th term. We next introduce operators  $\text{vec}(\cdot)$  and  $\text{mat}(\cdot)$  in order to map  $R$  to a vector  $r \in \mathbb{R}^9$  by stacking its elements row-wise and vice versa, as follows:

$$r = \text{vec}(R) = [r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}]^T \iff \text{mat}(r) = R. \quad (9)$$

Then,  $\mathcal{C}$  can be rewritten as a quadratic expression of  $\mathbf{r}$ ,

$$\mathcal{C} = \mathbf{r}^T \boldsymbol{\Omega} \mathbf{r}, \quad \boldsymbol{\Omega} = \sum_{i=1}^n \left( \mathbf{I}_3 \otimes \frac{\mathbf{v}_i \times \mathbf{p}_i}{\|\mathbf{v}_i \times \mathbf{p}_i\|} \right)^T \mathbf{u}_i \mathbf{u}_i^T \left( \mathbf{I}_3 \otimes \frac{\mathbf{v}_i \times \mathbf{p}_i}{\|\mathbf{v}_i \times \mathbf{p}_i\|} \right), \quad (10)$$

where  $\otimes$  denotes the Kronecker (matrix direct) product. We observe that  $\mathcal{C}$  is a symmetric function and therefore, for every minimizer with a negative determinant in the orthogonal group  $\mathcal{O}(3)$  (a reflection), there exists a minimizer in  $\mathcal{SO}(3)$  by means of negation. Thus, for convenience, we may cast our QCQP in  $\mathcal{O}(3)$  as follows,

$$\underset{\mathbf{r} \in \mathbb{R}^9}{\text{minimize}} \quad \mathbf{r}^T \boldsymbol{\Omega} \mathbf{r} \quad \text{s.t.} \quad \mathbf{g}(\mathbf{r}) = \mathbf{0}_6, \quad (11)$$

where  $\mathbf{g}(\mathbf{r}) \in \mathbb{R}^6$  is a vector-valued function that imposes the orthonormality constraints on  $\mathbf{r}$ :

$$\mathbf{g}(\mathbf{r}) = [\|\mathbf{r}_{1:3}\|^2 - 1, \|\mathbf{r}_{4:6}\|^2 - 1, \|\mathbf{r}_{7:9}\|^2 - 1, \mathbf{r}_{1:3}^T \mathbf{r}_{4:6}, \mathbf{r}_{1:3}^T \mathbf{r}_{7:9}, \mathbf{r}_{4:6}^T \mathbf{r}_{7:9}]^T. \quad (12)$$

We have thus arrived at a problem formulation that is commonly met in PnP methods in the literature. Such methods can be loosely categorized according to their use of a) unconstrained least squares (LS) [16,32], b) polynomial solvers [20,34,36], and c) quadratic programming techniques, including semidefinite programming (SDP) [1,26]. Several methods in the aforementioned first two categories have been adapted for the PnP problem, e.g. [22,32,34]. Here we derived the QCQP of Eq. (11), consequently we can apply the approach of SQPnP directly and solve it by employing Algorithms 1 and 2 described in Sec. 3 of [26]. Our SQPnP solver attempts local searches from special initial estimates of the rotation matrix and thereafter determines the solution of the QCQP by choosing the best amongst the recovered minimizers (cf. Sec. 2 in [26]). This approach has proven very effective for the PnP in practice. As will be confirmed by the experiments in Sect. 5, this also holds true for the quadratic program (11) formulated for PnP in this paper.

## 4.2 Estimating Camera Position from Known Orientation

Suppose  $\mathbf{r} \in \mathbb{R}^9$  is a minimizer of the QCQP in Eq. (11) and  $\mathbf{R} = \text{mat}(\mathbf{r}) \in \mathcal{O}(3)$  the corresponding orthonormal matrix. To recover the position  $\mathbf{C}$  of the camera in the world, we make use of the coplanarity constraint between the camera center and the plane defined by the 3D line and its projection, shown in Fig. 2.

Consider the 3D line representation of Eq. (1) as a parametric expression of a point,

$$\mathbf{L}(\lambda) = \mathbf{P} + \lambda \mathbf{u}, \quad \lambda \in \mathbb{R}, \quad (13)$$

uniquely identified by the parameter  $\lambda$ . We may thus resort to point transformations to obtain the corresponding 3D line  $\mathbf{L}_c(\lambda)$  in the local camera frame,

$$\mathbf{L}_c(\lambda) = \mathbf{R}^T (\mathbf{L}(\lambda) - \mathbf{C}) = \mathbf{R}^T (\mathbf{P} - \mathbf{C}) + \lambda \mathbf{R}^T \mathbf{u}. \quad (14)$$

In an ideal, noise-free setup,  $\mathbf{R}^T(\mathbf{P} - \mathbf{C})$  lies in the plane defined by  $\mathbf{C}$  and  $\mathbf{l}$ . Thus, it can be written as a linear combination of  $\mathbf{p}$  and  $\mathbf{v}$ ,

$$\mathbf{R}^T(\mathbf{P} - \mathbf{C}) = \mathbf{B}\mathbf{x}, \quad (15)$$

where  $\mathbf{B} = [\mathbf{p}/\|\mathbf{p}\| \ \mathbf{v}/\|\mathbf{v}\|] \in \mathbb{R}^{3 \times 2}$  is the basis of the local plane defined by  $\mathbf{l}$  and  $\mathbf{C}$ , and  $\mathbf{x} \in \mathbb{R}^2$ . We may therefore devise a linear least squares cost function on the position  $\mathbf{C}$  of the camera in the world,

$$\mathcal{J} = \sum_{i=1}^n \|\mathbf{R}^T(\mathbf{P}_i - \mathbf{C}) - \mathbf{B}_i \mathbf{x}_i\|^2, \quad (16)$$

where  $\mathbf{B}_i$  is the orthonormal basis of the plane defined by  $\mathbf{C}$  and the  $i$ -th line  $\mathbf{l}_i$  identified on the 2D plane  $Z = 1$  (see Fig. 2). Taking first order conditions of the cost function in Eq. (16) with respect to  $\mathbf{x}_i$  and  $\mathbf{C}$ , we obtain:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{x}_i} = \mathbf{0} \iff -\mathbf{B}_i^T \mathbf{R}^T(\mathbf{P}_i - \mathbf{C}) + \mathbf{x}_i = \mathbf{0}, \quad (17)$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{C}} = \mathbf{0} \iff \sum_{i=1}^n (\mathbf{P}_i - \mathbf{C}) - \mathbf{R} \sum_{i=1}^n \mathbf{B}_i \mathbf{x}_i = \mathbf{0}. \quad (18)$$

Substituting Eq. (17) into Eq. (18) yields the solution,

$$\mathbf{C} = \mathbf{R} \left( \sum_{i=1}^n (\mathbf{I}_3 - \mathbf{B}_i \mathbf{B}_i^T) \right)^{-1} \left( \sum_{i=1}^n (\mathbf{I}_3 - \mathbf{B}_i \mathbf{B}_i^T) \mathbf{R}^T \mathbf{P}_i \right). \quad (19)$$

## 5 Experiments

### 5.1 Synthetic Experiments

Using synthetic data, this section compares Matlab implementations of the proposed SQPnL solver and the following PnL methods: Ansar [2], AlgLS [22], DLT [25], LPnL-Bar\_LS [32], RPnL [35], ASPnL [32], SRPnL [30], OPnL [34] and HPnL (hidden variable PnL) [28]. The comparisons focus on both the reprojection error and the deviation from ground truth across multiple runs.

The following testing framework is adopted. Euclidean quantities are expressed in units of meters. World 3D lines are defined from pairs of points randomly sampled from an isotropic Gaussian distribution with standard deviation 3, i.e.  $\mathbf{M}_i \sim \mathcal{N}(\bar{\mathbf{M}}, 3^2 \mathbf{I}_3)$ , with  $\bar{\mathbf{M}} \equiv [1.5/4, 1.5/4, 4]^T$ . Similarly, camera poses comprising position  $\mathbf{C}$  and MRP [27] orientation parameters  $\boldsymbol{\psi}$  in the world frame are sampled from a zero-mean 6D Gaussian distribution

$$\begin{bmatrix} \mathbf{C} \\ \boldsymbol{\psi} \end{bmatrix} \sim \mathcal{N} \left( \mathbf{0}_6, \begin{bmatrix} \sigma_{\mathbf{C}}^2 \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_{\boldsymbol{\psi}}^2 \mathbf{I}_3 \end{bmatrix} \right), \quad (20)$$

where the standard deviations were chosen as  $\sigma_{\mathbf{C}} = 0.15$  and  $\sigma_{\boldsymbol{\psi}} = 0.001$ . The generated lines were filtered to ensure that they lie in front of the simulated

camera. The latter was assumed to have a focal length  $f = 1400$  pixels and image size  $1600 \times 1400$ . A collection of 200 random 3D lines was generated and then for every  $n \in \{4, \dots, 20\}$ , a population of 500 sets of  $n$  3D lines each was randomly sampled. For each such set, a camera pose was next generated using Eq. (20). Finally, the pose was used to project all sets on the image plane and the projection endpoints were perturbed with additive Gaussian noise  $\epsilon_i \sim \mathcal{N}(\mathbf{0}_2, \sigma_\epsilon^2 \mathbf{I}_2)$  with  $\sigma_\epsilon = 5$  pixels. For every set in a certain population, all PnL solvers under comparison were executed with their default parameters<sup>3</sup>.

Most solvers being compared return multiple solutions, some corresponding to 3D lines behind the camera. Since such solutions are not physically plausible, they are eliminated with cheirality checks [10], as follows. For each line segment, two 3D points were reconstructed from the intersection of the backprojecting rays defined by its endpoints and the known 3D line and the sign of their  $Z$  coordinate in the camera frame is examined. To account for these lines typically being skew, the reconstructed point was computed as the middle of their common perpendicular. This calculation is similar to that involved in the midpoint triangulation method [11]. Poses resulting in many reconstructed points being behind the camera (i.e.  $Z < 0$ ) were discarded. From the remaining poses, the one giving rise to the smallest reprojection error was retained and the average reprojection error for each solver was calculated across all its 500 executions.

The reprojection error quantifies the discrepancy between the actual image line segments and their corresponding projections predicted by the current pose estimate [13]. A limitation of line segment detection is that it inherently introduces uncertainty in the detected line segment endpoints [9]. To account for this, the reprojection error is computed by first projecting the 3D lines onto the image plane with the current pose estimate and then calculating the distances between the endpoints of the detected line segments and their closest points on the corresponding projected lines. For parity with approaches such as [28, 30, 32], SQPnL’s estimate is polished with one Gauss-Newton iteration on the reprojection error.

In addition to the reprojection errors for the PnL methods being compared, we also determined the reprojection error corresponding to the maximum likelihood pose estimate. This was obtained by iteratively minimizing the total reprojection error for each set’s noisy lines with the Levenberg-Marquardt (LM) non-linear least squares algorithm [7] initiated at the true pose. Plots of the average reprojection errors are in Fig. 3(a) which clearly demonstrates that our proposed SQPnL attains the smallest reprojection errors. To demonstrate the consistency of our solver in attaining errors that are similar to these of the maximum likelihood estimate, we also present in Fig. 3(b) results from exactly the same experiments for the maximum reprojection error. The latter is more informative regarding the consistency and the accuracy achieved by each PnL method, as its sensitivity to extremal values results in making readily apparent even a single large error. Still, SQPnL remains superior in this metric as well.

Further to the reprojection error, the errors for the estimated poses were also evaluated. Specifically, the rotation error for a true camera rotation  $\mathbf{R}_g$  and

<sup>3</sup> SQPnL employed maximum iterations  $T = 15$  and tolerance  $\epsilon = 10^{-7}$ .

**Table 1.** Average and median execution times (in ms) of several PnL solvers implemented in Matlab, computed across all executions for every  $4 \leq n \leq 20$ .

		(ours)	PnL Solver								
		SQPnL	Ansar	AlgLS	DLT	LPnL_Bar_LS	RPnL	ASPnL	SRPnL	OPnL	HPnL
Time	Mean	4.91	14.94	12.48	1.02	1.35	3.36	7.41	10.60	7.69	16.38
	Median	4.52	10.41	12.20	1.00	1.35	3.37	7.41	10.81	7.50	16.65

estimate  $\mathbf{R}_e$  is the angle of rotation about a unit vector that transfers  $\mathbf{R}_e$  to  $\mathbf{R}_g$ , given by  $\arccos((\text{trace}(\mathbf{R}_g \mathbf{R}_e^T) - 1)/2)$ . The translation error for a true translation  $\mathbf{t}_g$  and for an estimate  $\mathbf{t}_e$  is simply the Euclidean norm of their vector difference  $\|\mathbf{t}_g - \mathbf{t}_e\|$ . Figures 4 and 5 illustrate respectively the rotation and translation errors corresponding to the same experiments of Fig. 3. The rotational errors pertaining to SQPnL are the lowest, albeit often being indistinguishable from those of AlgLS [22]. A noteworthy observation for the maximum rotational error plots in Fig. 4(b) is that some methods occasionally yield erroneous estimates with errors as large as  $\pi$  rad, which in spite of being smoothed out by averaging in 4(a), are accentuated by the maximum operator in 4(b). This is more pronounced for DLT and ASPnL, and to a lesser degree for RPnL. With respect to translation, SQPnL clearly performs best, both in terms of the average and maximum translational errors.

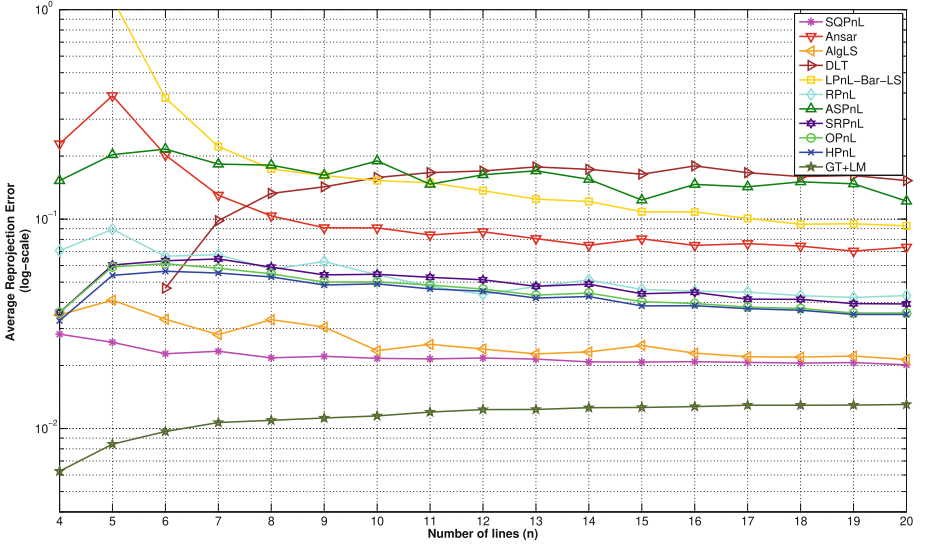
Timing measurements for the PnL methods under comparison are also provided. It is noted that they are all implemented in Matlab which is known not to favor time efficiency. Still, time comparisons can be indicative of the method’s relative performance. Execution times are provided in Table 1, showing that SQPnL is competitive also in terms of computational cost. Similarly to SQPnP, the execution time of SQPnL is dominated by the linear system solution invoked in every SQP iteration. Our current implementation employs Matlab’s general-purpose linear system solver `linsolve`. However, this operation can be considerably accelerated by exploiting the special structure of the system’s matrix. This optimization has been incorporated into the C++ implementation of SQPnP<sup>4</sup>, resulting in a tenfold improvement in performance.

## 5.2 Real image experiments

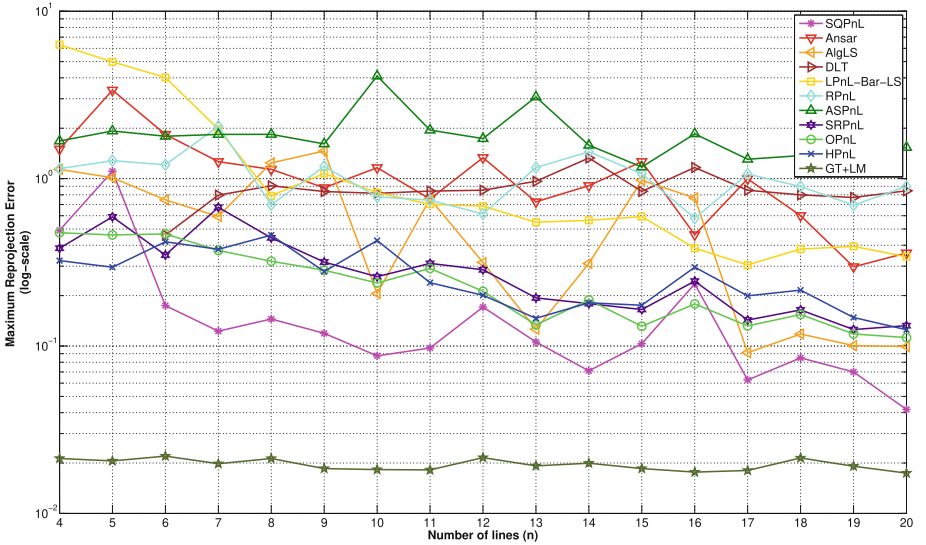
SQPnL was also evaluated with the aid of real images from VGG’s multi-view dataset<sup>5</sup>. Among others, this dataset includes 17 Oxford University building images, taken with a digital camera held at a person’s height. The images are organized into 5 sets and in addition to them, the dataset also includes matched 2D line segments along with their reconstructed 3D counterparts and camera projection matrices. We used the 3D lines and their corresponding 2D line segments from all 17 images as inputs to PnL estimation, whereas the poses extracted from the camera matrices were used as ground truth. SQPnL was compared

<sup>4</sup> Code can be accessed [here](#).

<sup>5</sup> <https://www.robots.ox.ac.uk/~vgg/data/mview/>



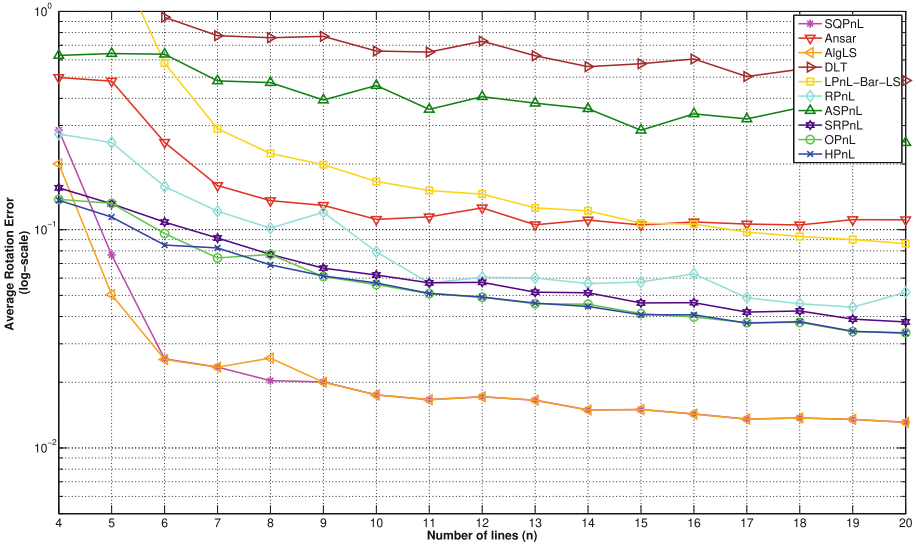
(a) average reprojection error



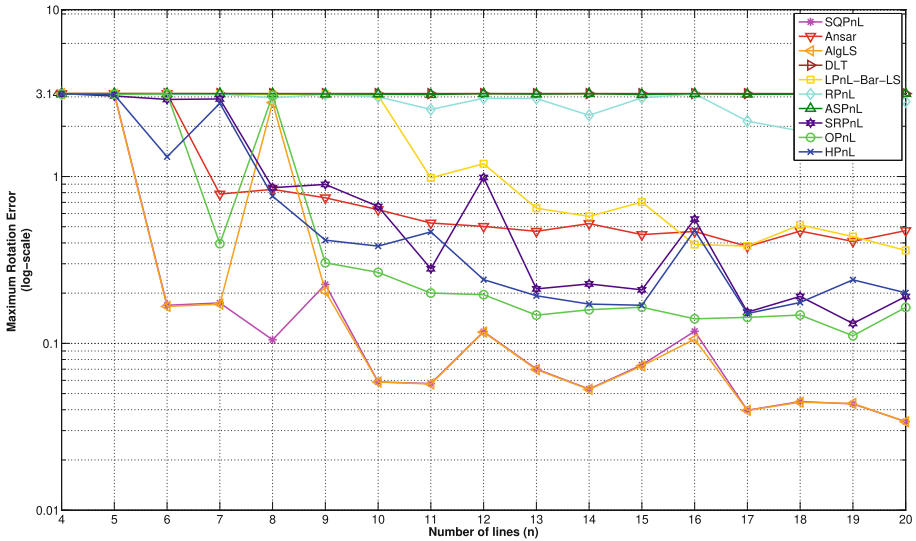
(b) maximum reprojection error

**Fig. 3.** Plots of the average and maximum reprojection error for 500 executions of each PnL solver on  $n$  random line segments,  $4 \leq n \leq 20$ . For each  $n$ , the line segments are repeatedly generated from a previously sampled line population and their endpoints are contaminated with additive Gaussian noise of standard deviation  $\sigma_{\epsilon} = 5$  pixels. Notice the different scales in the vertical axes of the plots.



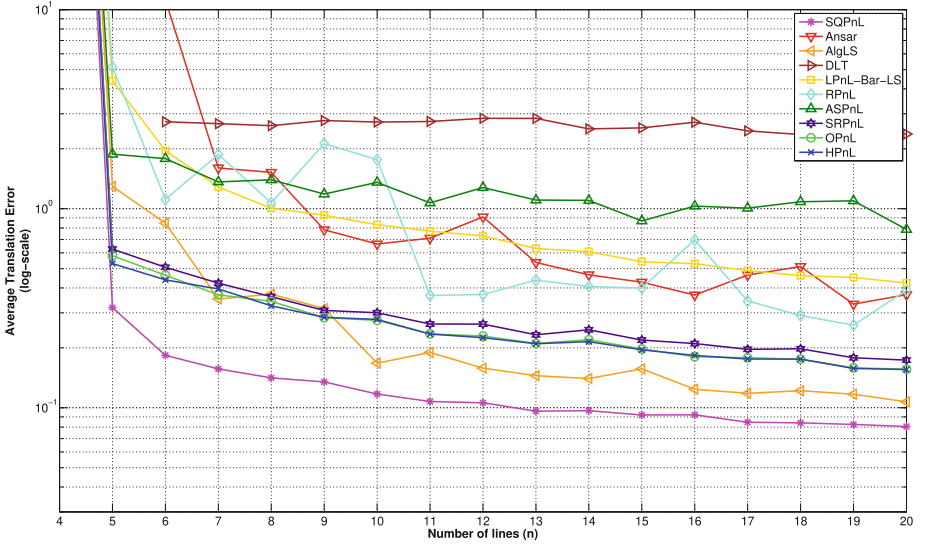


(a) average rotation error

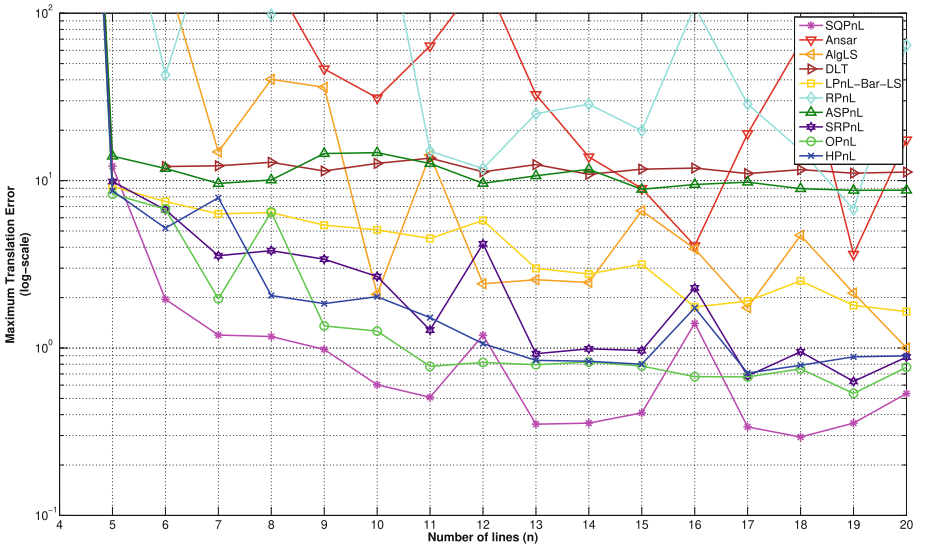


(b) maximum rotation error

**Fig. 4.** Plots of the average and maximum rotation error (in radians) for the experiments of Fig. 3. Note that methods DLT, ASPnL and RPnL have errors of  $\pi$  in the maximum error plots in (b). This is because they occasionally provide grossly erroneous estimates which are highlighted by the maximum operator, despite being attenuated by the averaging in (a). Another observation is that all methods have returned at least one highly erroneous estimate for  $n$  equaling 4 and 5.



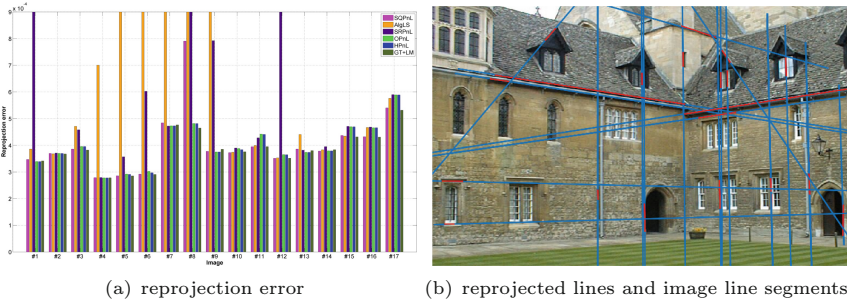
(a) average translation error



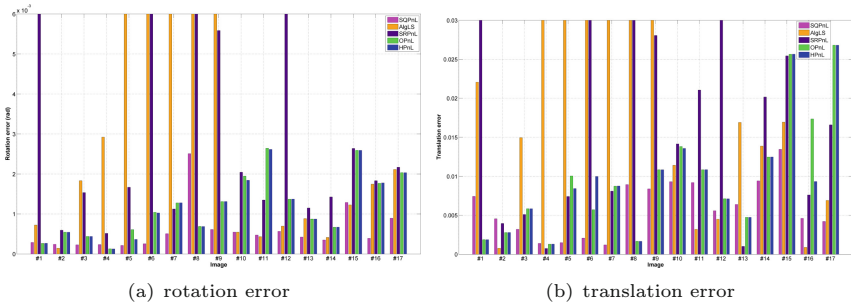
(b) maximum translation error

**Fig. 5.** Plots of the average and maximum translation error (in meters) for the experiments of Fig. 3. SQPnL clearly surpasses all other methods.

with the best performing state-of-the-art methods, as identified from the synthetic experiments in Sect. 5.1, i.e. AlgLS, SRPnL, OPnL and HPnL. Similarly to the synthetic experiments, we include in Fig. 6(a) a bar plot of the reprojection errors using the pose estimated by each method. Figure 6(b) visualizes



**Fig. 6.** (a) Bar plots of reprojection error for SQPnL and selected PnL methods applied to the 17 VGG building images. (b) Illustration of certain reprojected lines (blue) in comparison with their corresponding line segments detected in an image (red).



**Fig. 7.** Bar plots of the rotation and translation errors for SQPnL and selected PnL methods applied to the 17 VGG building images.

certain 3D lines reprojected with the estimated pose on an image. Furthermore, Fig. 7 illustrates the rotation and translation errors with respect to the ground truth, computed using the same metrics as in Sect. 5.1. Clearly, the proposed method performs better than the competing ones in the majority of cases.

## 6 Conclusions

This paper has presented a non-minimal PnL solver that employs a quadratic cost penalizing plane normal misalignment on the elements of the rotation matrix. Utilizing existing machinery for solving QCQP formulations addressing the PnP, this cost is minimized by conducting local searches in the vicinity of special feasible points from which the global minima are located in a few steps. The translation is recovered via solving a linear least squares problem. Comparative experiments have confirmed that the solver consistently recovers the correct pose, attaining accuracy that exceeds or at least matches the best competing methods across various levels of noise and spatial arrangements of 3D lines. A C++ implementation of the method is available at <https://github.com/terzakig/sqpnl>.

**Acknowledgements.** This work was partially funded by the EU’s H2020 & Horizon Europe research and innovation programmes under GAs 101017151 (FELICE) and 101120990 (SOPRANO).




## References

1. Agostinho, S., Gomes, J., Del Bue, A.: CvxPnPL: a unified convex solution to the absolute pose estimation problem from point and line correspondences. *J. Math. Imaging Vision* **65**(3), 492–512 (2023)
2. Ansar, A., Daniilidis, K.: Linear pose estimation from points or lines. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 578–589 (2003)
3. Caglioti, V.: The planar three-line junction perspective problem with application to the recognition of polygonal patterns. *Pattern Recogn.* **26**(11), 1603–1618 (1993)
4. Chebrolu, N., Låbe, T., Vysotska, O., Behley, J., Stachniss, C.: Adaptive robust kernels for non-linear least squares problems. *IEEE Rob. Autom. Lett.* **6**(2), 2240–2247 (2021)
5. Chen, H.: Pose determination from line-to-plane correspondences: existence condition and closed-form solutions. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(6), 530–541 (1991)
6. Chia, A.Y.S., Rajan, D., Leung, M.K., Rahardja, S.: Object recognition by discriminative combinations of line segments, ellipses, and appearance features. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(9), 1758–1772 (2012)
7. Dennis, J., Schnabel, R.: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics, SIAM (1996)
8. Dhome, M., Richetin, M., Lapreste, J.T., Rives, G.: Determination of the attitude of 3d objects from a single perspective view. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(12), 1265–1278 (1989)
9. Grompone von Gioi, R., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a line segment detector. *Image Process. Line* **2**, 35–55 (2012)
10. Hartley, R.I.: Chirality. *Int. J. Comput. Vision* **26**(1), 41–61 (1998)
11. Hartley, R.I., Sturm, P.: Triangulation. *Comput. Vis. Image Underst.* **68**(2), 146–157 (1997)
12. Hofer, M., Maurer, M., Bischof, H.: Efficient 3D scene abstraction using line segments. *Comput. Vis. Image Underst.* **157**, 167–178 (2017)
13. Kamgar-Parsi, B., Kamgar-Parsi, B.: Algorithms for matching 3D line sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(5), 582–593 (2004)
14. Kumar, R., Hanson, A.: Robust methods for estimating pose and a sensitivity analysis. *CVGIP: Image Underst.* **60**(3), 313–342 (1994)
15. Lebeda, K., Matas, J., Chum, O.: Fixing the locally optimized RANSAC. In: *Proceedings of the British Machine Vision Conference*, pp. 95.1–95.11 (2012)
16. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: an accurate  $O(n)$  solution to the PnP problem. *Int. J. Comput. Vision* **81**(2), 155 (2009)
17. Liu, S., Yu, Y., Pautrat, R., Pollefeys, M., Larsson, V.: 3D line mapping revisited. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445–21455 (2023)
18. Liu, Y., Huang, T., Faugeras, O.: Determination of camera location from 2-D to 3-D line and point correspondences. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(1), 28–37 (1990)
19. Lourakis, M., Pateraki, M.: Markerless visual tracking of a container crane spreader. In: *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2579–2586 (2021)

20. Lourakis, M., Terzakis, G.: A globally optimal method for the PnP problem with MRP rotation parameterization. In: International Conference on Pattern Recognition (ICPR), pp. 3058–3063 (2020)
21. Lourakis, M., Zabulis, X.: Model-based pose estimation for rigid objects. In: Chen, M., Leibe, B., Neumann, B. (eds.) ICVS 2013. LNCS, vol. 7963, pp. 83–92. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-39402-7\\_9](https://doi.org/10.1007/978-3-642-39402-7_9)
22. Mirzaei, F.M., Roumeliotis, S.I.: Globally optimal pose estimation from line correspondences. In: IEEE International Conference on Robotics and Automation, pp. 5581–5588 (2011)
23. Nakano, G.: Globally optimal DLS method for PnP problem with Cayley parameterization. In: British Machine Vision Conference, pp. 78.1–78.11 (2015)
24. Příbyl, B., Zemčík, P., Čaik, M.: Absolute pose estimation from line correspondences using direct linear transformation. *Comput. Vis. Image Underst.* **161**, 130–144 (2017)
25. Silva, M., Ferreira, R., Gaspar, J.A.: Camera calibration using a color-depth camera: points and lines based DLT including radial distortion. In: IROS Workshop on Color-Depth Camera Fusion in Robotics (2012)
26. Terzakis, G., Lourakis, M.: A consistently fast and globally optimal solution to the perspective-n-point problem. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 478–494. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58452-8\\_28](https://doi.org/10.1007/978-3-030-58452-8_28)
27. Terzakis, G., Lourakis, M., Ait-Boudaoud, D.: Modified Rodrigues parameters: an efficient representation of orientation in 3D vision and graphics. *J. Math. Imaging Vision* **60**(3), 422–442 (2018)
28. Wang, P., Chou, Y., An, A., Xu, G.: Solving the PnL problem using the hidden variable method: an accurate and efficient solution. *Vis. Comput.* **38**(1), 95–106 (2022)
29. Wang, P., Xu, G., Cheng, Y.: A novel algebraic solution to the perspective-three-line pose problem. *Comput. Vis. Image Underst.* **191**, 102711 (2020)
30. Wang, P., Xu, G., Cheng, Y., Yu, Q.: Camera pose estimation from lines: a fast, robust and general method. *Mach. Vis. Appl.* **30**(4), 603–614 (2019)
31. Wang, Q., Yan, Z., Wang, J., Xue, F., Ma, W., Zha, H.: Line flow based simultaneous localization and mapping. *IEEE Trans. Rob.* **37**(5), 1416–1432 (2021)
32. Xu, C., Zhang, L., Cheng, L., Koch, R.: Pose estimation from line correspondences: a complete analysis and a series of solutions. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1209–1222 (2017)
33. Yang, H., Antonante, P., Tzoumas, V., Carlone, L.: Graduated non-convexity for robust spatial perception: from non-minimal solvers to global outlier rejection. *IEEE Rob. Autom. Lett.* **5**(2), 1127–1134 (2020)
34. Yu, Q., Xu, G., Cheng, Y.: An efficient and globally optimal method for camera pose estimation using line features. *Mach. Vis. Appl.* **31**(6), 1–11 (2020). <https://doi.org/10.1007/s00138-020-01100-6>
35. Zhang, L., Xu, C., Lee, K.-M., Koch, R.: Robust and efficient pose estimation from line correspondences. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012. LNCS, vol. 7726, pp. 217–230. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-37431-9\\_17](https://doi.org/10.1007/978-3-642-37431-9_17)
36. Zheng, Y., Kuang, Y., Sugimoto, S., Åström, K., Okutomi, M.: Revisiting the PnP problem: a fast, general and optimal solution. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2344–2351 (2013)



# Wavefront Neural Radiance Fields for Multi-depth Reconstruction

Tsubasa Nakamura<sup>1</sup>, Ken Sakurada<sup>2</sup> , and Gaku Nakano<sup>1</sup>  

<sup>1</sup> NEC Corporation, 1753 Shimonumabe, Kawasaki, Japan  
tsubasa-nakamura@nec.com, g-nakano@nec.com

<sup>2</sup> National Institute of Advanced Industrial Science and Technology, 2-3-26 Aomi,  
Tokyo, Japan  
k.sakurada@aist.go.jp

**Abstract.** This paper proposes a novel NeRF (Neural Radiance Fields), called WF-NeRF, for accurately recovering wavefront signals with multiple depth values. Although range sensors such as LiDARs typically return depth as scalars, the proposed method uses 1D raw signals of LiDARs, i.e., all reflected signals of multiple objects in the path of a beam. Due to this, coarse sampling used in the conventional NeRFs is no longer required but only a single-pass sampling, thus improving learning and memory efficiency. Considering the property of LiDAR signals, where the signal intensity decays inversely proportional to the square of the distance as the beam light spreads over the wavefront, we introduce a new sampling strategy of the same distance signals on the wavefront and a loss function taking the relative error between the training and predicted values (MSRE: Mean Squared Relative Error). The wavefront sampling produces super-resolution-like effects and improves the accuracy of multiple-depth estimation. MSRE normalizes the decay of the observed signals and stabilizes the learning process. In experiments with an object occluded by a mesh, we show that the conventional NeRFs fail to reconstruct the 3D shape. On the other hand, the proposed WF-NeRF accurately recovers both the mesh and the object, even with a smaller number of input data.

**Keywords:** Neural Radiance Fields · LiDAR · Wavefront Signals

## 1 Introduction

With the widespread use of inexpensive LiDARs, automatic navigation technology is becoming more and more practical in robotics research. The LiDAR calculates distance based on Time-of-Flight, i.e., the time of transmission and reception of a laser beam with a finite thickness. Due to data stream capacity limitations, many commercial LiDAR systems can only provide the distance to a single representative point on the beam path. However, actual scenes consist of complex structures and objects that partially obstruct the beam. For example,

inexpensive low-resolution LiDARs cannot observe objects behind a grid pattern with 2–3 cm spacing, e.g., fences or nets. The degradation of map accuracy due to such commonplace objects is a fatal problem in automated navigation. For example, when a robot scans an indoor scene using a LiDAR, the LiDAR beams interfere with complex objects placed in front of the LiDAR, such as nets and houseplants. The sensor receives stronger reflections from closer structures, so the shape of background objects will be discarded, causing holes in the 3D map. The same problem occurs when using LiDAR together with other sensors in remote sensing or autonomous driving. Since LiDAR beams first interfere with branches, leaves, and fences, signals of the terrain in the distance are prevented.

On the other hand, 3D scene reconstruction from images is one of the major applications in computer vision and has been studied intensively for decades. The most fundamental theory is Structure-from-Motion (SfM) [7], which simultaneously estimates the 3D positions of image feature points and the camera motion. Since the density of the 3D point cloud obtained by SfM is sparse, various approaches have been devoted to recovering dense 3D shapes. Classical methods include Novel View Synthesis [1], which transfers pixel values to a new viewpoint using the camera position from SfM; Visual Hull [10], which estimates a 3D voxel from the silhouette of an object; Multi View Stereo [6], which extends the stereo theory to multiple viewpoints; and DTAM [14], which generates smooth and dense 3D surfaces on real-time using RGB cameras. Recently, Neural Radiance Fields (NeRF) [12] has attracted attention as an innovative approach for obtaining photorealistic novel views from RGB images alone.

Since the original NeRF (hereinafter referred to as Vanilla-NeRF) has a simple structure to learn MLP (Multi-layer Perceptron) by minimizing the photometric loss using a large number of images, various derivative techniques have been proposed. For example, reducing the number of images using image features (PixelNeRF [18], DietNeRF [8]), recovering microstructures in high resolution by extending the sampling range from lines to frustum (mip-NeRF [2], zip-NeRF [3]), accelerating training time by hashing positional encoding (Instant-NGP [13]), and improving memory efficiency by gridding the space using tensor decomposition (TensoRF [4]). These studies focus on generating visually natural images, often resulting in inaccurate 3D shapes.

Several studies have been proposed to incorporate 3D data measured by LiDARs or RGB-D cameras into a NeRF framework to recover highly accurate 3D shapes. UrbanRF [15] and Point-NeRF [17] use point clouds, and DS-NeRF [5] and D-Nerfacto [16] use depths. Those works showed that utilizing 3D information mitigates the inaccuracy around object boundaries. However, they treat LiDAR signals as mere zero-dimensional signals or scalar values. In other words, they ignore the characteristics of LiDARs, which irradiate a beam with a certain width and output a depth with the highest reflectance in the beam luminous flux. If multiple objects smaller than the beam’s spot size are on the beam path, only a single object with the highest reflectivity may be reconstructed.

This paper aims to present a method for incorporating the aforementioned property of LiDAR signals into NeRF to achieve highly accurate 3D shape reconstruction<sup>1</sup>. The contributions of this paper, outlined in Fig. 1, are as follows:

- **1-pass IFM sampling**

Raw signals from a LiDAR, distributing 1-dimensional in each depth direction, are used as the ground-truth data to determine sample points for ray marching. We apply the IFM (Inverse Function Method) to the GT signal at each iteration to add randomness to the sampling process and achieve high spatial resolution and memory efficiency. This step corresponds to a direct computation of coarse sampling in Vanilla-NeRF without rendering.

- **Wavefront sampling and accumulation**

We generate multiple rays within the spot of a LiDAR beam light flux. We conduct ray marching on these rays at one time and calculate a 1D signal by integrating points at the same distance from the center of the LiDAR. In the field of optics, this is equivalent to obtaining signals on the same wavefront.

- **Mean Squared Relative Error (MSRE)**

The 1D signals predicted in the wavefront sampling described above are used to calculate Loss against the training signals, i.e., the raw signals. We calculate the relative error of the predictions to the training signals to remove the inverse-square falloff.

## 2 Related Works

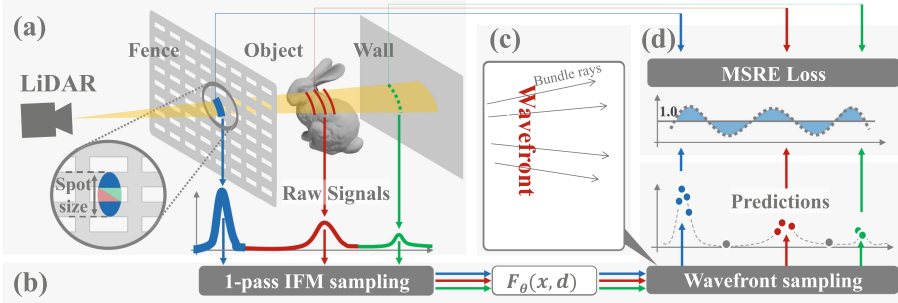
### 2.1 LiDAR Sensors

LiDARs internally record the time of flight between the transmitter’s emission of the light signal and the receiver’s reception of the reflected light from the object. After the transmitter emits a beam, the receiver continues to record the signal of the reflected light from an object for a certain period of time. It determines a representative value from the time-resolved signal. Many LiDARs on the market output a single depth with the transmitter’s direction for each point; thus, the 3D structure of a scene or an object is obtainable as a point cloud. While point cloud representation discretizes 3D space, it has several limitations about the sparsity of points and the memory inefficiency for large-scale scenes. Moreover, raw signals of LiDARs before representative depth is determined actually contain *rich* information, i.e., the reflected radiance of multiple objects on the path of beams. Some commercial products return raw signals or convert them into multi-return point clouds; however, NeRF-based approaches have not been explored extensively to handle such special signals or point clouds. These facts motivate us to develop the novel NeRF to handle wavefront 1D signals.

---

<sup>1</sup> Transient-NeRF [11], based on a similar motivation, was proposed on arXiv. However, it has yet to be peer-reviewed and code has not been released.





**Fig. 1.** An overview of the proposed WF-NeRF. (a) A beam emitted from a LiDAR sensor interferes with a mesh-like fence, an object, and a wall. The object and the wall reflect the beam passing through a gap in the mesh. The LiDAR receives those reflections. (b) The 1D raw signal of each beam has multiple reflected intensities at the interference distance with the object. The IFM is applied to their 1D distribution to select sampling points efficiently. The neural network predicts 1D signals from the Radiance Fields  $F_\theta$  of the sampling points. (c) Multiple rays are randomly flown within each beam for sampling to simulate the actual beam spread of the LiDAR. The predicted signals at the same distance of each ray are accumulated along the wavefront direction. (d) The difference between the raw signals and the predictions is used as a loss in the training process. Signals from relatively dark objects, i.e. weak reflections from distant objects, are normalized by taking the MSRE loss between the raw and predicted signals.

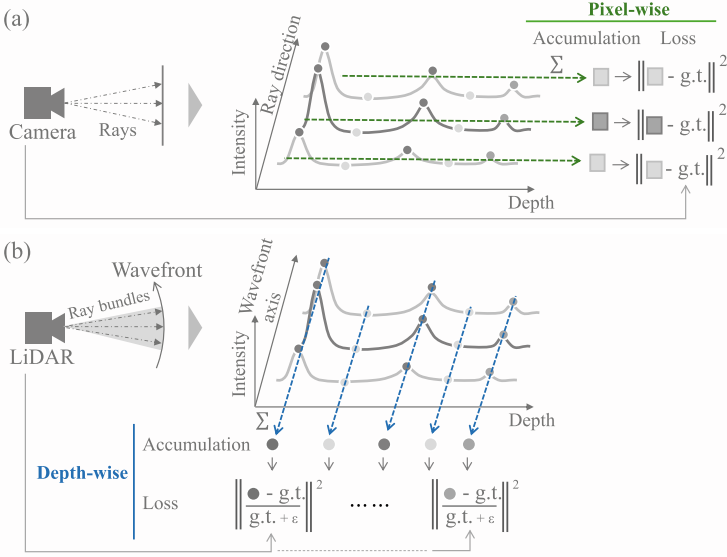
## 2.2 NeRF Utilizing LiDAR/Depth Sensors

DS-NeRF [5] is the first method that incorporate dense depth maps into the Vanilla-NeRF framework. The depth loss is minimized between the ground-truth and predictions. D-Nerfacto [16] uses a strategy similar to DS-NeRF to minimize the depth loss, but its framework is based on Nerfacto for faster computation. UrbanRF [15] introduces the LiDAR loss, which uses point clouds to actively reduce the volume density in empty regions in 3D space. The LiDAR loss improves convergence and accuracy. Point-NeRF [17] proposes neural point clouds for improving training efficiency. 3D neural points are initialized by LiDAR or SfM, then, a pruning algorithm is performed on 3D points having a low confidence. As these methods only assume single-return signals, using multi-return signals has yet to be investigated well. In Sect. 4, we will report that extending the conventional approaches to multi-return signals does not yield good reconstruction results.

## 3 WF-NeRF: Wavefront Neural Radiance Fields

### 3.1 Overview

Figure 1 shows a conceptual diagram of the proposed method. As described in Sect. 2.1, many commercial LiDARs convert the 1D raw signal of each beam to



**Fig. 2.** Sampling strategies of the conventional NeRFs and the proposed WF-NeRF. (a) Color sampling by the conventional NeRFs. Each pixel value is integrated along the ray direction. (b) Wavefront sampling by the proposed WF-NeRF. Each depth value within a beam is integrated along the wavefront direction, i.e. the same distance from the sensor.

a scalar depth to output the 3D structure as a point cloud, while the raw data contains rich information such as multiple reflections. The proposed method aims to achieve more accurate sensing by directly inputting raw signals to a Neural Radiance Field. The following sections describe the three techniques to recover multiple depths of the raw signals in a NeRF framework.

### 3.2 1-Pass IFM Sampling

We customize the inverse function method, which was originally developed in Vanilla-NeRF [12] and widely used in RGB-based variants, to reduce the amount of processing data. The inverse function method is a Coarse-to-Fine approach for sampling points of the 3D space. This method probabilistically selects sampling points in proportion to the slope of the cumulative density function (CDF) calculated from a probability density function (PDF). Vanilla-NeRF allocates many points in a region with high-intensity reflections where an object is supposed to exist. The 2-pass sampling requires a neural rendering for each coarse and fine model. In the proposed method, on the other hand, a LiDAR raw signal corresponds to the 1D density distribution along a ray. Therefore, the sampling point distribution can be determined based on the raw signal, which is then applied to the IFM to obtain fine sampling directly.

The proposed method has the advantage of performing neural rendering only once; however, LiDAR raw signals are more dense than depth maps. The amount of data is enormous and memory inefficient to be used as the ground-truth signal for training a NeRF model. We propose an efficient sampling method based on the conventional approach to overcome this issue.

The intensity of raw signals is close to zero (apart from internal and disturbance noise) around an empty area where no objects exist and no reflections are observed. As UrbanRF [15] and DS-NeRF [5] pointed out, those information give a strong constraint for estimating the distribution of a sparse space. Therefore, the spatial distribution can be efficiently estimated by adding moderate randomness to a region with zero-intensity signals at spatial sampling. Our sampling strategy is as follows:

1. Add random bias of 1–10% of the maximum signal to the raw signals before applying the IFM.
2. Apply the IFM at each iteration of a training phase to obtain random sampling points.

In contrast to Vanilla-NeRF, the PDF in the proposed method is a LiDAR raw signal added a weak bias. This leads to a gradient for the whole region in the depth direction when the PDF is converted to a CDF. When performing ray marching at every iteration of a training process, we can obtain random sampling points with a certain probability even at points near zero intensity. The proposed 1-pass IFM approach reduces memory consumption drastically. We will describe actual reduction rate in the experiment section, Sect. 4.2.

### 3.3 Wavefront Sampling and Accumulation

The light emitted by LiDARs is a beam with a finite thickness. Therefore, as shown in Fig. 1, when a beam interferes with multiple objects, the raw signal contains information about the objects’ surface. We aim to reproduce this physical model with Radiance Fields to recover object shape. Figure 2 illustrates the difference in the integration direction between the conventional NeRFs and the proposed method. Ray marching in the conventional NeRFs is a 1D integration along a ray (Fig. 2a). Conversely, the proposed method integrates in the wavefront direction perpendicular to the ray, as shown in Fig. 2b. We first set a cross-section (spot) of a ray bundle that simulates the signal of a LiDAR beam at a particular azimuth. Then, we randomly generate  $n$  rays passing through the cross-section and perform ray marching for all rays as in Vanilla-NeRF to obtain the intensity of each sampling point.

Similar to conventional NeRFs methods, a network that learns the intensity  $c$  and volume density  $\sigma$  of a 3D point  $\mathbf{x}$  viewed from a direction  $\mathbf{d}$  can be expressed as  $F_\theta(\mathbf{x}, \mathbf{d}) \rightarrow (c, \sigma)$ . In our method, we extended the intensity vector  $c$  from 3 dimensions of RGB to 4 dimensions of RGBI and used the fourth dimension as the intensity of beam refraction of the LiDAR. Note that in the main experiments, these RGB values are not used for training, but only in Sect. 4.2.

Suppose that a beam is composed of  $n$  rays. The signal intensity  $I_{\text{pred}}$  at  $\mathbf{x}_i$  on a beam with the distance  $r$  from the origin  $\mathbf{o}_i$  is obtained as follows:

$$I_{\text{pred}}(\mathbf{x}(r)) = \frac{1}{n} \sum_i^n I_i(\mathbf{x}(r)), \quad (1)$$

where

$$\begin{aligned} I_i(\mathbf{x}(r)) &= T(r) \sigma(\mathbf{x}_i(r)) c(\mathbf{x}_i(r), \mathbf{d}), \\ \mathbf{x}_i(r) &= \mathbf{o}_i + r\mathbf{d}_i, \\ T(r) &= \exp\left(-\int_0^r \sigma(\mathbf{x}_i(s)) ds\right). \end{aligned} \quad (2)$$

Note that  $T(r)$  represents the transmittance at the distance  $r$ . Unlike Vanilla-NeRF, the proposed method does not integrate  $I_i(\mathbf{x}(r))$  in the ray marching direction but treats it as a 1D intensity distribution w.r.t. the distance  $r$ . Suppose the intensities  $I_i(\mathbf{x}(r)), i \in n$  are obtained at the same interval within the beam flux. The distance  $r$  can be interpreted as a wavefront equidistant from the LiDAR. Thus,  $I_{\text{pred}}(\mathbf{x}(r))$  can be represented by the average of the intensities  $I_i(\mathbf{x}(r)), i \in n$ .

### 3.4 Mean Squared Relative Error (MSRE)

The proposed method calculates the loss for every distance  $r$  between the raw signal  $I_{\text{raw}}(r)$  and the predicted signal  $I_{\text{pred}}(r)$ . It should be noted here that raw signals of LiDARs decay in inverse proportion to the square of the distance. Ignoring the inverse-square falloff causes an exponential weighting by the distance, resulting in the recovery of the nearest object being prioritized, and distant objects may be lost. To overcome this issue, we employ the mean squared relative error to ensure uniform weighting independent of distance from the sensor:

$$L_{\text{LiDAR}} = \left( \frac{I_{\text{pred}}(r) - I_{\text{raw}}(r)}{I_{\text{raw}}(r) + \epsilon} \right)^2, \quad (3)$$

where  $\epsilon = 10^{-9}$  is merely a small value to avoid zero division.

## 4 Experiments

This section reports the performance evaluation of the proposed method. First, using synthetic data, we conducted an ablation study on the new wavefront sampling and MSRE loss function as well as on different backbone models. Next, quantitative and qualitative comparisons were made with the conventional NeRFs, also using synthetic data. We will visually validate the reconstruction results for real data captured by a consumer LiDAR. Finally, we will demonstrate an example of sensor fusion combining WF-NeRF with an RGB camera.

## 4.1 Implementations

**The Proposed Method.** The new approaches described in Sect. 3 can be incorporated with various NeRFs based on the standard ray marching rendering as a backbone model. Hereafter, we call the proposed method combined with Vanilla-NeRF, mip-NeRF, TensorRF and Nerfacto by WF-NeRF-V, WF-NeRF-M, WF-NeRF-T and WF-NeRF-N, respectively. The number of samplings of Vanilla-NeRF and mip-NeRF were set to 512. The resolution of TensorRF was initially set to 256 and finally to 1024. The number of rays forming one ray bundle in the wavefront sampling was set to 128, corresponding to the number of rays sampled in the ray marching in the fine network of Vanilla-NeRF. We implemented WF-NeRF on NerfStudio [16], a unified framework for NeRF model training and visualization.

**The Conventional Methods.** We chose three conventional NeRF methods using RGB cameras: Vanilla-NeRF [12], mip-NeRF [2], and TensorRF [4]. Also, we chose two methods using depth values: DS-NeRF [5] and D-Nerfacto [16]. Since DS-NeRF and D-Nerfacto assume point clouds or depth maps rather than LiDAR raw signals, we input the maximum value of raw signals in the depth direction. Unlike LiDAR raw signals, the depth map has the format limitation that only one representation depth can be obtained for a given coordinate (direction). Therefore, for fair comparisons, we prepared “2-layer depth” dataset, where each depth map has two depth channels (the front and back of the fence). We used the official DS-NeRF implementation and NerfStudio for the other methods.

**Evaluation Metrics.** Following the conventional methods, we used MAE (Mean Absolute Error) to evaluate the accuracy of 3D data.

**Table 1.** Ablation study of the proposed method using 45 LiDARs. WF-sampling denotes the wavefront sampling proposed in Sect. 3.3. The bold and underlined numbers depict the best and the second-best values for each column.

Model	Backbone	WF-sampling (Sect. 3.3)	Loss (Sect. 3.4)	Depth MAE↓
WF-NeRF-T	TensorRF		MSE	2.66
	TensorRF	✓	MSE	2.59
	TensorRF		MSRE	0.87
	TensorRF	✓	MSRE	<u>0.62</u>
WF-NeRF-N	Nerfacto		MSE	3.00
	Nerfacto	✓	MSE	2.78
	Nerfacto		MSRE	0.37
	Nerfacto	✓	MSRE	<b>0.18</b>

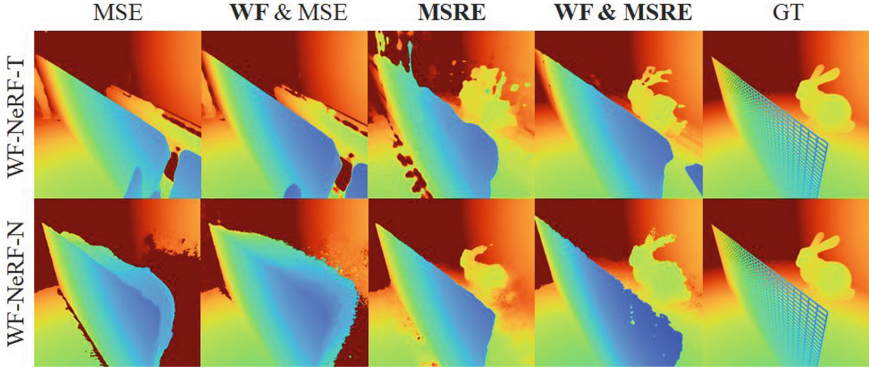


Fig. 3. Visual comparison of the ablation study: depth map prediction.

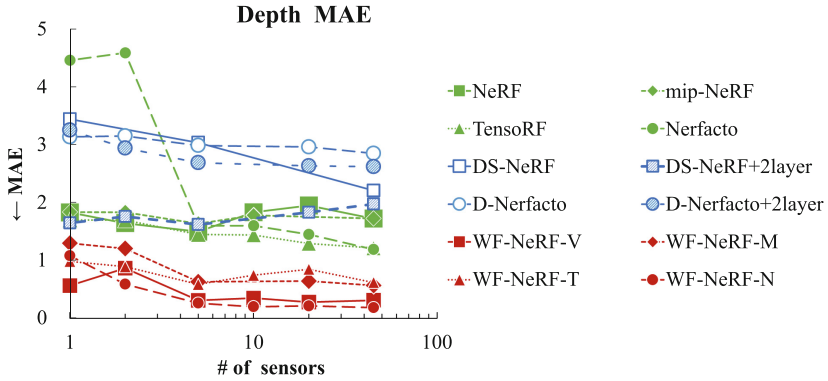


Fig. 4. Experimental results on synthetic data w.r.t. the number of sensors.

## 4.2 Synthetic Dataset

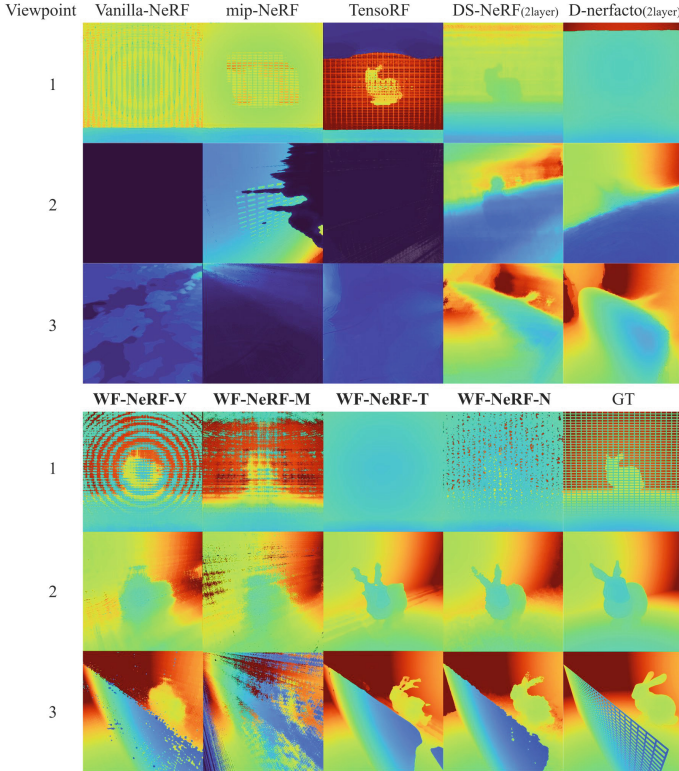
We generated LiDAR raw signals using a transient light simulator developed by Jarabo et al. [9]. In this simulation, all objects are set with a single-color Lambertian reflection model, providing very little texture information. Additionally, the simulation takes into account the spread of LiDAR beam spots, which are larger than the frustums defined by each pixels of an RGB camera. Due to this, the generated depth maps have lower spatial resolution than that of RGB camera. The LiDAR beam spot was a portrait style whose aspect ratio is 3:1, which we determined corresponding to the specification of Livox Mid-40 in the real data experiments. See Sect. 4.3 for more details. As shown in Fig. 1, a mesh-like fence was placed in front of an object with a grid spacing equal to the vertical width of the LiDAR beam spot. We used the Stanford bunny for the object. We set 45 viewpoints for each LiDAR or RGB camera in front of the fence to generate training data, and generated test images by placing 41 viewpoints behind the fence. The image resolution  $(x, y)$  is  $400 \times 400$  while the LiDAR raw signals

which have 3D resolution  $(x, y, z)$  were rendered by  $400 \times 400 \times 1600$  for a single view. The third parameter  $z$  here represents the range (depth) from the sensor. We set the number of points by 128 for the 1-pass IFM sampling described in Sect. 3.2. The depth resolution was reduced from 1600 to 128, which is 92% data reduction (1 GB per view to 82 MB per view). The computational time using the 1-pass IFM was about 3–4 h for training a sequence on a single RTX3080 GPU. We could not train a model without the 1-pass IFM because the amount of data exceeds the 12 GB VRAM of the GPU.

**Ablation Study.** We start with ablation studies to validate the effectiveness of the two proposed techniques: the wavefront sampling (WF-sampling) in Sect. 3.3 and MSRE in Sect. 3.4. In this experiment, we used 45 LiDAR views, WF-NeRF-T and WF-NeRF-N. As comparisons of MSRE for the proposed method, we used the conventional method MSE (Mean Squared Error) that uses absolute value of the signal for them. Table 1 indicates that WF-sampling and MSRE significantly improve depth estimation accuracy. Also, Fig. 3 shows the qualitative results with and without WF-sampling and MSRE. We can clearly see that the proposed approaches are required to reconstruct the fence and object. Interestingly, WF-NeRF-T has a smoother 3D shape, while WF-NeRF-N is more detailed but with tiny artifacts. This is caused by the difference in the spatial representation of the two backbones.

**Table 2.** Quantitative results on synthetic data.

Method		# of sensors		Depth MAE↓
		LiDAR	RGB cam.	
Conventional	Vanilla-NeRF	0	45	1.72
	mip-NeRF	0	45	1.72
	TensoRF	0	45	1.21
	DS-NeRF	45	0	2.20
	DS-NeRF (2-layer)	45	0	1.97
	D-Nerfacto	45	0	2.84
	D-Nerfacto (2-layer)	45	0	2.61
Proposed	WF-NeRF-V	45	0	0.31
		1	0	0.56
	WF-NeRF-M	45	0	0.57
		1	0	1.29
	WF-NeRF-T	45	0	0.62
		1	0	0.98
	WF-NeRF-N	45	0	0.18
		1	0	1.08



**Fig. 5.** Qualitative comparison of the depth map estimation on the synthetic dataset. Each column is the estimation result by each method. Each row displays the results from a different view angle.

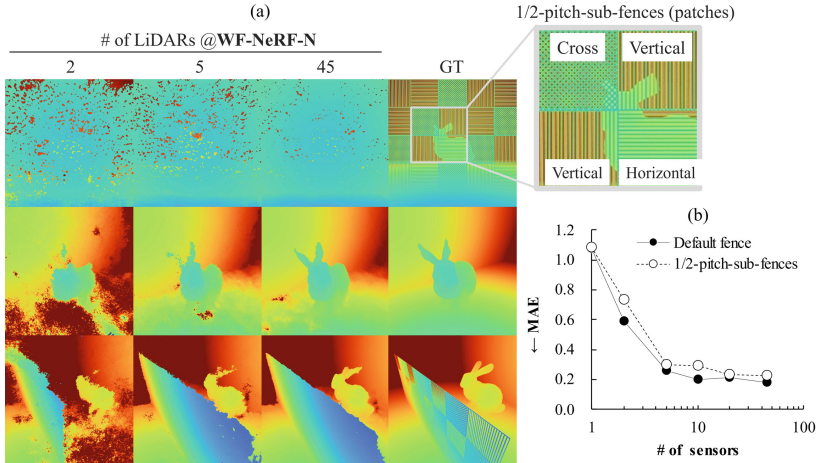
**Evaluation w.r.t. Varying the Number of Sensors.** In this experiment, we varied the number of sensors from 1 to 45 to compare the performance of the proposed model with the conventional NeRFs. The number of sensors refers to the number of cameras for the RGB-based methods (vanilla-NeRF, mip-NeRF, TensoRF, Nerfacto) and the total number of cameras and LiDARs for the other methods (DS-NeRF, D-Nerfacto, WF-NeRF-T and WF-NeRF-N.)

Figure 4 shows the transition of each metric with respect to the change in the number of sensors. The proposed method is more accurate than the conventional NeRFs even when the number of sensors is small. Unlike the widely used public datasets, the experimental setting assumed in this paper involves many occlusions. According to these results, we can confirm the effectiveness of the proposed method, which assumes the presence of multiple depth values in raw signals of LiDARs. Also, a detailed result is summarized in Tab. 2. WF-NeRF-T maintains higher depth accuracy than conventional NeRFs.

Figure 5 shows the depth map visualization with the number of sensors in Tab. 2. The two proposed methods successfully estimate multiple depths in the



scene. When the viewpoint angle changes (rows 2 and 3), we see that the fence, object, wall, and floor are reconstructed. On the other hand, Vanilla-NeRF, mip-NeRF, and D-NeRF appear to reconstruct the fence from the front view (row 1) but actually fail to estimate the multiple depths. DS-NeRF reconstructs the fence like a wall.



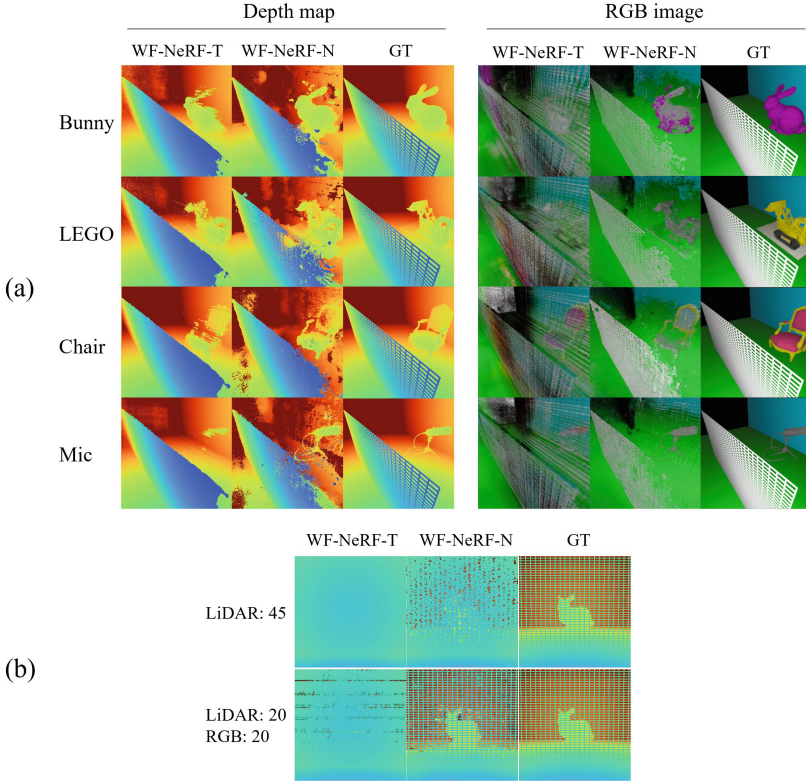
**Fig. 6.** (a) Results on various fence types. (b) Comparison of depth MAE w.r.t the number of sensors.

**Evaluation w.r.t. Varying the Fence Types.** Figure 6 shows the reconstruction results on various fence types consisting of half-pitch-sub-fences with different line directions. Although the reconstruction quality of the fence slightly decreases compared to Fig. 3 and Fig. 5, that of the scene behind the fence (the bunny and background) maintains the same level.

**Sensor Fusion with RGB Cameras.** As in DS-NeRF, the proposed method can perform sensor fusion by sharing a neural model for RGB cameras. In other words, LiDARs and RGB cameras capture a scene at the same time to recover high-resolution color images in addition to multi-depths. Each device can be installed in spatially different locations. In order to train the fused model for the different signals, the loss for each device is linearly combined using the weights  $\lambda$ :

$$L_{\text{fusion}} = L_{\text{camera}} + \lambda L_{\text{LiDAR}}. \quad (4)$$

Here,  $L_{\text{camera}}$  represents the photometric loss for RGB images used in the conventional NeRFs. We varied  $\lambda$  with decay for WF-NeRF-N to avoid overfitting, ranging from  $\lambda_{\text{max}} = 5.0 \times 10^{-12}$  to  $\lambda_{\text{min}} = 10^{-14}$  with multiplying a decay



**Fig. 7.** (a) Depth and RGB images rendered by the sensor fusion of 20 LiDARs and 20 cameras set in front of the fence. (b) Effect of the sensor fusion. LiDARs with RGB cameras provides a more detailed reconstruction.

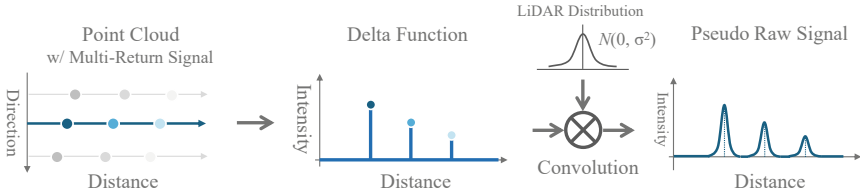
parameter 0.9999975 at each iteration. For other models, WF-NeRF-V and WF-NeRF-M,  $\lambda$  was set to ranging from  $10^{-8}$  to  $10^{-10}$  with decay parameter 0.99995.

Figure 7(a) shows qualitative results of the bunny, LEGO, Chair, and Mic generated by WF-NeRF-T and WF-NeRF-N. By mixing high resolution RGB image, detailed and colored 3D scene can be obtained with fewer sensors. Figure 7(b) shows a shape-reconstruction results of the fence and the bunny behind the fence. WF-NeRF-N provides more accurate and clearer shape than WF-NeRF-T.

### 4.3 Real Dataset

In this section, we report the experimental results in a real environment. We used a commercial LiDAR, Livox Mid-40 equipped with a multi-return mode, which accepts multiple reflection signals per beam up to three and outputs each signal as different point clouds. As shown in Fig. 8, To recover 1D raw signals, we merged the multi-return point clouds into a single point cloud and fit a

Gaussian distribution on the points of each ray. The standard deviation  $\sigma$  was set by 2cm based on the specification of Livox Mid-40. We manually measured the beam shape of Livox Mid-40 by  $3\text{cm} \times 1\text{cm}$  vertically long (cylindrical) at 3 m distance, then, adjusted the ray bundle setting to the beam shape. The depth resolution of the raw signals was reduced from 4800 to 128 by the 1-pass IFM sampling. The depth maps for conventional method (D-Nerfacto) were created by simply projecting point cloud data from the sensor viewpoint.



**Fig. 8.** Pseudo raw signals calculation from multi-return signals. Point cloud data per direction is converted to the delta function along distance, then convoluted with Gaussian distribution with the LiDAR depth accuracy  $\sigma$ .

We used the LiDAR (Livox Mid-40) to capture a scene consisting of multiple depths where an object is behind a mesh-like net (Fig. 9) and plants (Fig. 10). The mesh in Fig. 9 is of size 2.5 cm square. Figures 9 and 10 show an RGB image (not used for training), a point cloud, reconstruction results by D-Nerfacto (using 2-layered depth map) outputs, and reconstruction results by our methods. Our methods (WF-NeRF-N) reconstructs the net/plants and the scene behind them as continuous surface while point cloud output from LiDAR can only be represented as sparse point data. D-Nerfacto appear to reconstruct some of the scene behind net/plants, but much has disappeared and the details is lost. These results validate that the proposed methods can provide a reasonable estimation in real environments composed of fine objects and multiple depths.

## 5 Discussion

As reported in the experiments, WF-NeRF-T and WF-NeRF-N show unique trends numerically and visually. The two methods represent 3D geometric information differently, resulting in image quality differences. Since TensorRF performs tensor decomposition of the 3D coordinates and feature values, WF-NeRF-T is effective for reproducing flat structures like walls and floors due to the quantization effect. On the other hand, Nerfacto stores spatial features as a product of hash information. WF-NeRF-N provides more detailed shapes of the fence than WF-NeRF-T but loses surface continuity in some parts depending on the signal density of training data.

One of the current limitations in handling LiDAR raw signals with NeRFs is the memory inefficiency of increasing the number of sensors due to the large

amount of data. Although the 1-pass IFM sampling reduces the data size by an order of magnitude, the data size is still much larger than RGB images. A quick workaround is to use a sparse representation such as Octree or apply FFT decomposition, but those methods may lose high-frequency components. Another issue is that there are few commercial LiDARs that output raw signals. For example, SPAD sensors have a narrow range and are unsuitable for outdoor scenes. Full waveform LiDARs are too large for portable measurement. We fitted a Gaussian distribution to the peak values in the real data experiment but will need a more sophisticated and efficient approach to estimate the 1D signal distribution accurately. Improving sampling performance can make the beam scan sparse with increasing the beam spot size. Moreover, the scan time, which is a common issue of many LiDARs, can be shortened.

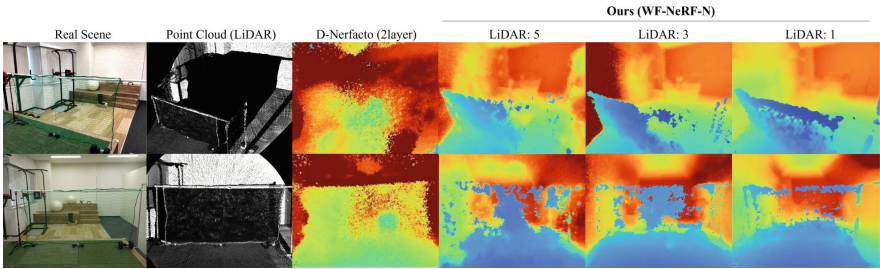


Fig. 9. Qualitative results on our real dataset *Net*.

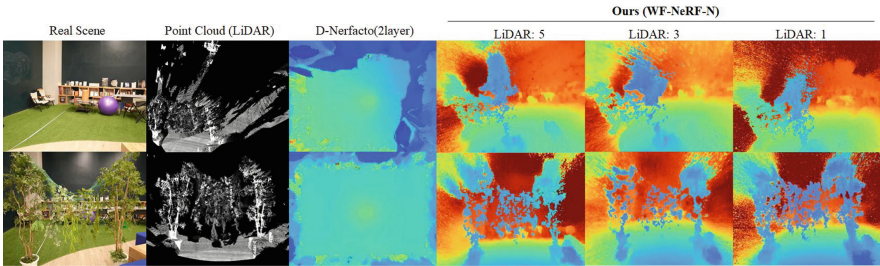


Fig. 10. Qualitative results on our real dataset *Plants*.

## 6 Conclusion

This paper presented a NeRF framework, WF-NeRF, to reconstruct multiple depth values of LiDAR raw signals. To obtain accurate estimation, we introduced two novel techniques: the MSRE (mean squared relative error) loss and

point sampling along the wavefront of LiDAR beams. We incorporated these techniques into Vanilla-NeRF (WF-NeRF-V), mip-NeRF (WF-NeRF-M), TensorRF (WF-NeRF-T) and Nerfacto (WF-NeRF-N) in the experiments. We set up an experimental environment consisting of a mesh-like fence and an object behind the fence so that LiDAR beams have multiple depth values. First, we reported by an ablation study that both MSRE and wavefront sampling are required to reconstruct multiple depth values. The quantitative results showed that using many sensors does not always lead to improvements for the conventional NeRFs in such conditions. Then, we demonstrated in synthetic and real data experiments that WF-NeRF-T and WF-NeRF-N successfully reconstruct an object occluded by the fence even with fewer sensors.

## References

1. Avidan, S., Shashua, A.: Novel view synthesis in tensor space. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1034–1040. IEEE (1997)
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: a multiscale representation for anti-aliasing neural radiance fields. In: ICCV, pp. 5855–5864 (2021)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: anti-aliased grid-based neural radiance fields. In: ICCV (2023)
4. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: TensorRF: tensorial radiance fields. In: ECCV, pp. 333–350. Springer, Heidelberg (2022)
5. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12882–12891 (2022)
6. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(8), 1362–1376 (2009)
7. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004). ISBN: 0521540518
8. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: semantically consistent few-shot view synthesis. In: ICCV, pp. 5885–5894 (2021)
9. Jarabo, A., Marco, J., Munoz, A., Buisan, R., Jarosz, W., Gutierrez, D.: A framework for transient rendering. *ACM Trans. Graph.* **33**(6) (2014). <https://doi.org/10/gfznb8>
10. Laurentini, A.: The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(2), 150–162 (1994). <https://doi.org/10.1109/34.273735>
11. Malik, A., Mirdehghan, P., Nousias, S., Kutulakos, K.N., Lindell, D.B.: Transient neural radiance fields for lidar view synthesis and 3d reconstruction (2023)
12. 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)
13. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* **41**(4), 102:1–102:15 (2022). <https://doi.org/10.1145/3528223.3530127>
14. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: dense tracking and mapping in real-time. In: ICCV, pp. 2320–2327. IEEE (2011)

15. Rematas, K., et al.: Urban radiance fields. In: CVPR (2022)
16. Tancik, M., et al.: Nerfstudio: a modular framework for neural radiance field development. In: Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings. ACM (2023). <https://doi.org/10.1145/3588432.3591516>
17. Xu, Q., et al.: Point-nerf: point-based neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5438–5448 (2022)
18. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: neural radiance fields from one or few images. In: CVPR, pp. 4578–4587 (2021)



# Visibility-Aware Pixelwise View Selection for Multi-View Stereo Matching

Zhentao Huang<sup>✉</sup>, Yukun Shi<sup>✉</sup>, and Minglun Gong<sup>✉</sup>

University of Guelph, Guelph, ON N1G 2W1, Canada  
{zhentao,yshi21,minglun}@uoguelph.ca

**Abstract.** The performance of PatchMatch-based multi-view stereo algorithms is greatly influenced by the chosen source views used for matching cost computation. Existing methods usually detect occlusions in a rather ad-hoc way, which can negatively impact the computation. In contrast, our paper introduces an innovative approach that deliberately models view visibility. We present a novel visibility-guided pixelwise view selection scheme that progressively refines the set of source views for each pixel in the reference view using visibility information from validated solutions. Furthermore, the Artificial Multi-Bee Colony (AMBC) algorithm is leveraged to parallelly search optimal solutions for different pixels. To ensure smoothness of neighboring pixels and better manage textureless areas, rewards are assigned to solutions that come from validated sources. Our method, validated through experiments on two datasets, improves detail recovery in occluded and low-textured regions, demonstrating noteworthy performance on demanding scenes. Our implementation is available at <https://github.com/Ricky-S/Visibility-Aware-Pixelwise-View-Selection>.

**Keywords:** Multi-view stereo · PatchMatch · Artificial multi-bee colony · Visibility-aware

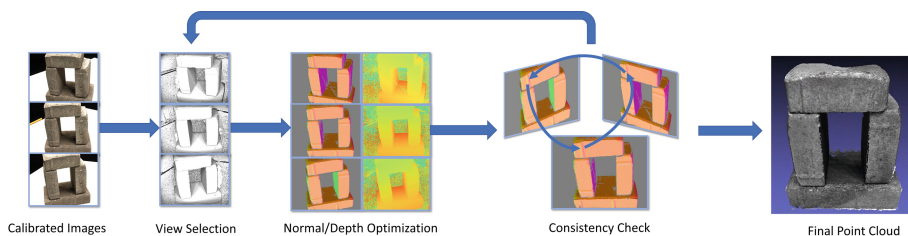
## 1 Introduction

Multi-view stereo (MVS), which estimates dense 3D point clouds from a set of input images, is an important research topic and supports many downstream applications, such as autonomous driving, 3D reconstruction, and virtual reality. Despite significant progress made in recent years [4, 12, 37, 44], reconstructing accurate and complete 3D point cloud models remains challenging. Obstacles such as occlusions, low/repetitive textures, and view-dependent appearances often hinder the process.

Inspired by the success of MVSNet [55], numerous learning-based methods [15, 27, 35, 45, 56, 59] have been proposed in recent years and shown outstanding

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78456-9\\_9](https://doi.org/10.1007/978-3-031-78456-9_9).



**Fig. 1.** The proposed algorithm iteratively performs view selection, normal/depth optimization, and geometry consistency check. Once the process converges, 3D point clouds from different views are fused together to produce the final model.

performances. While they are top-ranked for various MVS datasets [1, 21, 57], they have two major limitations, namely, network complexity and dataset diversity [46]. The former often results in high computational time and memory requirements, while the latter poses difficulties in obtaining diverse and labeled training data. Although self-supervised MVS approaches have been proposed [10, 16, 20, 30], their performance remains unsatisfactory. Therefore, we believe that non-learning-based MVS approaches, which leverage available geometric constraints to their fullest extent, offer significant value.

Recently, PatchMatch-based methods [13, 37, 51, 52] show excellent capability in depth map estimation. Following [3], these methods generally have a four-step pipeline: random initialization, propagation, view selection, and refinement. View selection is an essential factor here because correct matches can only be found from nearby unoccluded views, and occlusions are common under the MVS setting. Yet, existing approaches often resort to ad-hoc view selection methods (*e.g.* top- $n$  views with the lowest matching cost [13]) without considering visibility constraints. Therefore, two motivating questions are whether we can make the view selection process visibility-aware and how much benefit we can gain from such enhancement.

To this end, we develop a pixelwise view selection approach, which progressively updating the source views used for each pixel. The selected views will be used for both matching cost calculation and depth/normal consistency check, which leading to a set of validated solutions. These validated solutions guide future view selections through visibility checks; see Fig. 1.

Even with a proper set of source views selected, searching the optimal depth and normal for each pixel is still a challenging problem due to the large solution space and numerous false local optima. To address this issue, we employ three strategies: 1) utilizing a swarm-based optimization framework, the Artificial Multi-Bee Colony algorithm [48], to store multiple solutions for one pixel and avoid being trapped in local optima; 2) using both intra-image and inter-image solution propagation to speed up convergence; and 3) incorporating a smoothness term in intra-image propagation to handle low/repetitive texture areas more effectively. Under the same set of parameters, the final algorithm outperforms existing non-learning-based methods on the DTU dataset and also achieves note-



worthy performance on the more challenging subset of the Tanks-and-Temples dataset.

## 2 Related Work

### 2.1 Learning-Based MVS

The huge success of the Convolutional Neural Network (CNN) has sparked interest in introducing learning to 3D shape reconstruction. Two natural paths are to estimate a 2.5D depth map for each input image or to operate on 3D voxels. Depth map based approaches are proposed to infer per-pixel depth either from a single image [25] or to use visual cues extracted from stereo pairs [14, 58]. Additional processing is then needed to consolidate multiple depth maps into a single 3D point cloud [31, 49]. Voxel-based methods [9, 42] utilize 3D convolution operators to encode and decode geometric features in discretized 3D space directly but are limited to the relatively low voxel resolution. To address these limitations, 3D point cloud based [43, 53] and implicit surface based [7, 33] approaches are also proposed.

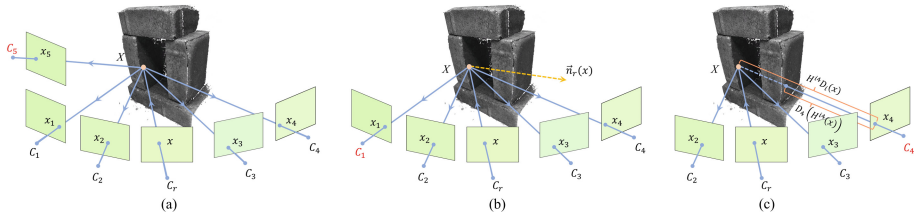
SurfaceNet [17] is one of the first learning-based MVS, which combines images and camera parameters as input and outputs the voxel surface reconstruction. It uses a 3D CNN to regularize and infer the surface voxels. MVSNet [55] improves 3D reconstruction by first extracting 2D features from a reference image and source images. However, forming 3D cost volumes is memory-consuming, [56] replaces 3D CNNs with sequential 2D CNNs to reduce memory but cost more runtime. In recent years, [8, 15, 54] integrate the coarse-to-fine strategy into MVS reconstruction. This architecture can help reduce memory consumption and achieve higher-resolution outputs. Note that the variance-based cost metric in these methods is under the assumption that a pixel is visible in all input images, Vis-MVSNet [59] integrates the occlusion information into the MVS network via the matching uncertainty estimation to suppress the influence of occluded pixels. Instead of cost volume approaches, [6] directly processes the target scene as point clouds and refines point clouds iteratively.

### 2.2 Non-learning-Based MVS

Non-learning-based MVS approaches infer depth information from matching costs of rectified image patches. According to [38], Non-learning-based MVS can be categorized into four types to represent the scene: volumetric based [22, 39], point cloud based [12, 23], mesh based [40, 41], and depth map based [13, 51]. Recently, to harness parallel capabilities and achieve optimal performance, PatchMatch series [13, 51] have been widely used in this field. The core idea of Patchmatch [2] is to establish matches between patches randomly and iteratively by performing an efficient nearest-neighbor search. Adapting this idea to the stereo matching problem, [3] proposes the concept of using planes as support windows assigned to every pixel. [51] applies the pyramid structure and

geometric consistency. To address textureless regions, [52] uses a coarse-fitting plane hypothesis.

Motivated by the concept of PatchMatch Stereo and recognizing the importance of different views in MVS, we propose a method that takes advantage of the parallel processing capabilities of optimization algorithms and the physical constraints of images.



**Fig. 2.** Illustration for the progressive view selection process: (a) initially, only source views with poor triangulation angle ( $C_5$ ) will be removed from the source view set; (b) once the normal of  $X$  is estimated, views with poor incident angles ( $C_1$ ) will be removed; and (c) validated solutions are used to further remove occluded views ( $C_4$ ) from the set.

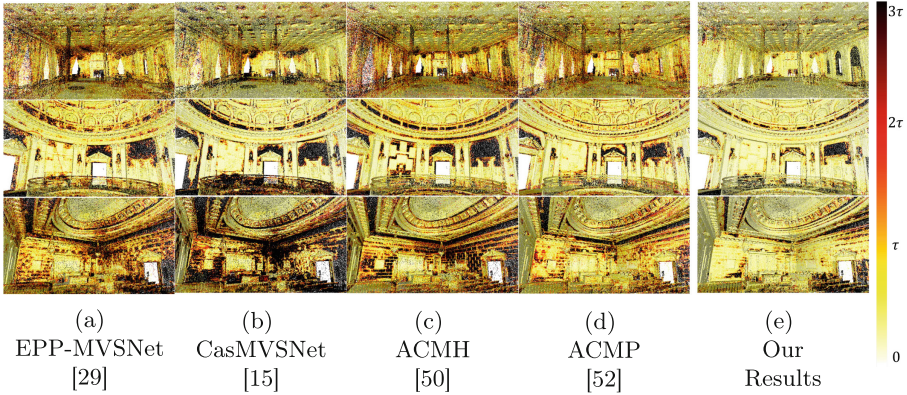
### 2.3 Artificial Multi-Bee-Colony Alg.

Artificial Bee Colony (ABC) algorithm [18] and its variants [61] are optimization frameworks that simulate the foraging behavior of honey bees. Compared with other population-based algorithms, the ABC algorithm can achieve equal or better performance with fewer parameters.

Our work leverages the idea of AMBC to build the MVS framework. We not only apply the between-colony communication idea for propagating solutions between different pixels of the same image but also for propagation among different images. In addition, rewards are added to validated solution propagation, which allows simple yet effective enforcement of smoothness constraints.

## 3 Visibility-Aware MVS

Given a set of input 2D images  $I = \{I_i | i = 1 \dots N\}$  with known camera parameters  $C = \{C_i | i = 1 \dots N\}$ , the goal of MVS is to estimate pixel-wise depth maps  $d = \{d_i | i = 1 \dots N\}$  for every view and fuse them into a 3D point cloud. Specifically, when processing a reference image  $I_{ref}$ , MVS algorithms normally estimate a local fitting plane  $P$  for each pixel  $x$  in  $I_{ref}$ 's local coordinates, using some of the remaining views as source images  $I_{src} \in \{I\} - I_{ref}$ . The plane  $P$  depicts both the depth and normal information of the local geometry, which are denoted as  $d_{ref}(x)$  and  $\mathbf{n}_{ref}(x)$ , respectively.



**Fig. 3.** Visual comparison of the quality of point clouds obtained by different algorithms on three Advanced scenes in the Tanks-and-Temples dataset. The visualization is provided by the official website, with pixel color representing the distance from the ground truth to the nearest reconstructed point. The legend is on the right. Our results show the lowest errors.

Figure 1 illustrates an overview of our method. We construct a three-phased process that evolves over cycles. An initial set of source views is selected for each pixel in each reference image based on only camera parameters  $C$ . These source views are used to compute matching costs, based on which an AMBC algorithm is used to search for the optimal solution (depth and normal) for each pixel. A geometry consistency check is then performed by projecting the optimal solution of a given pixel to all its source views. The solution is considered validated if both depth and normal are consistent with the corresponding pixels in these source views. Validated solutions are further used for: 1) guiding the future source view selection process as some source views may be determined as occluded; 2) communicating pixels information both within the same image and across different images; and 3) fusing into the final point cloud.

### 3.1 Pixelwise View Selection

The selection of source views for matching cost calculation strongly impacts the quality of reconstruction results. Previous work has proposed to use triangulation angle, incident angle, and image resolution-based geometric priors to perform pixelwise view selection [37]. While we acknowledge the importance of resolution-based geometric prior in handling large-scale scenes, its benefit for reconstructing small scenes is limited since all objects are captured under similar resolutions. Hence, we removed this term from our implementation for simplicity. Instead, we add a visibility-based term, which handles occlusions based on geometric information instead of heuristics.

The actual terms used for view selection are based on available information. At the beginning of the process, we have yet to gain prior knowledge of scene

geometry. Therefore, when processing image  $I_i$  as the reference view, only the triangulation angle term is used for view selection. All nearby views whose triangulation angle with  $I_i$  is between  $[10^\circ, 30^\circ]$  are selected into the source view set  $\{I_{src}\}$ ; see Fig. 2 (a). It is worth noting that the same set of views is used for all pixels in  $I_i$ .

Once the depth  $d_i(x)$  and normal  $\mathbf{n}_i(x)$  for each pixel  $x$  in  $I_i$  are estimated, the incident angle term will be used. If a given view  $I_j$  has a poor incident angle, *i.e.* the angle between  $\mathbf{n}_i(x)$  and the viewing vector of  $I_j$  is greater than  $80^\circ$ ,  $I_j$  will be removed from the source view set  $\{I_{src}\}$ ; see Fig. 2 (b). As a result, the set  $\{I_{src}\}$  will be adaptively determined for different pixels in  $I_i$ .

Finally, once validated depth and normal are found for different views (details on solution validation will be discussed in Sect. 3.3), the visibility term will be introduced. That is, for a given pixel  $x$  in reference view  $I_i$ , we will first backproject  $x$  to a 3D scene point  $X$  using the estimated depth  $d_i(x)$ . The 3D point  $X$  is then projected to each view  $I_j$  in set  $\{I_{src}\}$ . Without losing generality, here we assume the projection of  $X$  on image  $I_j$  is pixel  $y$ . We consider the  $X$  is occluded in  $I_j$  if and only if a validated solution is found at pixel  $y$  and the depth  $d_j(y)$  is smaller than the distance to 3D point  $X$ .  $I_j$  will be removed from source view set  $\{I_{src}\}$  if  $X$  is occluded in  $I_j$ ; see Fig. 2 (c).

### 3.2 Search with AMBC Algorithm

To apply AMBC to MVS, we represent each pixel  $x$  in the reference image  $i$  as a bee colony, where a candidate solution  $\langle d_i(x), \mathbf{n}_i(x) \rangle$  is represented as a food source. Each pixel stores a fixed amount of candidate solutions, which is a preset parameter called the food number  $F$ . The optimal results are obtained by dispatching three types of bees: employed, onlooker, and scout bees. We parameterize the solution space in the Euclidean scene space, as Gipuma [13] does, which avoids the need for epipolar rectification. This approach also enables the generation of dense normals, which can be used for point cloud fusion [19].

*Random Initialization.* We randomly generate  $F$  candidate solutions for each pixel, where  $F$  is the food number. We follow [32] to randomly sample the normal vector over the visible hemisphere uniformly. A trial count  $T(\cdot)$  is set to zero initially and maintained for each candidate solution. It is designed to track whether each candidate solution is updated through iterations.

*Matching Cost Evaluation.* The similarity between two patches related via plane-induced homography defines whether the candidate solution depicts the scene correctly. In binocular stereo, the matching cost is straightforward. When extending to multi-view stereo, we adopt the following equation to aggregate the matching cost of the solution  $\langle d_i(x), \mathbf{n}_i(x) \rangle$  in pixel  $x$  of the reference image  $i$ :

$$C(d_i(x), \mathbf{n}_i(x)) = \begin{cases} \frac{\sum_{j \in \{S_{src}\}} m(i, j)}{|\{S_{src}\}| - 1}, & \text{if } |\{S_{src}\}| > 1 \\ +\infty, & \text{otherwise} \end{cases} \quad (1)$$

where  $m(i, j)$  represents the matching cost between two patches from reference view  $i$  and source view  $j$ , and  $\{S_{src}\}$  is the set of suitable source views for pixel  $x$ . We adopt bilaterally weighted Normalized Cross-Correlation [37] as our matching cost function. The aggregation cost is divided by  $|\{S_{src}\}| - 1$  rather than  $|\{S_{src}\}|$  because we expect the results to be an unbiased sample estimate that prefers a larger  $\{S_{src}\}$  set, which is more likely to capture the true normal and depth than a smaller set.

*Employed Bees.* The employed bees search within the local colonies by randomly perturbing each candidate solution. For solution  $y_x$ , the perturbed one  $y'_x$  is generated by:

$$y'_x = y_x + R(-1, 1)(y_n - y_x) \quad (2)$$

where  $R(-1, 1)$  returns a value uniformly distributed between  $[-1, 1]$ , and  $y_n$  is another candidate solution randomly selected within the colony. The matching cost of the perturbed solution is then evaluated based on Eq. 1. If  $C(y'_x) < C(y_x)$ , we replace  $y_x$  with  $y'_x$  and set  $T(y_x) = 0$ . Otherwise, we set  $T(y_x) = T(y_x) + 1$ .

*Onlooker Bees.* The onlooker bees search in neighboring colonies. Following [13], we adopt a red-black checkerboard pattern for sampling. It divides the image into red and black groups. Pixels in the same color group can be processed in parallel without interfering with others.

For every candidate solution  $y_x$  with a black label, we randomly select a colony with a red label following the pattern in [51], and vice versa. The solution  $y_n$  with the lowest matching cost at the selected colony is then compared against  $y_x$ . If  $C(y_n) < C(y_x)$ , we replace  $y_x$  with  $y_n$ . Otherwise, we set  $T(y_x) = T(y_x) + 1$ .

*Scout Bees.* Both employed and onlooker bees search within the solution space that is spanned by the existing candidates. Hence, through iterations, they lead the colony's candidate solutions to converge to a smaller and smaller range. The scout bees are therefore introduced to perform global searching and avoid potential local optimum. We only perform global searching for  $F - 1$  candidate solutions since we want to always keep the best candidate in the colony. For each remaining solution  $y_x$ , if  $T(y_x)$  exceeds a preset threshold (empirically set to 10), scout bee replaces it with a randomly generated solution. The new solution is evaluated via Eq. 1, and we reset  $T(y_x) = 0$ .

### 3.3 Geometric Consistency Check

Due to noise and/or deviation from Lambertian property, mismatches sometimes have lower matching costs than correct matches. To filter out these mismatches, the consistency check is applied. As shown in Fig. 1, our approach alternates between the depth/normal estimation stage and the consistency check stage until the whole process converges. That is, once normal and depth map calculation is completed for all views, the algorithm will cross-check the obtained depth/normal among these views. The solutions that pass the check will be marked as

validated for the next calculation cycle. It is worth noting that both validated and unvalidated solutions are continued to be refined in further optimizations. However, only validated solutions are used for:

- Enforcing smoothness constraint during intra-image solution propagation;
- Propagating solutions among different views;
- Providing visibility information during future source view selection.

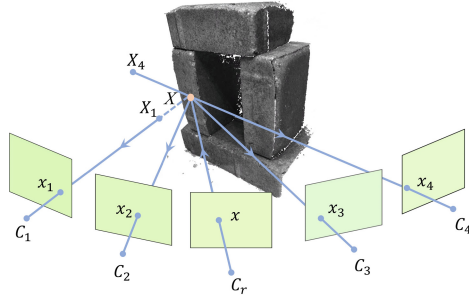
Ideally, if a solution  $\langle d_i(x), \mathbf{n}_i(x) \rangle$  for the pixel  $x$  in image  $I_i$  is correct, it should be consistent with all corresponding solutions in the source views used for pixel  $x$ . That is, after applying homography transform  $H^{ij}$  between view  $I_i$  and one of the source views  $I_j$ , we should have  $H^{ij}(d_i(x)) = d_j(H^{ij}(x))$  and  $H^{ij}(\mathbf{n}_i(x)) = \mathbf{n}_j(H^{ij}(x))$ . In practice, we relax the consistency requirement for  $\langle d_i(x), \mathbf{n}_i(x) \rangle$  to:

$$\begin{aligned} |H^{ij}(d_i(x)) - d_j(H^{ij}(x))| &< T_{depth}, \\ \arccos(H^{ij}(\mathbf{n}_i(x)) \cdot \mathbf{n}_j(H^{ij}(x))) &< T_{normal} \end{aligned} \quad (3)$$

where  $T_{depth}$  and  $T_{normal}$  are two preset thresholds. Note that depth comparison is conducted in the disparity space in the experiment to automatically adapt to sampling variation. In addition, instead of requiring all source views used for pixel  $x$  to be consistent with  $\langle d_i(x), \mathbf{n}_i(x) \rangle$ , we relax the constraint by allowing a small percentage of these views not satisfying the above conditions. That is,  $\langle d_i(x), \mathbf{n}_i(x) \rangle$  is labeled as validated if it is consistent with 70% or more of the source views  $\{I_{src}\}$  selected for pixel  $x$ .

*Propagation Between Views.* Initially, solution propagation only appears in neighboring colonies within the same image (intra-image propagation) via onlooker bees. It is one of the vital concepts in PatchMatch-based multi-view stereo. In this paper, we propose inter-image propagation as well, which enables the solution to propagate between pixels in different images related by consistency check. Figure 4 shows an example of propagation between views. The key concept is to propagate already validated solutions to views without validated solutions at the corresponding locations. This helps to speed up the convergence and prevent potential local optimum.

*Smoothness Constraint.* Not being able to handle textureless areas properly is a significant limitation for the PatchMatch-based methods [36]. In binocular stereo, this problem is often addressed by introducing an additional smoothness term, which converts per-pixel optimization into global optimization. Solving global optimization under the MVS setting can be highly computationally expensive. Hence, we utilize a simple yet effective approach, which applies rewards to solutions propagated by onlooker bees. The key observation is that the matching cost for the correct match and mismatches are similar in textureless areas. Adding a small reward to the validated solutions propagated by onlooker bees effectively encourages the same fitting plane being selected at the current solution. The smoothness is therefore enforced, and flat textureless surfaces can be



**Fig. 4.** Illustration of inter-view propagation. The solution in reference view  $C_r$  is validated through two of its neighboring views ( $C_2, C_3$ ). It is then projected to views  $C_1, C_4$  as candidate solutions at pixels  $x_1, x_4$ .

properly modeled. For areas with distinct textures, the small reward will not affect the search for optimal solutions.

### 3.4 Fusion

Follows [13, 37], after obtaining all the depth and normal maps, we fuse them into a single point cloud. More specifically, for  $N$  images in the scene, we consequently select each image as the reference image and convert its depth map to 3D points in the world coordinate, then project them to the rest  $N-1$  views. If the disparity difference is less than 0.5 pixels, and the angle between normals is smaller than  $30^\circ$ , they are considered as projections of the same 3D point. The depth and normal are then averaged into a single 3D point in the result point cloud.

**Table 1.** Quantitative results for non-learning-based approaches on **full** DTU dataset. Lower is better. Our method ranks first in terms of Completeness and Overall metrics.

Method	Acc.(mm)	Comp.(mm)	Overall
Furukawa [12]	0.605	0.842	0.724
Tola [44]	0.307	1.097	0.702
COLMAP [37]	0.400	0.532	0.664
Campbell [4]	0.753	0.540	0.647
Gipuma [13]	<b>0.273</b>	0.687	0.480
Ours	0.385	<b>0.388</b>	<b>0.386</b>

## 4 Experiments

We evaluate our method on two widely-used datasets, DTU [1] and Tanks-and-Temples [21], under the same set of parameters. Here we present both evaluation

results and ablation studies. The ablation study, which examines the critical components of pixel-wise view selection, smoothness constraints, and the pixel-wise view selection, is detailed in the supplementary materials. Additionally, these materials include an extensive visual comparison with other methods, offering further insight into the performance and effectiveness of our approach.

#### 4.1 Datasets and Settings

The DTU Robot Image dataset [1] contains 124 different scenes captured by a structured light scanner mounted on an industrial robot arm. Each scene has been taken from 49 or 64 positions. The image resolution is  $1600 \times 1200$ , and the camera calibration parameters are provided. Most importantly, this dataset captures objects at a close distance and hence visibility handling is a major concern, for which our approach aims to address.

The Tanks-and-Temples dataset [21] contains 21 scenes with image resolution of  $1920 \times 1080$ . Unlike DTU, it does not provide ground truth camera poses, so we utilized COLMAP [37] to estimate them. The dataset is divided into training and test sets, with the latter further split into Intermediate and Advanced subsets. The Advanced subset contains larger scenes with more complex view-point changes than the intermediate one. Since the Advanced subset has more occlusion issues, it is the focus of our study.

**Table 2.** Quantitative results on DTU **evaluation set**. Both learning-based and non-learning-based approaches are listed for impartial comparison.

Method	Acc.(mm)	Comp.(mm)	Overall
Non-Learning-based			
Furukawa [12]	0.613	0.941	0.777
Tola [44]	0.342	1.190	0.766
Campbell [4]	0.835	0.554	0.695
Gipuma [13]	<b>0.283</b>	0.873	0.578
COLMAP [37]	0.411	0.657	0.534
Ours	0.405	<b>0.381</b>	<b>0.393</b>
Learning-based			
SurfaceNet [17]	0.450	1.040	0.745
MVSNet [55]	0.396	0.527	0.462
P-MVSNet [27]	0.406	0.434	0.420
R-MVSNet [56]	0.383	0.452	0.417
CasMVSNet [15]	<b>0.325</b>	0.385	0.355
PatchMatchNet [45]	0.427	<b>0.277</b>	0.352
UniMVSNet [35]	0.352	0.278	<b>0.315</b>



## 4.2 Point Cloud Evaluation

For the DTU datasets, we present two versions of quantitative results. The first version compares non-learning-based methods on the full DTU dataset, whereas the second follows [55] and evaluates both learning- and non-learning-based methods on the validation set (22 scenes) only. Table 1 shows that our approach performs the best in both completeness and overall metrics among non-learning-based approaches. Table 2 further demonstrates that our performance is comparable to learning-based methods.

For the Tanks-and-Temples dataset, the generated point clouds undergo evaluation through the dataset’s official website, employing F-scores for assessment, as detailed in Table 3. Our methodology distinguishes itself within the non-learning-based segment, managing to secure performance on par with learning-based approaches, particularly within the Advanced subset. This achievement is largely attributed to the method’s handling of occlusions. The Intermediate subset is presented in the supplementary material.

**Table 3.** Quantitative results of F-score (the higher the better) on Tanks-and-Temples [21] Advanced subset, divided into learning-based and non-learning-based methods. The best results within each category are highlighted in bold.

Method		Advanced						
		mean	Aud.	Bal.	Cou.	Mus.	Pal.	Tem.
Non-L.-based	COLMAP [37]	27.24	16.02	25.23	34.70	41.51	18.05	27.94
	PLC [24]	34.44	23.02	30.95	42.50	49.61	26.09	34.46
	ACMH [50, 51]	33.73	21.69	32.56	40.62	47.27	24.04	36.17
	ACMM [51]	34.02	23.41	32.91	41.17	48.13	23.87	34.60
	ACMP [52]	37.44	<b>30.12</b>	34.68	<b>44.58</b>	50.64	27.20	37.43
	ACMMP [50]	37.84	30.05	35.36	44.51	50.95	27.43	<b>38.73</b>
	<b>Ours</b>	<b>38.26</b>	24.97	<b>44.25</b>	41.57	<b>53.11</b>	<b>28.52</b>	37.11
Learning-based	PatchMatchNet [45]	32.31	23.69	37.73	30.04	41.80	28.31	32.29
	CasMVSNet [15]	31.12	19.81	38.46	29.10	43.87	27.36	28.11
	AttMVS [28]	31.93	15.96	27.71	37.99	52.01	29.07	28.84
	GBi-Net [34]	37.32	29.77	42.12	36.30	47.69	31.11	36.93
	EPP-MVSNet [29]	35.72	21.28	39.74	35.34	49.21	30.00	38.75
	TransMVSNet [11]	37.00	24.84	44.59	34.77	46.49	34.69	36.62
	MVSFormer [5]	40.87	28.22	<b>46.75</b>	39.30	52.88	35.16	42.95
	GeoMVSNet [60]	<b>41.52</b>	30.23	46.53	<b>39.98</b>	<b>53.05</b>	<b>35.98</b>	<b>43.34</b>
	ET-MVSNet [26]	40.41	28.86	45.18	38.66	51.10	35.39	43.23
	APD-MVS [47]	39.91	<b>32.54</b>	42.79	39.24	51.03	33.08	40.77

## 5 Conclusion

In this study, we introduce a visibility-aware, pixelwise view selection technique tailored for PatchMatch-based multi-view stereo. This approach progressively refines view selection for each pixel as further insights into scene geometry are acquired. The chosen views are utilized for both matching cost assessment and consistency verification.

While utilizing the Artificial Multi-Bee Colony (AMBC) to find optimal solutions for distinct pixels concurrently, we employ between-colony onlooker bees for both intra-image and inter-image evolution propagation. To address the dearth of photometric clues in regions with low texture, we have integrated rewards that motivate the propagation of validated solutions to nearby pixels, thereby effectively applying the smoothness constraint. Upon testing our approach on both the DTU and Tanks-and-Temples datasets, it has been proven that our method outperforms existing non-learning-based techniques, particularly in intricate scenes characterized by complex occlusions. The ablation study affirms that our two primary components - visibility-aware pixelwise view selection and smoothness rewards - considerably enhance the 3D reconstruction of occluded and low-textured regions.

## References

1. Aanaes, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B.: Large-scale data for multiple-view stereopsis. *Int. J. Comput. Vis.* 1–16 (2016)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **28**(3), 24 (2009)
3. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: *The British Machine Vision Conference (BMVC)*, vol. 11, pp. 1–11 (2011)
4. Campbell, N.D., Vogiatzis, G., Hernández, C., Cipolla, R.: Using multiple hypotheses to improve depth-maps for multi-view stereo. In: *European Conference on Computer Vision*, pp. 766–779. Springer, Heidelberg (2008)
5. Cao, C., Ren, X., Fu, Y.: Mvsformer: multi-view stereo by learning robust image features and temperature-based depth. *Trans. Mach. Learn. Res.* (2023)
6. Chen, R., Han, S., Xu, J., Su, H.: Visibility-aware point-based multi-view stereo network. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(10), 3695–3708 (2021). <https://doi.org/10.1109/TPAMI.2020.2988729>
7. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5939–5948 (2019)
8. Cheng, S., et al.: Deep stereo using adaptive thin volume representation with uncertainty awareness. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2524–2534 (2020)
9. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: a unified approach for single and multi-view 3d object reconstruction. In: *European Conference on Computer Vision*, pp. 628–644. Springer, Heidelberg (2016)

10. Dai, Y., Zhu, Z., Rao, Z., Li, B.: Mvs2: deep unsupervised multi-view stereo with multi-view symmetry. In: 2019 International Conference on 3D Vision (3DV), pp. 1–8. IEEE (2019)
11. Ding, Y., et al.: Transmvsnet: global context-aware multi-view stereo network with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8585–8594 (2022)
12. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(8), 1362–1376 (2009)
13. Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 873–881 (2015)
14. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 270–279 (2017)
15. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2495–2504 (2020)
16. Huang, B., Yi, H., Huang, C., He, Y., Liu, J., Liu, X.: M3vsnet: unsupervised multi-metric multi-view stereo network. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 3163–3167. IEEE (2021)
17. Ji, M., Gall, J., Zheng, H., Liu, Y., Fang, L.: Surfacenet: an end-to-end 3d neural network for multiview stereopsis. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2307–2315 (2017)
18. Karaboga, D., Basturk, B.: Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. In: International Fuzzy Systems Association World Congress, pp. 789–798. Springer, Heidelberg (2007)
19. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. *ACM Trans. Graph. (ToG)* **32**(3), 1–13 (2013)
20. Khot, T., Agrawal, S., Tulsiani, S., Mertz, C., Lucey, S., Hebert, M.: Learning unsupervised multi-view stereopsis via robust photometric consistency. *arXiv preprint [arXiv:1905.02706](https://arxiv.org/abs/1905.02706)* (2019)
21. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph. (ToG)* **36**(4), 1–13 (2017)
22. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. *Int. J. Comput. Vision* **38**(3), 199–218 (2000)
23. Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(3), 418–433 (2005)
24. Liao, J., Fu, Y., Yan, Q., Xiao, C.: Pyramid multi-view stereo with local consistency. In: *Computer Graphics Forum*, vol. 38, pp. 335–346. Wiley Online Library (2019)
25. Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(10), 2024–2039 (2015)
26. Liu, T., Ye, X., Zhao, W., Pan, Z., Shi, M., Cao, Z.: When epipolar constraint meets non-local operators in multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 18088–18097 (2023)
27. Luo, K., Guan, T., Ju, L., Huang, H., Luo, Y.: P-mvsnet: learning patch-wise matching confidence aggregation for multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10452–10461 (2019)

28. Luo, K., Guan, T., Ju, L., Wang, Y., Chen, Z., Luo, Y.: Attention-aware multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1590–1599 (2020)
29. Ma, X., Gong, Y., Wang, Q., Huang, J., Chen, L., Yu, F.: Epp-mvsnet: epipolar-assembling based depth prediction for multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5732–5740 (2021)
30. Mallick, A., Stückler, J., Lensch, H.: Learning to adapt multi-view stereo by self-supervision. arXiv preprint [arXiv:2009.13278](https://arxiv.org/abs/2009.13278) (2020)
31. Mao, W., Wang, M., Huang, H., Gong, M.: A robust framework for multi-view stereopsis. *Vis. Comput.* **38**(5), 1539–1551 (2022)
32. Marsaglia, G.: Choosing a point from the surface of a sphere. *Ann. Math. Stat.* **43**(2), 645–646 (1972)
33. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4460–4470 (2019)
34. Mi, Z., Di, C., Xu, D.: Generalized binary search network for highly-efficient multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12991–13000 (2022)
35. Peng, R., Wang, R., Wang, Z., Lai, Y., Wang, R.: Rethinking depth estimation for multi-view stereo: a unified representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8645–8654 (2022)
36. Romanoni, A., Matteucci, M.: Tapa-mvs: textureless-aware patchmatch multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10413–10422 (2019)
37. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision, pp. 501–518. Springer, Heidelberg (2016)
38. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 06), vol. 1, pp. 519–528. IEEE (2006)
39. Seitz, S.M., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. *Int. J. Comput. Vision* **35**(2), 151–173 (1999)
40. Tang, J., Han, X., Pan, J., Jia, K., Tong, X.: A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4541–4550 (2019)
41. Tang, J., Han, X., Tan, M., Tong, X., Jia, K.: Skeletonnet: a topology-preserving solution for learning mesh reconstruction of object surfaces from rgb images. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021)
42. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: efficient convolutional architectures for high-resolution 3d outputs. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2088–2096 (2017)
43. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6411–6420 (2019)
44. Tola, E., Lepetit, V., Fua, P.: Daisy: an efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(5), 815–830 (2009)

45. Wang, F., Galliani, S., Vogel, C., Speciale, P., Pollefeys, M.: Patchmatchnet: learned multi-view patchmatch stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14194–14203 (2021)
46. Wang, X., et al.: Multi-view stereo in the deep learning era: a comprehensive review. *Displays* **70**, 102102 (2021)
47. Wang, Y., et al.: Adaptive patch deformation for textureless-resilient multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1621–1630 (2023)
48. Wang, Y., Qian, Y., Li, Y., Gong, M., Banzhaf, W.: Artificial multi-bee-colony algorithm for k-nearest-neighbor fields search. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016, pp. 1037–1044 (2016)
49. Wu, S., Bertholet, P., Huang, H., Cohen-Or, D., Gong, M., Zwicker, M.: Structure-aware data consolidation. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(10), 2529–2537 (2017)
50. Xu, Q., Kong, W., Tao, W., Pollefeys, M.: Multi-scale geometric consistency guided and planar prior assisted multi-view stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022)
51. Xu, Q., Tao, W.: Multi-scale geometric consistency guided multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5483–5492 (2019)
52. Xu, Q., Tao, W.: Planar prior assisted patchmatch multi-view stereo. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12516–12523 (2020)
53. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4541–4550 (2019)
54. Yang, J., Mao, W., Alvarez, J.M., Liu, M.: Cost volume pyramid based depth inference for multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4877–4886 (2020)
55. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: depth inference for unstructured multi-view stereo. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 767–783 (2018)
56. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5525–5534 (2019)
57. Yao, Y., et al.: Blendedmvs: a large-scale dataset for generalized multi-view stereo networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1790–1799 (2020)
58. Yin, Z., Shi, J.: Geonet: unsupervised learning of dense depth, optical flow and camera pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1983–1992 (2018)
59. Zhang, J., Li, S., Luo, Z., Fang, T., Yao, Y.: Vis-mvsnet: visibility-aware multi-view stereo network. *Int. J. Comput. Vision* **131**(1), 199–214 (2023)
60. Zhang, Z., Peng, R., Hu, Y., Wang, R.: Geomvsnet: learning multi-view stereo with geometry perception. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21508–21518 (2023)
61. Zhu, G., Kwong, S.: Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl. Math. Comput.* **217**(7), 3166–3173 (2010). <https://doi.org/10.1016/j.amc.2010.08.049>. <https://www.sciencedirect.com/science/article/pii/S0096300310009136>



# Geometrically Consistent Light Field Synthesis Using Repaint Video Diffusion Model

Soyoung Yoon and In Kyu Park<sup>(✉)</sup> 

Department of Electrical and Computer Engineering, Inha University, Incheon 22212, Korea  
pik@inha.ac.kr

**Abstract.** We propose to repaint an image-to-video diffusion model to synthesize light fields that are geometrically consistent. Despite significant advancements in diffusion models for novel view synthesis, applying these models to generate a light field, *i.e.*, fronto-parallel multiple views, has been challenging because of persistent visual and geometric consistency issues. By utilizing advances in video diffusion, we extend the temporal consistency of video diffusion to the geometric consistency of multi-view settings. We fine-tune the image-to-video diffusion model framework for optimized multi-view diffusion by incorporating multi-view data with camera parameters. Furthermore, we propose integrating a repaint method during the sampling (denoising process) to achieve enhanced accurate camera control in multi-view diffusion, improving consistency by maintaining the known region in the input image. This approach enables the application of light field synthesis that requires precise camera control and demonstrates the ability of diffusion models to generate light fields with wide baselines, leveraging their unique generative power.

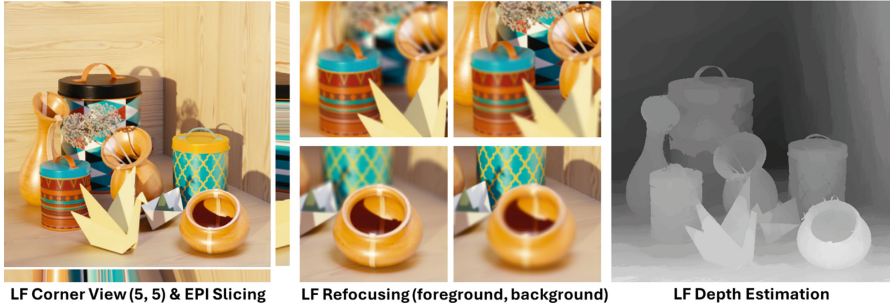
**Keywords:** Light Field · Novel View Synthesis · Video Diffusion Model

## 1 Introduction

Light field (LF) is a vector function that describes the direction and intensity of light rays from different angles in 3D space. It is mainly expressed as a 4D function of  $(x, y, u, v)$  for the intensity and direction of light rays passing through a 2D surface. Given that a light field image captures multi-view information, it offers the advantage of enabling various post-processing tasks in a single light field image. These tasks include viewpoint change, refocusing, and depth estimation. However, obtaining a light field image is a challenge. Multiple cameras [38] must be arranged horizontally and vertically, or a special light field camera [24] is required to obtain a light field. In addition, although light field can be acquired

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78456-9\\_10](https://doi.org/10.1007/978-3-031-78456-9_10).



**Fig. 1.** Results obtained from an LF synthesized from a single image. Left: Corner view (5, 5) for generated LF. Middle: LF refocusing on the foreground and background. Right: Depth estimation from generated LF.

through the special light field camera, a limitation exists, that is, only a narrow baseline is expressed. Hence, the importance of synthesizing light fields with a wide baseline has been raised.

Previous works aimed at synthesizing light fields have leveraged developments in deep learning, utilizing methods that combine deep neural networks with 3D representations. Works such as [1, 2, 11, 12, 30] successfully generated light field from a single image, but their scope was limited to light field with relatively narrow baselines or specific scenes. Within the domain of novel view synthesis (NVS), Neural Radiance Field (NeRF) [23] and 3D Gaussian Splatting [13] have received particular attention. NeRF employs volumetric rendering based on 3D representations to reconstruct target views inherently. Nonetheless, it requires scene-specific training and suffers from blurring problems when a significant deviation exists between the target and the sparsely provided input views. 3D Gaussian Splatting, although an attempt to address the limitations of NeRF, maintains the constraint of requiring scene-specific training.

Recent works on NVS [16–20, 27, 28, 36, 37, 40, 42] have progressed with diffusion models. This generative model approach has the advantage of producing plausible random samples from a learned conditional distribution, addressing issues related to unseen regions from the input viewpoint. However, a significant drawback is the difficulty in generating geometrically consistent sequences with almost no geometric prior available. To address the challenge of maintaining geometric view consistency, recent works have employed diffusion models with cross-attention modules in conjunction with conditioning techniques on the input viewpoint or training alongside 3D networks to preserve geometric consistency.

Despite the progress in NVS, particularly for object-centric scenes, considerable research on light field synthesis has not yet progressed. Existing 360-degree NVS models do not fit light field synthesis, which requires multiple parallel viewpoints and significant consistency among these viewpoints. Therefore, we propose to extend the video diffusion model to a multi-view diffusion model specifically for light field synthesis. By leveraging the video diffusion model, we demonstrate

the feasibility of transforming temporal consistency into multi-view consistency. This method can resolve the issue of consistency among viewpoints without additional 3D networks and representations. Light field synthesis demands the representation of complex and precise camera movements and relies heavily on information obtained from input images. To address these concerns, our approach utilizes depth-based warping from an input view to target views during the sampling process and fills holes in unseen regions through inpainting without any additional training. We integrate a RePaint [21] method with the multi-view diffusion model and achieve enhanced accurate light field synthesis. An example of the generated light field result and its applications is shown in Fig. 1.

The novelty and contribution of this paper are summarized as follows.

- We extend the video diffusion model into a multi-view diffusion model for light field synthesis by fine-tuning it with multi-view data and relative camera parameters
- We employ repainting at every frame within the multi-view diffusion process for synthesizing precise viewpoint light field images
- We demonstrate a geometrically consistent light field synthesis on a single image that is commonly obtainable, showing the robustness of our method

## 2 Related Works

### 2.1 Light Field Synthesis

Light field synthesis began as angular super-resolution, generating a dense light field from a sparse light field. The work [12] proposed a method for synthesizing an  $8 \times 8$  grid of sub-aperture images (SAIs) from four-corner SAIs. This field has since evolved to include research on generating light fields from a single input image. Srinivasan *et al.* [30] were among the first to demonstrate a learning-based approach for light field synthesis from a monocular image. Their method involved estimating occluded regions to achieve light field synthesis. Ivan *et al.* [11] and Bae *et al.* [1] sought to use geometric information through appearance flow and proposed a new loss function. Li *et al.* [15] suggested a method for light field synthesis that adaptively estimates multi-plane images (MPI) representations, allowing for the accommodation of data with various geometric scales. Similarly, Bak *et al.* [2] proposed synthesizing light fields by estimating layers in a per-pixel manner, thereby achieving accurate layer representations. Unlike other methods, we have approached light field synthesis via diffusion. Although LFDiff [7] has recently been proposed for light field synthesis utilizing diffusion, it synthesizes from macro-pixels that include spatial-angular representations through a position-aware conditioning scheme. This method requires synthesis on a patch-by-patch basis across the entire image. Such an approach leads to issues such as fragmentation in each patch and the limitation of only being able to synthesize  $5 \times 5$  light fields. By contrast, we utilize video diffusion to simultaneously synthesize each of the SAIs as individual frames, enabling us to synthesize light fields larger than  $5 \times 5$ .

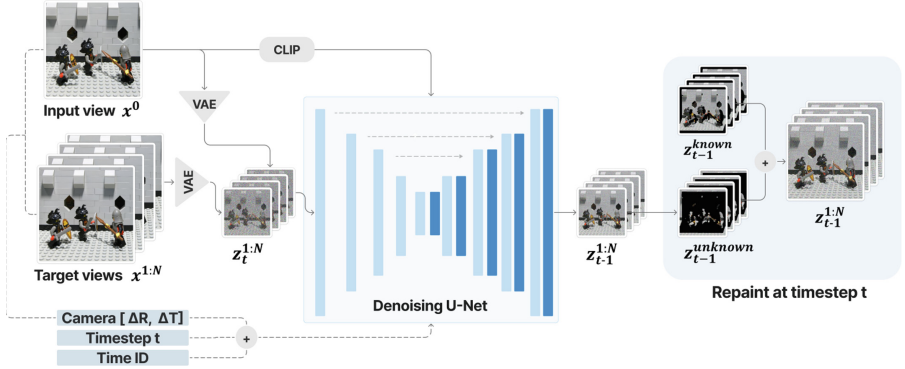


## 2.2 Novel View Synthesis with Diffusion

Recently, NVS research has progressed with diffusion models, which are a type of generative model, owing to their ability to generate plausible samples from the learned conditional distribution. This capability effectively addresses the issue of unseen regions from the input viewpoint. The first application of diffusion to NVS, 3DiM [36], utilized conditioning techniques on input viewpoints and camera poses to create a novel view from the input image. However, training from scratch presented a challenge because of insufficient 3D data. Zero-1-to-3 [18] addressed this challenge by fine-tuning a pretrained large image diffusion model for NVS, maintaining 2D priors while learning from 3D data. Issues such as camera alignment and consistency across multiple views persisted, prompting recent studies to either train additional 3D networks [16, 17] or employ extra 3D representations [19, 40]. Efforts have also been made to train multi-view diffusion models [20, 27, 28, 37, 42] that generate multiple views simultaneously to ensure consistency among views. However, these approaches often result in models that can only generate fixed views or require post-processing. Similarly, other methods [5, 6, 10] use pretrained diffusion models for multi-view inpainting, integrating the results with scene geometry like 3D meshes or point clouds to maintain consistency. While these inpainting techniques are similar to ours, they focus more on merging views with 3D scene geometry. In contrast, our method enhances view consistency by fine-tuning a pretrained video diffusion model for multi-view synthesis, overcoming limitations seen in image diffusion models, and providing improved consistency in complex scenes without extensive 3D reconstruction.

## 2.3 Video Diffusion Model

Existing multi-view models have evolved from image diffusion models that integrate cross-view attention. Nonetheless, for improved multi-view consistency, the temporal priors of video diffusion models can be leveraged to extend NVS models. ViVid-1-to-3 [14] utilized image and video diffusion models for NVS to generate consistent views. Similarly, SVD-MV [4], IM-3D [22], and SV3D [32] fine-tune a video diffusion model for NVS. However, their capability is confined to rendering only 360-degree views of a 3D object. Different from previous video diffusion models designed for object-centered NVS, our approach targets NVS for general scenes such as a light field. Recent developments in video generation have introduced models with more precise camera control with general scenes, such as MotionCtrl [34] and CameraCtrl [8], which aim to improve camera pose adjustments. While MotionCtrl relies on numerical values, CameraCtrl uses plücker embeddings for better accuracy. However, both face challenges with movement scaling and generating accurate light fields when detailed camera parameters are involved. To address these issues, we introduce repaint techniques during sampling in video diffusion models, which enhance camera control and enable the synthesis of high-quality light fields.



**Fig. 2.** Overall framework of our proposed method. Noise latents  $z_t^{1:N}$  are processed by the denoising UNet  $\epsilon_\theta$ , which incorporates the relative camera parameters  $(\Delta\mathbf{R}, \Delta\mathbf{T})$  for each multi-view. The denoised latents  $z_{t-1}^{1:N}$  are then combined with the warped images at the repaint phase and modified. The final multi-view images are generated through iteratively denoising steps.

## 3 Proposed Method

### 3.1 Geometrically Consistent Multi-view Diffusion Model

We propose to leverage the consistency of video diffusion for light field synthesis. We extend a video diffusion model to a multi-view diffusion model by fine-tuning it with a multi-view dataset including light fields and corresponding camera parameters. This process allows us to learn the relationship between the differences in viewpoints of each multi-view, thereby enabling its application in synthesizing light fields. The overview of the entire architectural structure is described in Fig. 2.

**Video Diffusion Architecture.** We adopt the architecture of Stable Video Diffusion (SVD) [3, 4], a large-scale open-set video diffusion model, consisting of a temporal-aware UNet with multiple layers. Each layer includes a sequence of one residual block with Conv3D layers and two spatial and temporal transformer blocks with attention layers. To adapt SVD, which inherently possesses temporal consistency, to a multi-view diffusion model for light field, we fine-tune it using a multi-view dataset. Camera parameters are incorporated into the model to differentiate and learn the relationships among the views, thereby enabling synthesis for the SAIs of the light field.

**Camera Embedding.** In the SVD model, temporal consistency and motion are encapsulated into vector conditions comprising ‘fps id’, ‘motion bucket id’, and ‘noise augmentation strength’. These conditions are combined and added to the original time embedding to incorporate temporal dynamics into the diffusion process. To enhance this architecture for multi-view consistency and motion,

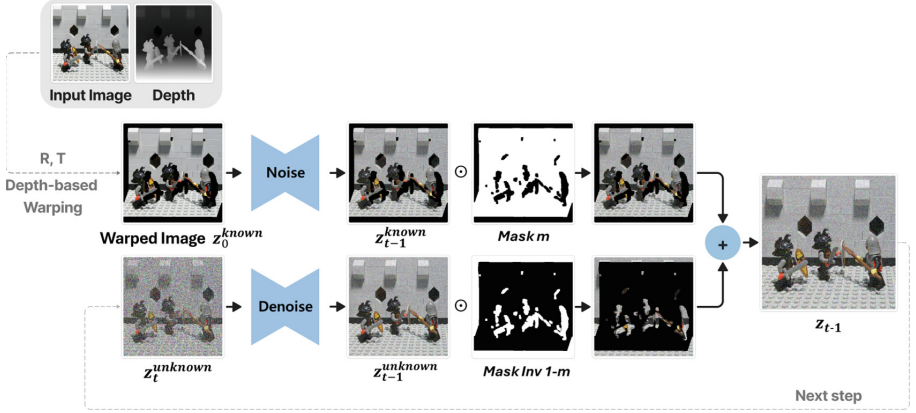
the approach involves calculating the relative rotation  $R$  and translation  $T$  from the camera matrices of each input multi-view and the source view. These values capture the rotation and movement required to transition between the views, thereby enhancing the model with spatial dynamics alongside its inherent temporal capabilities. The camera pose embeddings, derived from these  $R$  and  $T$  values, are concatenated, linearly transformed, and then merged with the noise timestep embedding. This enhanced embedding is supplied to every residual block within the UNet architecture. Such embeddings are integrated into the output features at each block, ensuring that the model incorporates spatial (multi-view consistency and camera motion) and temporal information throughout the video diffusion process. This approach allows the SVD model to synthesize images that maintain temporal coherence and display consistent, dynamic perspectives across multiple views, thus meeting the demands of multi-view and light field synthesis.

**Loss Function for Training.** The loss function for training is defined as follows. Given multi-view data that include the source view  $\mathbf{x}^0$ , target view  $\mathbf{x}^{1:N}$ , and camera poses  $(\Delta\mathbf{R}, \Delta\mathbf{T}) \in \mathbf{P}$ , the training proceeds with conditioning on the source view and camera poses  $\mathbf{y} = (\mathbf{x}^0, \mathbf{P})$ . Each viewpoint image (source and target views) is passed through an encoder to be represented in a latent space  $\mathbf{z}^0, \mathbf{z}^{1:N}$ , and the source view is concatenated with the noise-augmented target view. CLIP [26] encoding is also employed to provide cross-attention within the network.

$$\mathcal{L} = \mathbb{E}_{\epsilon, t, (\mathbf{z}_0^{1:t}, \mathbf{z}^0, \mathbf{P})} [\| \epsilon - \hat{\epsilon}_\theta(\mathbf{z}_t^{1:N}, t, \mathbf{y}) \|_2^2] \quad (1)$$

### 3.2 Repainting for Light Field Synthesis

We enhance the video diffusion model to a multi-view diffusion model capable of synthesizing light fields, incorporating camera embeddings that account for relative rotation and translation to ensure consistency across generated multiple views. However, including only the magnitudes of relative rotation and translation poses a challenge in precisely managing camera poses, which is particularly crucial for camera movement and light field synthesis in general scenes, beyond only object-centered view synthesis. To address this challenge, we employ depth-based image warping to preserve portions of the image that can be maintained from the input view while subtly adjusting camera movements. For areas within the warped image that require prediction, we integrate RePaint [21] for inpainting directly within the multi-view diffusion model, allowing us to predict the necessary regions. This strategy enhances light field synthesis by obviating the need for separate inpainting training. The process of repainting in multi-view diffusion is illustrated in Fig. 3.



**Fig. 3.** Detailed process of repaint in multi-view diffusion. The entire process operates within the latent space for every view. Given an input image and its depth, depth-based warping is applied to sample the known parts as warped images, and the outputs from the denoising UNet are used to sample the unknown (inpainted) parts, influencing each denoising step’s output.

**Depth-Based Image Warping.** Depth information for the source image is required to perform warping for a desired target viewpoint. For acquiring depth information, monocular depth estimation (MDE) is utilized. MiDaS [25] serves as the MDE model, converting the relative inverse depth obtained into depth for warping purposes.

The first step in the warping process involves transforming the pixel coordinates  $(x, y)$  to camera coordinates. This transformation uses the depth  $d$  of the pixel and the inverse of the intrinsic matrix  $\mathbf{K}_1$  of the original view. The camera coordinates  $\mathbf{p}_c$  are calculated using the depth and the inverse intrinsic matrix to convert the pixel coordinates, as follows:

$$\mathbf{p}_c = d \cdot \mathbf{K}_1^{-1} \cdot [x, y, 1]^T \quad (2)$$

After converting into camera coordinates,  $\mathbf{p}_c$  is transformed to the target camera frame using the transformation matrix  $\mathbf{T}_{21} = \mathbf{T}_2 \cdot \mathbf{T}_1^{-1}$ , derived from the extrinsic parameters of the original and target cameras.

$$\mathbf{p}'_c = \mathbf{K}_2 \cdot \mathbf{T}_{21} \cdot \mathbf{p}_c \quad (3)$$

Finally, weights based on depth and the proximity of pixels are calculated to align the transformed coordinates with the pixel grid. These weights are then used to distribute the transformed coordinates among the nearest pixels, thereby effectively achieving warping to the target viewpoint with consideration for depth and spatial relationships.

$$[x', y', d']^T = \frac{\mathbf{p}'_c}{\mathbf{p}'_{c,z}} \quad (4)$$

**Repaint for Sampling Light Field.** For inpainting warped images, we apply a repaint to multi-view diffusion without separate training on mask distributions, relying solely on a video diffusion model trained on multi-view data. This approach facilitates predicting the masked unknown parts, thereby eliminating the necessity for inpainting models specifically trained with mask conditions tailored to the camera movements and intervals typically used in view synthesis.

The repaint method works together with the denoising process at each step. During the reverse step from  $\mathbf{z}_t$  to  $\mathbf{z}_{t-1}$ , known parts are sampled to  $\mathbf{z}_{t-1}$  by adding noise from the warped image, whereas unknown parts are sampled by removing the predicted noise from  $\mathbf{z}_t$ .

$$\mathbf{z}_{t-1}^{\text{known}} \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad (5)$$

$$\mathbf{z}_{t-1}^{\text{unknown}} \sim \mathcal{N}(\mu_\theta(\mathbf{z}_t, t), \Sigma_\theta(\mathbf{z}_t, t)) \quad (6)$$

Given a mask  $\mathbf{m}$ , the unknown area can be denoted as  $\mathbf{m} \odot \mathbf{z}_{t-1}^{\text{unknown}}$  and the known area as  $(1 - \mathbf{m}) \odot \mathbf{z}_{t-1}^{\text{known}}$ . Merging the two areas, we use the following equation to calculate  $\mathbf{z}_{t-1}$ :

$$\mathbf{z}_{t-1} = (1 - \mathbf{m}) \odot \mathbf{z}_{t-1}^{\text{known}} + \mathbf{m} \odot \mathbf{z}_{t-1}^{\text{unknown}} \quad (7)$$

This process is repeated at every denoising step, ensuring that the masked area is filled to match the known region. The warped image and mask are passed to the multi-view diffusion model for each target frame, and as the denoising process proceeds, the repaint process is integrated with the multi-view diffusion. This allows the known region warped to the target viewpoint to facilitate accurate viewpoint transitions while filling the empty areas through repaint. The empty areas are filled using the video diffusion model trained on multi-view data, thereby ensuring consistent multi-view generation.

## 4 Experimental Results

### 4.1 Experimental Settings

To evaluate our proposed method, we present experimental results that demonstrate light field synthesis from a single center-view image using existing light field datasets. We also provide light field synthesis results from a single image with general scenes, tested across various baselines and camera movements to illustrate our method’s broad applicability and generalization. It is important to note that recent NVS methods, such as those in [16–18], among others, were excluded from our comparison. These methods are primarily designed for object-centric NVS and represent camera parameters with only 4 degrees of freedom (DoF), focusing on rotation, which includes azimuth, elevation, camera orientation, and distance (radius). As a result, they are not suitable for light field synthesis, which requires handling more complex camera movements.

**Table 1.** Quantitative results (PSNR  $\uparrow$ /SSIM  $\uparrow$ /LPIPS  $\downarrow$ ) of LF synthesis. The best results are marked in bold and the second-best results are underlined.

Method	HCI-new			HCI-old			STFGantry		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Srinivasan <i>et al.</i> [30]	27.175	0.7678	0.061	29.608	0.7946	0.053	20.874	0.6746	0.086
Li <i>et al.</i> [15]	27.202	0.7782	0.060	31.673	0.8806	0.046	21.651	0.7133	0.072
Bak <i>et al.</i> [2]	27.930	0.7955	0.066	31.9327	0.8658	0.047	21.747	0.7021	0.076
LFdiff [7]	30.665	<b>0.9135</b>	<b>0.025</b>	<u>33.600</u>	<b>0.9207</b>	<b>0.023</b>	<u>24.264</u>	<u>0.7850</u>	<u>0.068</u>
Ours	<b>37.081</b>	<u>0.9093</u>	<u>0.048</u>	<b>35.308</b>	<u>0.9018</u>	<u>0.031</u>	<b>33.344</b>	<b>0.8392</b>	<b>0.049</b>

**Training Settings.** We utilize 20 scenes from the light field synthetic dataset HCI-new [9] and 9 scenes from the real light field dataset STFGantry [31] as our training data. Additionally, to facilitate light field synthesis in general scenes and allow for flexible camera control, we utilize a subset of data from RealEstate10K [43] dataset, which includes multiple camera trajectories, comprising 105 scenes. During the training stage, we also utilize the camera parameters as inputs along with multi-view. During the training stage, we also use the camera parameters as inputs along with multi-view and fine-tune a temporal-aware UNet for video diffusion, leveraging the AdamW optimizer at a learning rate of  $1e-4$  and a batch size of 1. We set a noise schedule based on a log-normal distribution to adjust the noise levels dynamically across training timesteps.

**Inference Settings.** To generate samples of the light field, we use fine-tuned multi-view diffusion with the repaint technique, and a single image (center view) along with its corresponding depth map as input. On the basis of the camera parameters of target viewpoints, we employ depth-based warping to generate warped images and masks, which are then used in the repaint sampling process. We employ 25 steps of the deterministic DDIM sampler [29] for this purpose. The repaint is applied at a rate of 70% during these 25 steps. Details on the application rate of repaint can be found in *Supplementary Material*.

## 4.2 Light Field Synthesis with Single View

For light field synthesis, we use camera parameters provided with the data and target viewpoint images that are depth-based warped. These warped target viewpoint images are then guided in the sampling (denoising process) through the repaint for synthesis. We demonstrate the results of synthesizing a  $5 \times 5$  light field from a single-view image for the HCI-new [9], HCI-old [35], and STFGantry [31] datasets, comparing them quantitatively and qualitatively with existing light field synthesis methods.

**Quantitative Evaluation.** The quantitative results of our  $5 \times 5$  light field synthesis are presented in Table 1, where we compare the results of our method with

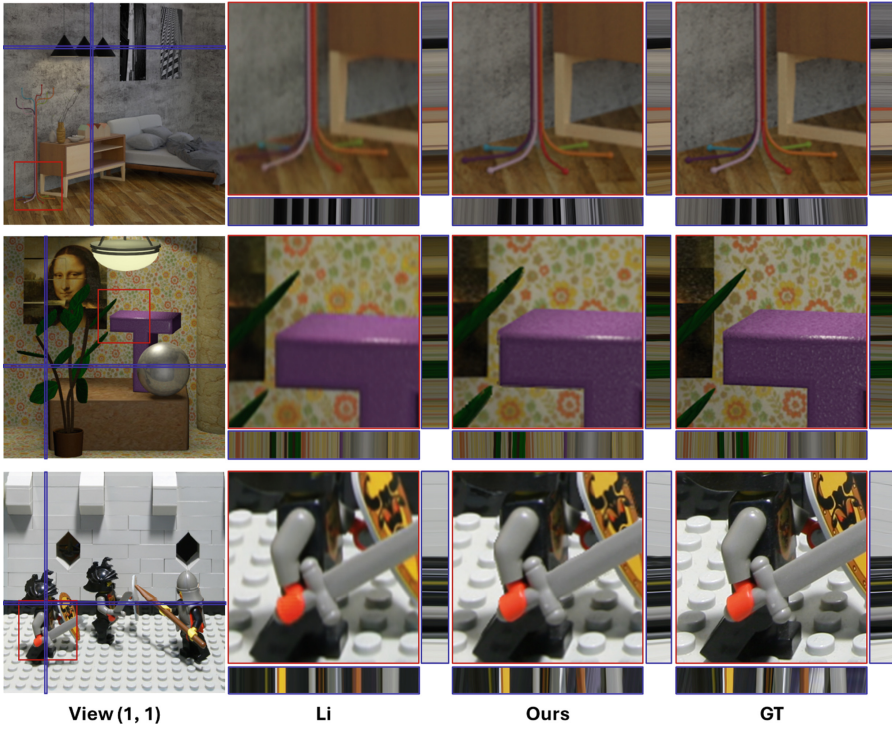


Fig. 4. Qualitative results of LF synthesis by SAI and EPI. Comparison of the zoom-in results of the synthesized corner view (1,1) with the GT.

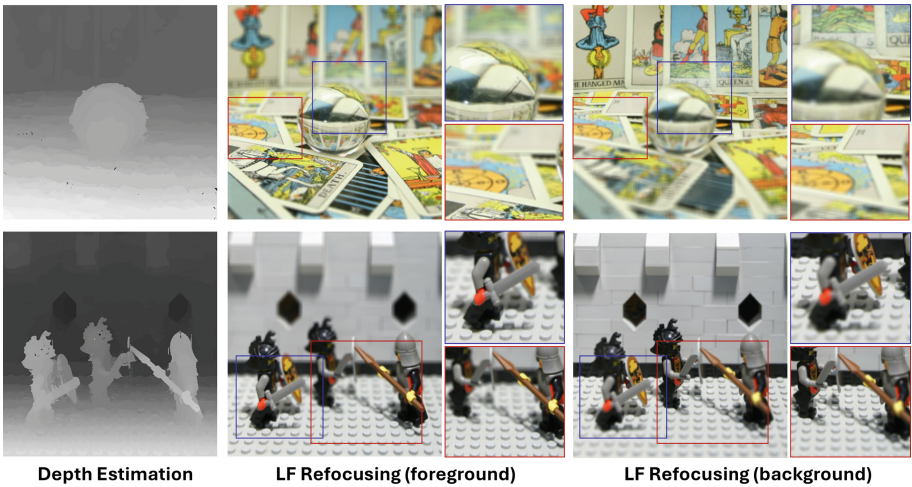


Fig. 5. Qualitative results of LF synthesis by depth estimation and refocusing from the synthesized LF scenes of STFgantry.

those of Srinivasan *et al.* [30], Li *et al.* [15], Bak *et al.* [2], and LFDiff [7]. Relative to other light field synthesis methods, our approach shows the first- or second-best performance quantitatively. Although we place second in the HCI-new and HCI-old datasets, our results are closely competitive, and we achieve a significant lead as the first in the Stanford dataset. Interpretation of these outcomes suggests that the comparison methods, except for that of Srinivasan *et al.*, include warping results through shifting based on light field disparity. This finding likely explains the good results observed for light fields with smaller baselines, such as HCI-new and HCI-old, given that they accurately represent the dense and narrow motion of light fields. Conversely, our method employs general homography matrix-based warping for repaint, which leads to significantly improved results in datasets with relatively larger baselines, such as the Stanford dataset, compared with the results of other methodologies (achieving improvements in PSNR +9.08 dB, SSIM [33] +0.054, and LPIPS [41] -0.019). This result indicates the potential for synthesizing light fields with broad baselines using our approach.

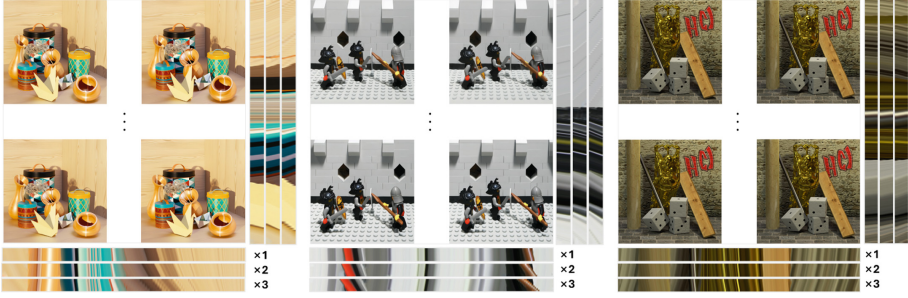
**Qualitative Evaluation.** Figure 4 shows the results of synthesizing a  $5 \times 5$  light field, comparing our method with that of Li *et al.* [15] on different datasets. It first presents the corner views of the light fields synthesized using our proposed method and then provides a zoomed-in view of the corner SAI for a clear comparison with the method of Li *et al.* and the ground truth (GT). The comparison shows that our method yields results that are sharper and closer to the GT. Moreover, the movement directions in the epipolar plane images (EPIs) extracted results closely resemble those of the GT, demonstrating the effectiveness of our approach in capturing dynamic aspects of the scene. In Fig. 5, we present the depth estimation results for the synthesized light field using the depth estimation algorithm CAE [39] and refocusing results. Depth estimation and refocusing work effectively, indicating that the synthesized light fields successfully preserve geometric depth information. This finding demonstrates our method’s capability to maintain accurate geometric details in the light field synthesis.

### 4.3 Wide Baseline Light Field

We present the results when the baseline of a  $5 \times 5$  light field is expanded to 2 and 3 times its original distance. Moreover, we demonstrate that light field synthesis from a single image is feasible even in generally captured scenes, and synthesis along non-light field camera trajectories is achievable.

**Evaluation of LF Synthesis (Baseline  $\times 2$ ,  $\times 3$ ).** We exhibit the results for light fields with expanded baselines. Initially, we maintain the same baseline but generate a  $9 \times 9$  light field, doubling the number of viewpoint images horizontally and vertically, to show a light field with twice the baseline. This outcome, being verifiable against the GT, presents quantitative results, as shown in Table 2. Although the increase in viewpoint count to  $9 \times 9$  leads to a challenging synthesis scenario and slightly decreased quantitative metrics such as PSNR, SSIM, and LPIPS, the results remain notably robust. This finding demonstrates that





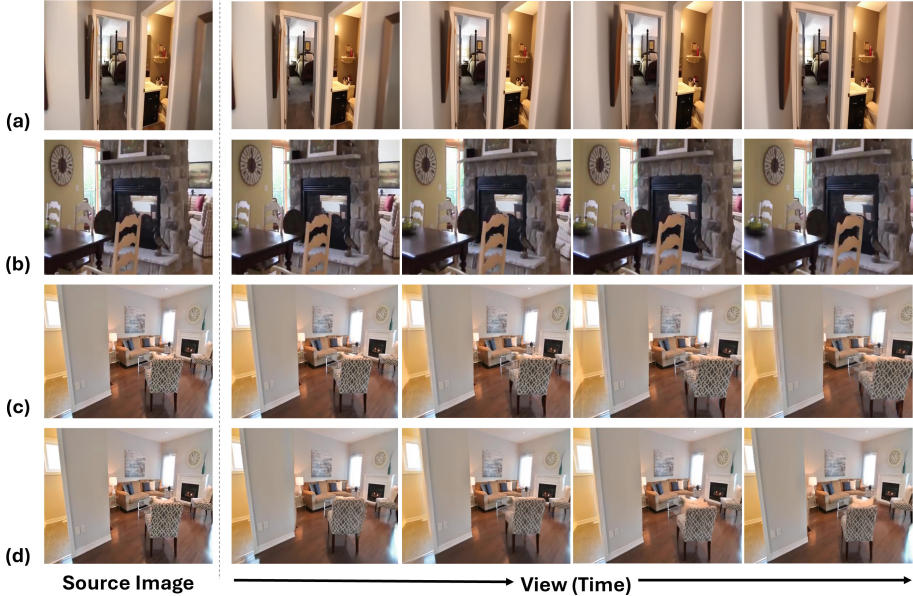
**Fig. 6.** Qualitative results of LF synthesis by SAIs (corner views of LF with baseline  $\times 3$ ) and EPIs (with baseline  $\times 1$ ,  $\times 2$ ,  $\times 3$ ).

**Table 2.** Quantitative results (PSNR  $\uparrow$ /SSIM  $\uparrow$ /LPIPS  $\downarrow$ ) of LF synthesis ( $5\times 5$ ,  $9\times 9$ ).

	5 $\times$ 5 LF			9 $\times$ 9 LF		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
HCI-new	37.081	0.9093	0.048	35.253	0.8637	0.066
HCI-old	35.308	0.9018	0.031	34.367	0.8634	0.041
STFGantry	33.344	0.8392	0.049	32.467	0.7782	0.070

our synthesis approach can effectively handle high complexity while maintaining an acceptable quality threshold. The robust performance in the face of increased synthesis difficulty is illustrated in Fig. 6, which presents results with a baseline expanded to 3 times alongside the twice-expanded results, using SAIs and EPIs to show the effect of increased camera motion. The results demonstrate a consistent increase in camera motion through EPIs and SAIs, highlighting the effectiveness of our approach in generating high-quality complex light fields.

**NVS from a Single Image.** We demonstrate the feasibility of synthesizing light fields from a single image in commonly captured scenes and illustrate the potential for synthesis along non-light field camera trajectories in Fig. 7. The input of the same single image can yield diverse synthesized viewpoints, depending on the selected camera trajectory. This diversity is guided by warping the image with specific camera embeddings and using the repaint method. The last row in Fig. 7, particularly those from the four-corner views when synthesizing the light field, showcases the robust capability of our system to manage viewpoint synthesis with camera control from a general image while maintaining consistency across synthesized viewpoints. These experimental results underline our approach’s practical applicability and versatility in realistic settings.



**Fig. 7.** Qualitative results of viewpoint synthesis along a continuous camera trajectory from a single frame for the RealEstate10K dataset. (a), (b), and (c) show synthesized views along different points of the camera trajectory; (d) shows a corner view of LF synthesized from the same single frame as in (c).

## 5 Conclusion

In this paper, we introduced a novel approach to light field synthesis using diffusion models, addressing the challenge of geometric consistency across multiple views. We successfully demonstrated the conversion of temporal into multi-view consistency, significantly improving light field synthesis from single images. Our contributions included extending video diffusion models for multi-view synthesis and integrating a repaint method to enhance camera control precision. Through the experimental results of light field synthesis, our work demonstrated outcomes comparable to or surpassing those of existing light field methods. Furthermore, by presenting extended baseline light field results and generalization across different camera trajectories, we illustrated the versatility and applicability of our approach in various scenes.

**Limitations and Future Work.** Given the challenges of memory constraints and the need for consistency, producing light fields with dense SAIs at high angular resolutions remains challenging. The use of latent video diffusion, which operates in latent space, leads to detail loss when transitioning to pixel space compared with the original input. Nevertheless, advancements in variational

autoencoder are expected to improve the generation of light fields, aligning them closely and consistently with the input image.

**Acknowledgment.** This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.RS-2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development (Inha University) and No.RS-2021-II212068, Artificial Intelligence Innovation Hub and IITP-2024-RS-2024-00360227, Leading Generative AI Human Resources Development. This work was supported by Inha University Research Grant.

## References

1. Bae, K., Ivan, A., Nagahara, H., Park, I.K.: 5D light field synthesis from a monocular video. In: Proceedings of 25th International Conference on Pattern Recognition (ICPR), pp. 7157–7164 (2021)
2. Bak, J., Park, I.K.: Light field synthesis from a monocular image using variable LDI. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 3399–3407 (2023)
3. Blattmann, A., et al.: Align your latents: high-resolution video synthesis with latent diffusion models. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 22563–22575 (2023)
4. Blattmann, A., et al.: Stable video diffusion: scaling latent video diffusion models to large datasets. arXiv preprint [arXiv:2311.15127](https://arxiv.org/abs/2311.15127) (2023)
5. Chung, J., Lee, S., Nam, H., Lee, J., Lee, K.M.: LucidDreamer: domain-free generation of 3D gaussian splatting scenes. arXiv preprint [arXiv:2311.13384](https://arxiv.org/abs/2311.13384) (2023)
6. Fridman, R., Abecasis, A., Kasten, Y., Dekel, T.: SceneScape: text-driven consistent scene generation. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS), vol. 36, pp. 39897–39914 (2024)
7. Gao, R., Liu, Y., Xiao, Z., Xiong, Z.: Diffusion-based light field synthesis. arXiv preprint [arXiv:2402.00575](https://arxiv.org/abs/2402.00575) (2024)
8. He, H., et al.: CameraCtrl: enabling camera control for text-to-video generation. arXiv preprint [arXiv:2404.02101](https://arxiv.org/abs/2404.02101) (2024)
9. Honauer, K., Johannsen, O., Kondermann, D., Goldluecke, B.: A dataset and evaluation methodology for depth estimation on 4D light fields. In: Proceedings of Asian Conference on Computer Vision (ACCV), pp. 19–34 (2016)
10. Hu, Y., et al.: O<sup>2</sup>-recon: completing 3D reconstruction of occluded objects in the scene with a pre-trained 2D diffusion model. In: Proceedings of AAAI Conference on Artificial Intelligence, vol. 38, pp. 2285–2293 (2024)
11. Ivan, A., Park, I.K.: Joint light field spatial and angular super-resolution from a single image. *IEEE Access* **8**, 112562–112573 (2020)
12. Kalantari, N.K., Wang, T.C., Ramamoorthi, R.: Learning-based view synthesis for light field cameras. *ACM Trans. Graph. (TOG)* **35**(6), 1–10 (2016)
13. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph. (TOG)* **42**(4), 1–14 (2023)
14. Kwak, J.G., Dong, E., Jin, Y., Ko, H., Mahajan, S., Yi, K.M.: ViVid-1-to-3: novel view synthesis with video diffusion models. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6775–6785 (2024)

15. Li, Q., Khademi Kalantari, N.: Synthesizing light field from a single image with variable MPI and two network fusion. *ACM Trans. Graph. (TOG)* **39**(6), 1–229 (2020)
16. Liu, M., et al.: One-2-3-45++: fast single image to 3D objects with consistent multi-view generation and 3D diffusion. In: *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10072–10083 (2024)
17. Liu, M., et al.: One-2-3-45: any single image to 3d mesh in 45 seconds without per-shape optimization. In: *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, pp. 22226–22246 (2023)
18. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: zero-shot one image to 3D object. In: *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9298–9309 (2023)
19. Liu, Y., et al.: SyncDreamer: generating multiview-consistent images from a single-view image. In: *Proceedings of International Conference on Learning Representations (ICLR)* (2024)
20. Long, X., et al.: Wonder3D: single image to 3D using cross-domain diffusion. In: *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9970–9980 (2024)
21. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: RePaint: inpainting using denoising diffusion probabilistic models. In: *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11461–11471 (2022)
22. Melas-Kyriazi, L., et al.: IM-3D: iterative multiview diffusion and reconstruction for high-quality 3D generation. In: *Proceedings of Forty-first International Conference on Machine Learning (ICML)* (2024)
23. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020. LNCS*, vol. 12346, pp. 405–421. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58452-8\\_24](https://doi.org/10.1007/978-3-030-58452-8_24)
24. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: *Light Field Photography with a Hand-held Plenoptic Camera*. Research Report CSTR 2005-02, Stanford university (2005)
25. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **44**(3), 1623–1637 (2020)
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695 (2022)
27. Shi, R., et al.: Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110* (2023)
28. Shi, Y., et al.: MVDream: multi-view diffusion for 3D Generation. In: *Proceedings of Twelfth International Conference on Learning Representations (ICLR)* (2024)
29. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: *Proceedings of Eighth International Conference on Learning Representations (ICLR)* (2020)
30. Srinivasan, P.P., Wang, T., Sreelal, A., Ramamoorthi, R., Ng, R.: Learning to synthesize a 4D RGBD light field from a single image. In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 2243–2251 (2017)
31. Vaish, V., Adams, A.: The (new) Stanford Light Field Archive (2008). <http://lightfield.stanford.edu/papers.html>

32. Voleti, V., et al.: SV3D: novel multi-view synthesis and 3D generation from a single image using latent video diffusion. arXiv preprint [arXiv:2403.12008](https://arxiv.org/abs/2403.12008) (2024)
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process. (TIP)* **13**(4), 600–612 (2004)
34. Wang, Z., et al.: MotionCtrl: a unified and flexible motion controller for video generation. In: *Proceedings of ACM SIGGRAPH 2024 Conference*, pp. 1–11 (2024)
35. Wanner, S., Meister, S., Goldluecke, B.: Datasets and benchmarks for densely sampled 4D light fields. In: *Proceedings of Conference on Vision, Modeling, and Visualization (VMV)*, vol. 13, pp. 225–226 (2013)
36. Watson, D., Chan, W., Brualla, R.M., Ho, J., Tagliasacchi, A., Norouzi, M.: Novel view synthesis with diffusion models. In: *Proceedings of Eleventh International Conference on Learning Representations (ICLR)* (2023)
37. Weng, H., et al.: Consistent123: improve consistency for one image to 3D object synthesis. arXiv preprint [arXiv:2310.08092](https://arxiv.org/abs/2310.08092) (2023)
38. Wilburn, B., et al.: High performance imaging using large camera arrays. *ACM Trans. Graph. (TOG)* **24**(3), 765–776 (2005)
39. Park, I.K., Lee, K.M.: Robust light field depth estimation using occlusion-noise aware data costs. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **40**(10), 2484–2497 (2018)
40. Yang, J., Cheng, Z., Duan, Y., Ji, P., Li, H.: ConsistNet: enforcing 3D consistency for multi-view images diffusion. In: *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7079–7088 (2024)
41. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595 (2018)
42. Zheng, C., Vedaldi, A.: Free3D: consistent novel view synthesis without 3D representation. In: *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9720–9731 (2024)
43. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph. (TOG)* **37**(4), 1–12 (2018)



# An Empirical Evaluation of the Impact of Solar Correction in NeRFs for Satellite Imagery

Devjyoti Chakraborty<sup>1</sup>, Kriti Ghosh<sup>1</sup>, Zaki Sukma<sup>1,2</sup>(✉), In Kee Kim<sup>1,2</sup>,  
Lakshmith Ramaswamy<sup>1,2</sup>, Suchendra M. Bhandarkar<sup>1,2</sup>,  
and Deepak R. Mishra<sup>2,3,4</sup>

<sup>1</sup> School of Computing, University of Georgia, Athens, GA 30602, USA

<sup>2</sup> Institute for Artificial Intelligence, University of Georgia, Athens, GA 30602, USA  
zaki.sukma@uga.edu

<sup>3</sup> Center for Geospatial Research, University of Georgia, Athens, GA 30602, USA

<sup>4</sup> Department of Geography, University of Georgia, Athens, GA 30602, USA

**Abstract.** Neural Radiance Fields (NeRFs) have received significant attention in reconstruction of complex 3D scenes due to their novel view synthesis capabilities. Their volumetric scene representation capabilities and flexibility in handling inherent challenges in satellite imagery distinguish NeRFs from previous approaches to satellite image analysis. Nonetheless, the evaluation of NeRF variants within the context of satellite image analysis remains limited. This study presents a comprehensive assessment of the NeRF and its variants using quantitative and qualitative metrics. We systematically evaluate and compare the performance of NeRF, Sat-NeRF, Shadow NeRF (S-NeRF), and their solar corrective variants. Our analysis explores hyperparameter tuning, rendering quality, memory utilization, and computational requirements while showing how NeRF variants tailored for satellite imagery show promise. Given the unique challenges presented by satellite imagery, this comparative study presents a thorough evaluation of various NeRF variants on an expanded dataset and offers insights into performance and efficiency trade-offs.

**Keywords:** NeRF · Satellite Images · 3D Reconstruction

## 1 Introduction

The proliferation of commercial satellites and the wide availability of datasets has significantly advanced research in the 3D reconstruction of satellite images [1, 7]. This field is actively explored at the intersection of computer vision and remote sensing. Various methodologies, such as Structure from Motion (SfM) [12], Multi-View Stereo [14], and Convolutional Neural Networks [13], have been developed to create 3D models from satellite imagery.

---

D. Chakraborty, K. Ghosh and Z. Sukma—**Co-first authors:** D. Chakraborty, K. Ghosh, and Z. Sukma have contributed equally to this work.

Neural Radiance Fields (NeRF) [9] has recently attracted increasing attention due to its capacity to implicitly model 3D object representations and synthesize novel views. NeRF variants such as Shadow-NeRF (S-NeRF) [3] and Satellite NeRF (Sat-NeRF) [8], have also been developed in 3D satellite image reconstruction. Each model adapts the original NeRF paradigm to address specific challenges in 3D reconstruction that manifest in the context of distinct satellite imaging applications.

Despite the novelty and demonstrated advantages of these models, there is a lack of comprehensive evaluation of these NeRF models for satellite image analysis. While each NeRF variant has been evaluated independently, often within a limited scope, there are several limitations. First, most existing performance studies have relied on relatively small datasets (usually having only four scenes), and have not typically considered the effects of physical scene properties, such as occlusion [3, 8]. Second, in these studies scene elevation evaluation has primarily relied upon predicted surface altitudes. Third, the previous studies rely upon datasets with restricted viewing angles inherent in satellite imagery which, when combined with the limited number of scenes explored in these studies, offer very few insights into model performance on datasets with similar challenges. Fourth, they have generally overlooked the impact of solar correction, which plays a critical role in modeling the shadows and ambience lighting. These limitations make the existing studies less generalizable. On the other hand, comprehensive quantitative and qualitative evaluation is indispensable for not only aiding the potential users but also for guiding future advancements of this domain.

Towards overcoming the above limitations, this work aims to provide a **holistic and detailed evaluation of existing NeRF implementations tailored for satellite imagery**. We enhance the evaluation dataset (**with 20 additional scenes**) and employ both quantitative and qualitative analyses to compare the performance of three state-of-the-art models, namely, *base NeRF*, *S-NeRF*, and *Sat-NeRF*.

This work has the following scientific contributions.

1. To the best of our knowledge, this is **the first study to provide a comparative analysis of NeRF models for 3D reconstruction from satellite images** within a broader, standardized, and uniform environment. We present a three-step pipeline for robust evaluation of 3D reconstruction algorithms that incorporates image quality and perceptual metrics like PSNR and SSIM and surface elevation comparison based on MAE.
2. Our **evaluation includes a larger and more diverse range of scenes**, providing insights into the performance of NeRF in scenarios not explored by previous studies.
3. Towards understanding model performance across various land cover types, we have developed a unique **hybrid evaluation framework that incorporates segmentation labels** to analyze the spatial, categorical and geometric distribution of the synthesized novel views and Digital Surface Models (DSMs).

The key findings from our study include the following. First, we noted that solar corrections significantly improve the accuracy of NeRF models across most extended evaluation datasets. Second, the original NeRF shows suboptimal results in reconstructing 3D models from satellite images, as indicated by performance metrics in both the training and evaluation stages. Third, S-NeRF and Sat-NeRF are more robust and thus better suited for scenarios involving the presence of shadow regions and transient objects.

## 2 Background

### 2.1 Three-Dimensional (3D) Scene Reconstruction

Reconstructing 3D scenes from 2D images is a key task in computer vision, with the aim of extracting spatial coordinates of 3D points from images taken at various angles. [9, 12–14]. Traditional techniques, *e.g.*, SfM with bundle adjustment [4, 10–12], are common yet face substantial challenges which include handling scenes with little to no texture, dealing with occlusions, resolving ambiguous image features, and accounting for transient objects, leading to sparse and inaccurate outputs. Further complexity arises from the need to accurately model scenes using various camera models, especially in satellite imagery where the RPC model is prevalent. This introduces numerous approximations, complicating the reconstruction process and affecting the fidelity of the resulting 3D models.

NeRF and its extensions have tried to address the limitations of traditional methods like SfM by representing scenes as continuous volumetric functions, resulting in denser representations. Each variant introduces different optimizations that further enhance the learned 3D scene representation. For example, Sat-NeRF utilizes RPC models to mitigate inaccuracies from camera model approximations while Shadow-NeRF aims to model the effect of shadows and ambient lighting.

Unlike most NeRF-related studies that typically focus on indoor or outdoor scenes captured by drones or commercial cameras, our study uniquely evaluates NeRF variants within the realm of satellite imagery. In the next subsections, we will provide the background of the NeRF variants examined in this work.

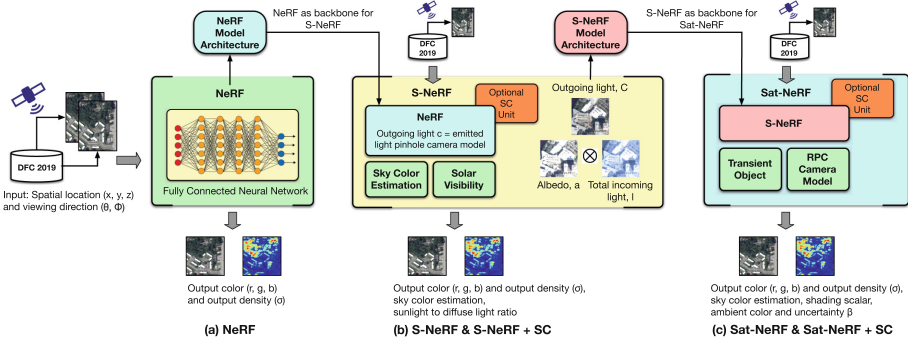
### 2.2 Brief Introduction of NeRF Models

**NeRF** (Fig. 1a) is a continuous volumetric function that predicts the RGB color  $c$  and the volume density  $\sigma$  of a scene at a given 3D point  $X$  (spatial location  $x, y, z$ ) from a viewing direction  $d$  ( $\theta, \phi$ ):

$$F : (X, d) \rightarrow (c, \sigma) \tag{1}$$

For training, NeRF uses images and their corresponding camera poses, projecting rays from the camera into the 3D space which intersect the image plane. Points along these rays are sampled to determine 3D coordinates in space, with





**Fig. 1. Overview of NeRF, S-NeRF, Sat-NeRF and their Solar Correction (SC) variants.** NeRF [9] is the base model that uses an MLP as its backbone. The basic NeRF architecture is modified in S-NeRF [3] to add sky color estimation, solar visibility, and optional SC for enhanced performance on satellite imagery. Sat-NeRF [8] unifies NeRF and S-NeRF, modifying the model to accommodate bundle-adjusted RPC camera parameters and transient objects.

each ray  $r$  described as  $r(t) = o + td$ , where  $o$  is the origin and  $d$  is the directional vector of the ray.  $t$  represents the points that are sampled along  $r(t)$ . The color  $c(r)$  of a ray is given by:

$$c(r) = \sum_{i=1}^N T_i \times \alpha_i \times c_i, \tag{2}$$

where  $T_i$ ,  $\alpha_i$ , and  $c_i$  denote the transmittance (probability of light reaching the point), opacity (probability of light being absorbed or scattered at the point), and color at the  $i_{th}$  point. Both  $\alpha$  and  $T$  are dependent on volume density  $\sigma$ . The color of a ray is computed by summing these weighted point-wise color values along the ray, with the model’s accuracy assessed by comparing predicted colors to actual image pixels.

$$\sum_{r \in R} \|c(r) - c_{gt}(r)\|_2^2 \tag{3}$$

NeRF has shown high quality reconstruction capabilities for simple indoor scenes and artificial data. However, instances where input images are scarce, have a lower range of viewing angles, belong to complex outdoor scenes lead to adverse effect in output quality. Furthermore, NeRF requires a high amount of computational resources and has no generalizability, requiring retraining for each new scene.

S-NeRF and Sat-NeRF are variants of NeRF, aimed to introduce schemes to deal with complex scenes and produce outputs close to its real-world counterpart.

**S-NeRF** incorporates additional elements into the base NeRF architecture, as depicted in Fig. 1b. This extension introduces a solar visibility layer and a sky color estimation layer and accepts a novel input, the solar direction  $\omega_s = (\theta_s, \phi_s)$ . S-NeRF, in turn, generates two new outputs, namely,  $s(x, \omega_s)$  and  $sky(\omega_s)$ .

The first output,  $s(x, \omega_s)$  quantifies the ratio between incoming solar light and diffuse skylight (solar light not directly from the sun), with values ranging from 0 (no solar visibility) to 1 (complete solar visibility). The second output,  $sky(\omega_s)$  serves as a learned estimator for the diffuse skylight. The total irradiance  $l$  (Eq.-(4)) is the weighted sum of the known white light source and learned light sources. The color of the point becomes a point-wise product of albedo  $c_a$  and irradiance  $l$  (Eq.-(5)).

$$l = s + (1 - s)sky \quad (4)$$

$$c(x, \omega_s) = C_a(x).l(X, \omega_s) \quad (5)$$

With NeRF’s pixel-based RGB sum squared error, S-NeRF adds a solar correction term and a  $L_1$  norm loss for solar absorption by visible surfaces. This results in photo-realistic novel views, as well as a better estimation of various illumination conditions, altitudes, and colors.

**Sat-NeRF** (Fig. 1c) further enhances Shadow-NeRF, tackling transient objects (*e.g.*, cars and vegetation) in input images. With the spatial coordinates  $x$  of a point and the direction of solar rays  $\omega$ , Sat-NeRF takes an additional input  $t_j$ , the learned transient embedding of image  $j$ .

The model adopts S-NeRF’s shadow irradiance approach and introduces a new output  $\beta$  to weigh the impact of transient objects (cars, vegetation) in the scene, based on the transient embedding  $t_j$ . As a result the Eq.-(1) becomes  $F : (x, \omega, t_j) \rightarrow (\sigma, c_a, s, a, \beta)$ .

### 2.3 Solar Correction

Prior observations [3] have highlighted that the shading scalar, when unconstrained, can lead to unrealistic results. To mitigate this issue, a solar correction approach is employed, which is formulated by

$$L_{SC} = \sum_{r \in R_{SC}} \left( \sum_{i=1}^{N_{SC}} (T_i - s_i)^2 + 1 - \sum_{i=1}^{N_{SC}} T_i \alpha_i s_i \right) \quad (6)$$

where  $T_i$  and  $\alpha_i$  are the transmittance and opacity of the  $i^{th}$  point or ray  $r$ . The initial part of the equation tries to keep the value of the shadow scalar as close to the transmittance, modeling the behavior of the shadow scalar similar to that of transmittance for different scene areas.  $R_{SC}$  denotes the secondary batch of rays used for solar correction. These rays follow the viewing direction of the solar rays ( $\omega$ ). The later part of the equation ensures that the shadow scalar always takes a value between 0 and 1 (when integrated over  $r$ ) to make sure albedo value  $c_a(x)$  remains the primary explanation for non-occluded regions. The solar correction  $L_{SC}$  and depth supervision  $L_{DS}$  terms are added to the loss to get the final multitask loss function:

$$L = L_{RGB}(\mathcal{R}) + \lambda_{SC}L_{SC}(\mathcal{R}_{SC}) + \lambda_{DS}L_{DS}(\mathcal{R}_{DS}) \quad (7)$$

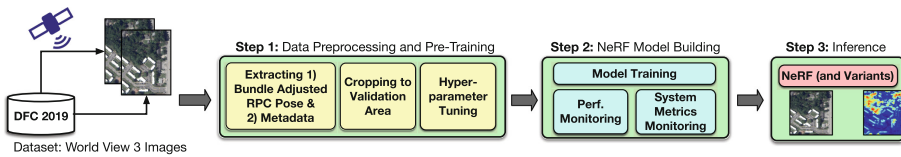
Sat-NeRF’s final loss function combines multiple terms to achieve accurate reconstructions of dynamic satellite imagery. The main components include:

- (1) **Photometric Loss:** This measures the pixel-wise difference between the predicted and actual color values, ensuring realistic rendering.
- (2) **Transient Object Loss:** Addresses moving objects in satellite scenes. It penalizes inconsistencies in object appearance across different views, encouraging the model to render them correctly.
- (3) **Shadow Loss:** Addresses the issue of shadows dynamically changing based on the solar position, guiding the model to learn and reproduce these shadows accurately.
- (4) **Depth Supervision (optional):** This term, inspired by DS-NeRF [2], can be added to reinforce training with additional depth-prior information.
- (5) **Solar Correction Loss:** Introduces the effect of additional solar rays to refine outputs even further.

By integrating these components, Sat-NeRF enhances its ability to render dynamic satellite scenes, capturing transient objects and generating realistic shadows with improved fidelity.

### 3 Evaluation Methodology

This section outlines the experiments and methods we used to analyze NeRF and its variations (S-NeRF and Sat-NeRF).



**Fig. 2.** Evaluation pipeline for NeRF models.

**Evaluation Pipeline.** Figure 2 illustrates the evaluation pipeline, which is composed of three steps: (1) data preprocessing and hyperparameter setup (2) model training, and (3) inference and evaluation.

**Step (1)** focuses on data preparation and setup before model training by processing the dataset for ensuring their compatibility with our models. Key procedures involve extracting refined RPC camera poses and metadata for each scene and applying image augmentations, *e.g.*, cropping, to fit the validation area requirements of our model configurations. Additionally, we fine-tune hyperparameters to optimize the models for training.

**Step (2)** is the NeRF (and its variants) model building. We train five different NeRF models across 24 scenes, evaluating their performance throughout the training using both quantitative metrics like PSNR, SSIM, and MAE.

**Step (3)** is to perform the quantitative analysis of the outputs from NeRF and its variants. We use image quality and geometric metrics. We generate and compare novel views produced by the trained models – views not seen during training – with the actual RGB images and DSM to assess the models’ accuracy.

**NeRF Models.** We evaluate five different NeRF models. The five models include base (original) NeRF model and two NeRF models tailored for the satellite imagery: Sat-NeRF and S-NeRF. Additionally, we also evaluate the solar correction (SC) version of both Sat-NeRF and S-NeRF as detailed in Sect. 2.3.

**Datasets.** One of our contributions is the expansion of datasets for NeRF evaluations. Previous research typically utilized only four areas of interest (AOIs) from the 2019 IEEE GRSS Data Fusion Contest [6]. However, we have successfully augmented our evaluation with an additional 20 AOIs. This substantial increase in datasets enables a more extensive assessment of NeRF models.

Our expanded dataset includes imagery from the Maxar WorldView-3 satellite, captured between 2014 and 2016, focusing on Jacksonville (JAX), Florida, USA. This broader collection of AOIs enhances our ability to rigorously test NeRF performance across a diverse array of scene characteristics. Each AOI is represented by 24 images, with 2–3 images randomly chosen for testing purposes. To maintain consistency, all images have been cropped to  $800 \times 800$  pixels, corresponding to a physical coverage area of approximately  $256 \times 256m$ , considering a GSD of  $0.3m$ . The data preprocessing phase also involves extracting crucial metadata for NeRF model inputs, *i.e.*, sun azimuth and camera parameters.

**Experimental Hardware and Training Configurations.** We perform our experiment on a server having two Intel Xeon CPUs with 26-core at 2.10 GHz, 256 GB of RAM, and two Nvidia RTX A6000 GPUs. Our SW environment is based on Ubuntu 20.04, with CUDA v11.8, and cuDNN v8.5. All the NeRF models are implemented using PyTorch v1.7.1 and PyTorch Lightning v1.3.7.

Each NeRF model employs an architecture of 8 fully connected layers, each containing 256 units. Training is conducted using the Adam optimizer [5], starting with an initial learning rate of  $5 \times 10^{-4}$ . We apply a learning rate decay strategy, utilizing a step scheduler that reduces the rate by a factor of 0.9 at each iteration. To optimize training duration and performance across numerous AOIs and model configurations, we do not preset the batch size and number of training steps. Instead, these parameters are determined through grid search, allowing for a more tailored and efficient training process.

**Performance Metrics.** We use three performance metrics for the quantitative analysis of NeRF models: (1) PSNR, (2) SSIM, and (3) MAE.

(1) **PSNR** (Peak Signal to Noise Ratio) measures the level of noise or distortion in an image by comparing it to a reference image (ground truth). PSNR is

expressed as  $\text{PSNR} = 20 \cdot \log_{10}(\text{MAX}/\sqrt{\text{MSE}})$ , where MAX indicates the maximum possible pixel value in the image (*e.g.*, 255 for 8-bit images), and MSE is the mean squared error between the original/reference image and the reconstructed/generated image. Higher PSNR values, indicating lower error, signal less distortion and better image quality, making high PSNR scores a goal for improved outcomes.

(2) **SSIM** (Structural Similarity Index) evaluates the structural similarity between two images, considering factors like luminance, contrast, and structure to quantify the preservation of the reference image’s structural integrity. Higher SSIM values denote a closer match to the reference image, implying better perceptual quality. SSIM is particularly valuable as it mirrors human visual perception, offering a measure of the perceptual accuracy of the generated images.

(3) **MAE** (Mean Absolute Error) calculates the average absolute difference between predicted and altitude values in ground truth. A lower MAE means the model’s altitude predictions are more accurate, reflecting greater precision in height estimation. This metric is essential for tasks in geospatial analysis and terrain modeling that demand precise elevation data.

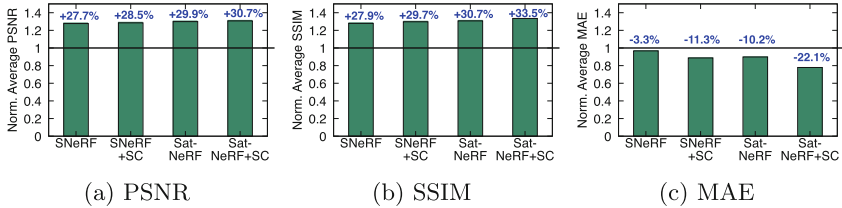
For the qualitative analysis, we **compare the predicted RGB and DSM outputs with the test images from our datasets**, providing a direct visual assessment of the model’s performance.

Additionally, we have developed a **hybrid evaluation framework** that incorporates both quantitative and qualitative analysis to better understand the performance differences among the NeRF variants. Our framework uses the segmentation labels provided with the dataset to better understand model performance across different land cover types. Specifically, the labels are used to segment both the ground truth and model-predicted RGBs and DSMs into ‘tree’ and ‘non-tree-covered urban’ regions. We then compare these segments based on PSNR, SSIM, and MAE.

## 4 Evaluation Results

We tested five different NeRF models on 24 different Worldview3 scenes. The evaluation results of PSNR, SSIM and MAE for all 24 scenes are shown in Table 1, 2 and 3. We also present the (normalized) performance comparison of all five model variants across all scenes over the base NeRF, as shown in Fig 3. The results show that the performance of satellite-tailored models was generally better than that of the base NeRF model across PSNR, SSIM, and MAE metrics for all scenes. Specifically, **the Sat-NeRF +SC model demonstrated 30.7% higher average PSNR values, 33.5% higher average SSIM, and 22.1% lower average MAE.** This trend can be attributed to the following findings:

- (1) PSNR-wise observations reveal that **model variants with SC outperformed other models in 71% of scenes** (Table 1). Specifically, the base NeRF and non-SC models performed better in only 7 out of 24 scenes. S-NeRF+SC and SatNeRF+SC models demonstrated significantly higher



**Fig. 3.** Normalized performance of all models over the base NeRF.

average PSNR values, with increases of 28.5% and 30.7%, respectively, compared to the base NeRF. Moreover, these models exhibited slightly higher PSNR values (0.58% to 0.60%) over their counterparts without SC. In a broader analysis, S-NeRF, Sat-NeRF, and their SC-enhanced versions showed commendable performance in 91% of the scenes. The base NeRF model only outperformed these advanced models in 2 out of the 24 scenes. Among these, Sat-NeRF+SC emerged as the top performer in terms of average normalized PSNR, achieving a value of 1.31, closely followed by Sat-NeRF at 1.29, with the base NeRF normalized to 1, as shown in Fig. 3a.

- (2) **In SSIM-wise observations, S-NeRF, Sat-NeRF, and their version with SC performed well in approximately 95.8% of scenes** (Table 2). Specifically, the base NeRF outperformed these models in just one out of 24 scenes. Furthermore, Sat-NeRF+SC demonstrated the highest average normalized SSIM, achieving a value of 1.34, followed closely by Sat-NeRF with a value of 1.30 over the base NeRF (Fig. 3b).
- (3) According to MAE, these four satellite-image-tailored variants performed well in 75% of scenes (Table 3) compared to the base NeRF. The base NeRF model only outperformed these models in 6 out of 24 scenes. Sat-NeRF+SC exhibits the lowest average normalized MAE with a value of 0.77, indicating accurate elevation details, followed by S-NeRF+SC with a value of 0.89 (Fig. 3c) compared to the base NeRF.

#### 4.1 Further Analysis with JAX-412 Dataset

We present scene **JAX-412** as a specific example of overall performance trends to describe the evaluation results further. Figure 5 reports the quantitative results, and Fig. 6 and 7 show the qualitative evaluation results. The followings are our observations.

**1. Base NeRF:** The baseline NeRF model showed the weakest performance with a decreasing validation PSNR and SSIM curve and increasing MAEs. The results indicate that the performance during the scene reconstruction suffered from a drop in image quality and an increased error in elevation details. However, our further analysis revealed that this base model is not well-adjusted for satellite imagery due to the following reasons.

**Table 1. Scene performance metrics (PSNR).** boldface: model wise best performance, shaded: scene wise best performance, \*: previously explored scenes

AOI (JAX)	NeRF	S-NeRF	S-NeRF w/ SC	Sat-NeRF	Sat-NeRF w/ SC
068*	9.75	21.3	21.15	21.06	21.94
004*	17.59	25.33	22.09	25.11	24.89
214*	16.68	22.46	23.79	23.12	23.60
260*	9.95	19.77	21.33	20.75	21.87
017	10.04	15.82	16.38	15.97	16.38
018	13.33	21.16	20.79	21.69	22
020	22.83	21.95	25.07	24.82	24.69
022	16.69	19.83	19.98	19.74	20.1
028	20.09	21.56	22.3	21.92	22.14
031	20.51	18.77	18.57	19.85	19.52
033	19.3	22.41	22.67	22.94	22.43
070	22.82	21.81	20.79	21.83	22.87
072	16.6	20.91	21.81	21.75	21.67
175	17.08	19.84	20.16	20.12	19.90
236	9.55	25.15	23.14	24.78	24.12
280	24.85	25.22	25.26	24.86	24.76
359	5.37	21.79	22.06	21.94	22.12
412	19.48	24.29	24.75	24.37	23.84
416	18.97	28.11	23.24	28.39	28.00
427	21.91	20.78	19.88	21.16	20.94
467	20.99	21.65	21.64	20.10	20.11
474	19.77	20.79	21.76	21.62	21.55
505	17.52	21.96	22.41	22.27	22.27
559	16.75	15.44	19.34	16.23	17.89

- (1) The base NeRF usually requires a **larger volume of training images per scene** than the dataset we used. While a larger training set aids scene estimation, the limited satellite imagery dataset presents challenges in scene reconstruction due to scarce scene details.
- (2) **Satellite datasets are limited in** the number and distribution of **viewing angles**. Off-nadir positions offer crucial details about vertical structures, aiding in 3D reconstruction [3]. However, our dataset primarily comprises viewing angles below  $35^\circ$  off-nadir. Consequently, information from wider off-nadir positions (*e.g.*, positions A, B, C, D in FigLNCSsubsubsection4) is unavailable, restricting the base NeRF’s reconstruction abilities.

**Table 2. Scene performance metrics (SSIM)**, boldface: model wise best performance, shaded: scene wise best performance, \*: previously explored scenes

AOI (JAX)	NeRF	S-NeRF	S-NeRF w/ SC	Sat-NeRF	Sat-NeRF w/ SC
068*	0.22	0.84	0.83	0.83	0.86
004*	0.47	0.86	0.72	0.86	0.86
214*	0.83	0.92	<b>0.94</b>	0.93	0.94
260*	0.08	0.76	0.85	0.81	0.84
017	0.14	0.26	0.27	0.25	0.29
018	0.38	0.74	0.71	0.75	0.76
020	0.86	0.78	0.89	0.88	0.88
022	0.60	0.77	0.79	0.78	0.80
028	0.70	0.76	0.79	0.78	0.79
031	0.81	0.79	0.80	0.81	0.82
033	0.81	0.8608	0.87	0.87	0.87
070	0.89	0.85	0.84	0.88	0.89
072	0.53	0.80	0.81	0.83	0.83
175	0.70	0.76	0.77	0.77	0.77
236	0.34	0.82	0.79	0.80	0.81
280	<b>0.90</b>	0.90	0.90	0.89	0.90
359	0.30	0.78	0.79	0.78	0.78
412	0.72	0.88	0.89	0.88	0.88
416	0.7615	<b>0.95</b>	0.88	<b>0.95</b>	<b>0.95</b>
427	0.82	0.78	0.72	0.80	0.80
467	0.83	0.84	0.83	0.83	0.93
474	0.68	0.73	0.76	0.77	0.76
505	0.63	0.85	0.86	0.85	0.86
559	0.68	0.56	0.75	0.62	0.74

- (3) Images captured at different timestamps exhibit significant non-correlation due to **varying lighting conditions**. Inconsistent lighting conditions can lead to erroneous results in NeRF, affecting the accurate representation of color and shadows within the scene [3]. This inconsistency may present a challenge for the model when estimating novel views.
- (4) The base NeRF employs a pinhole camera model, whereas satellites utilize **RPC camera models**. However, the pinhole camera model’s usage introduces approximations that may compromise prediction accuracy.
- (5) Real-world satellite images often have further complexities, *e.g.*, **transient objects** like cars and trees, which the base NeRF model does not effectively address.

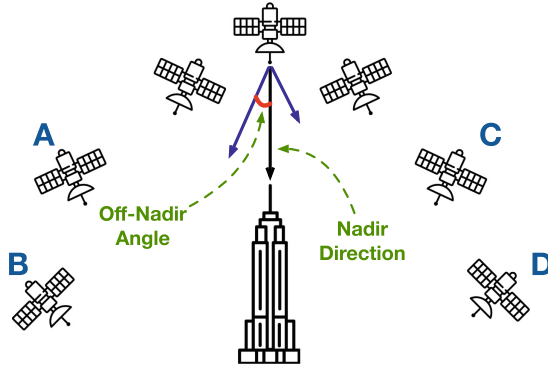


**Table 3. Scene performance metrics (MAE).** boldface: model wise best performance, shaded: scene wise best performance, \*: previously explored scenes

AOI (JAX)	NeRF	S-NeRF	S-NeRF w/ SC	Sat-NeRF	Sat-NeRF w/ SC
068*	2.74	3.11	2.18	3.30	1.89
004*	3.32	2.53	2.97	2.22	1.96
214*	4.65	7.14	4.98	6.17	4.85
260*	3.03	3.65	3.50	3.43	2.21
017	4.35	4.23	3.86	3.94	3.36
018	3.25	2.83	3.43	3.27	2.91
020	3.26	3.37	3.33	4.17	3.82
022	3.42	3.06	2.55	2.79	2.72
028	4.72	3.10	2.70	2.91	2.80
031	5.36	3.37	3.19	3.34	3.31
033	1.72	3.879	3.50	3.65	3.12
070	3.65	3.38	3.56	3.24	2.64
072	3.21	2.13	1.39	1.58	1.40
175	4.49	4.80	4.04	4.48	4.49
236	3.70	3.76	3.91	3.77	3.87
280	2.76	3.24	2.69	2.84	2.62
359	3.55	3.47	3.03	3.16	3.02
412	2.52	3.52	3.15	3.02	3.16
416	3.42	2.32	2.39	2.14	1.84
427	2.83	3.30	5.32	2.80	2.89
467	2.78	3.03	2.90	2.79	0.83
474	1.80	3.13	2.07	2.71	2.51
505	5.51	3.70	2.80	3.13	2.76
559	7.12	4.26	3.83	3.99	3.74

Qualitatively, while the base NeRF model’s color predictions were accurate, the base NeRF’s predicted output had inconsistencies like blurred reconstruction and lack of sharp edges (or lack of volumetric details), as shown in the region ‘A’ of Fig 7a. It also showed incomplete depth information in predicted DSM output, evident in the region ‘B’ of Fig 6a where large portions lack elevation detail.

**2. S-NeRF and S-NeRF with SC:** In contrast to the base NeRF evaluation, both S-NeRF and S-NeRF with SC showed positive trends, supported by increasing PSNR and SSIM scores shown in Fig. 5f, 5g, 5n, and 5o and decreasing MAE scores shown in Fig. 5b and 5p. On average, S-NeRF and S-NeRF+SC showed 27.7% to 28.5% higher PSNR values, 27.9% to 29.7% higher

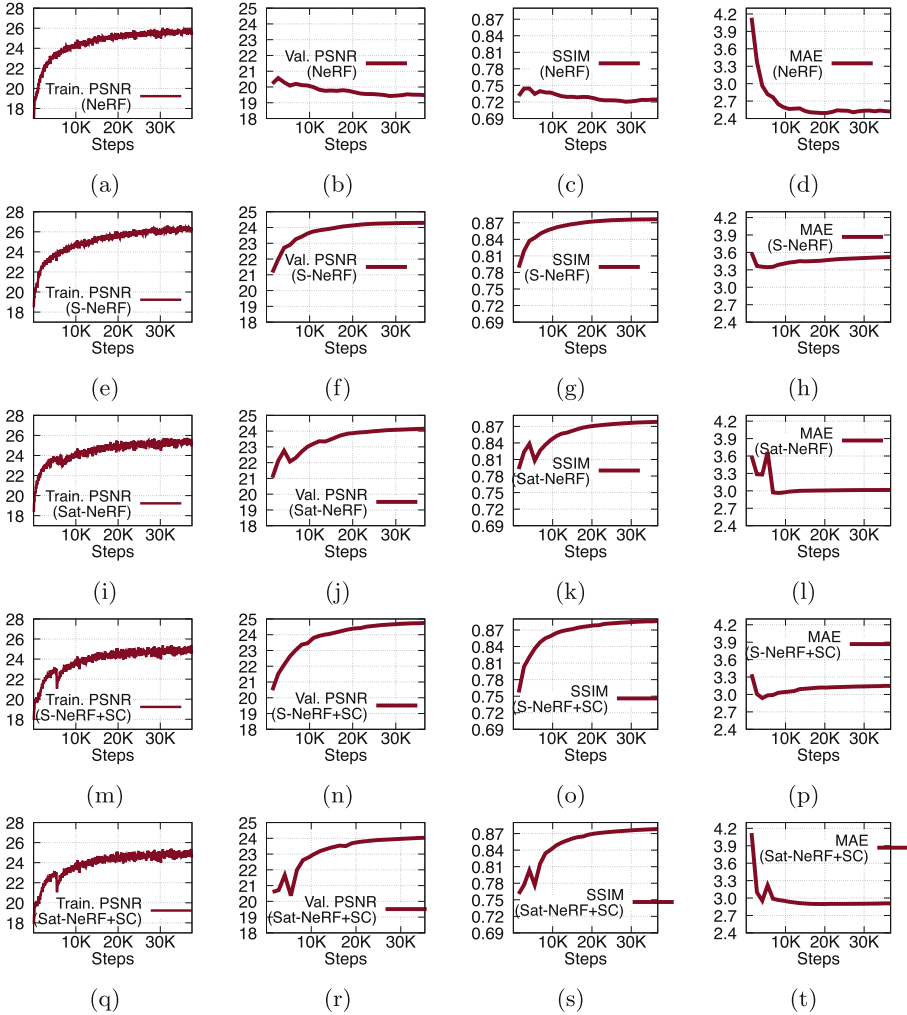


**Fig. 4.** Example of restricted view in the satellite imagery with off-nadir angle  $< 35^\circ$

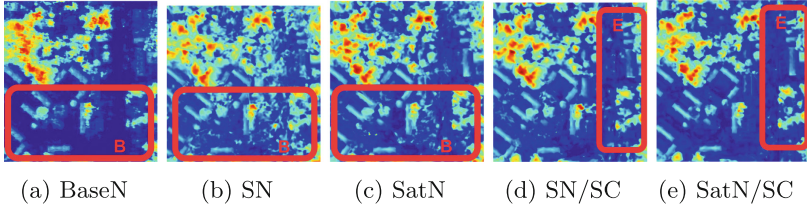
SSIM values and 3.3% to 11.3% lower MAE values compared to base NeRF. These metrics generally plateaued after 15K training steps, but extended training to 35K steps to stabilize the rendering quality. These models, designed to better handle shadows [3], outperformed the base NeRF in PSNR, SSIM, and MAE values. Their continuous improvement in image quality throughout training showed their potential as superior alternatives to the base NeRF.

Qualitatively, **S-NeRF** and **S-NeRF with SC** showed notable performance in capturing scene depth, although some areas still show blurriness as visible in the regions ‘C’ or ‘D’ in Fig. 6. These comparisons suggest that the SC variant excels over its predecessor, with S-NeRF showing improved depth in DSM outputs (Fig. 7b), particularly in highlighted regions like ‘C’. However, certain limitations persist due to the lack of bundle adjustment for refining camera parameters and an explicit mechanism for handling transient objects. These factors can compromise scene reconstruction and depth accuracy, especially with transient objects present in the scene. Additionally, both S-NeRF and its SC variant aim to accurately model shadows relative to scene geometry. This can potentially lead to challenges in scenes with minimal altitude variation or limited shadow coverage [3]. Consequently, they showed 1.11% to 2.28% lesser normalized average PSNR values, 0.78% to 4.23% lesser normalized average SSIM values and up to 24.17% higher MAE values compared to Sat-NeRF variants.

**3. Sat-NeRF and Sat-NeRF with SC:** Both Sat-NeRF and Sat-NeRF with SC also showed an increasing trend in PSNR, shown in Fig. 5j and 5. Additionally, the **Sat-NeRF with SC** consistently achieved higher SSIM values, outperforming other NeRF models (Fig. 5). The trend in MAE was decreasing, aligning with the trends observed in PSNR and SSIM (Fig. 5l and 5). They showed 29.9% to 30.7% higher normalized average PSNR values, 30.8% to 33.6% higher normalized average SSIM values and 10.15% to 20.09% lesser normalized average MAE values compared to base NeRF. They showed 1.12% to 2.33% higher normalized average PSNR values, 0.78% to 4.41% higher normalized average SSIM values and upto 19.46% lesser MAE values compared to



**Fig. 5.** Evaluation metrics of the five NeRF variants during training and validation. From (a) to (d) for NeRF results: (a) Training PSNR, (b) Validation PSNR, (c) Validation SSIM, and (d): validation MAE. From (e) to (h) for S-NeRF results: (e) Training PSNR, (f) Validation PSNR, (g) Validation SSIM, and (h): validation MAE. From (i) to (l) for Sat-NeRF results: (i) Training PSNR, (j) Validation PSNR, (k) Validation SSIM, and (l): validation MAE. From (m) to (p) for S-NeRF+SC results: (m) Training PSNR, (n) Validation PSNR, (o) Validation SSIM, and (p): validation MAE. From (q) to (t) for Sat-NeRF+SC results: (q) Training PSNR, (r) Validation PSNR, (s) Validation SSIM, and (t): validation MAE.



**Fig. 6. DSM outputs from all NeRF variants.** BaseN: Base NeRF, SN: S-NeRF, SatN: Sat-NeRF, SC: Solar Correction



**Fig. 7. RGB output from all NeRF variants.** BaseN: Base NeRF, SN: S-NeRF, SatN: Sat-NeRF, SC: Solar Correction

S-NeRF variants. We also observed an initial spike in PSNR, SSIM, and MAE for Sat-NeRF models, followed by stabilization. This behavior aligns with the delayed utilization of the uncertainty coefficient starting from epoch 2 [8].

In the qualitative evaluation, Sat-NeRF provides enhanced clarity in RGB prediction (region ‘D’ in Fig. 6 and 7). Moreover, the Sat-NeRF with SC showed superior capability in handling transient objects (region ‘D’ of Fig. 7), along with offering more detailed object visuals and depth estimates (region ‘E’ of Fig. 7). The improved performance of both Sat-NeRF and Sat-NeRF with SC can be attributed to three key factors:

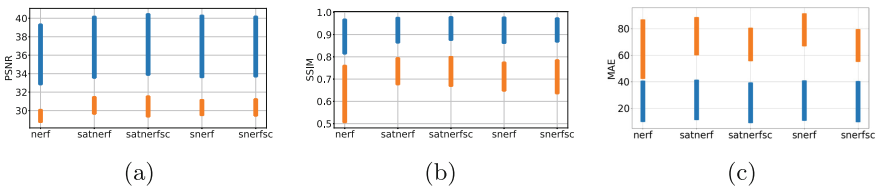
- (1) **Both Sat-NeRF and Sat-NeRF with SC utilize RPC-based sampling**, eliminating the need to model specific satellite features or pinhole camera approximations. As a result, this method consistently delivers superior results across various scenarios.
- (2) **Refining the RPC camera via bundle adjustment helps** reduce the chance of projecting non-coincident image points, as detailed in [8].
- (3) **Effectively handling transient objects** improves the model’s performance, especially in accurately estimating depth, as shown in region ‘E’ of Fig 7.

## 4.2 Land-Cover Study Using JAX-412 and JAX-260

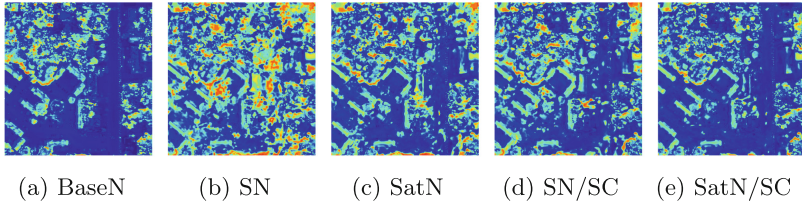
During further exploration of the models, we have studied the performance variation of the models based on land cover types. Here, we use JAX-412 and JAX-260 as examples of different land-cover types for our Land-cover study.

We observed that **tree regions exhibit higher performance in terms of PSNR, SSIM, and MAE, indicating superior image quality and DSM elevation accuracy for all the scenes.** All five models perform better on tree regions across all scenes—20–22% based on mean PSNR and 25–40% better based on mean SSIM. Non-tree urban areas are prone to error, while water-bodies were identified as weak areas with localized errors. These are likely due to varying reflections at different timestamps in the training set and a potential lack of texture. Rooftops and directly illuminated white surfaces demonstrated minimal image quality errors but exhibited significant DSM inaccuracies, highlighting inaccuracy in depth reconstruction. Additionally, shadow areas were found to be prone to errors, further impacting performance metrics. The general trend can be viewed in Fig. 8a, 8b, and 8c. Moreover, the specific examples of weak-area (water-body, buildings, shadows) can be verified from the error maps in Fig. 9 and 11 (warmer shades are erroneous).

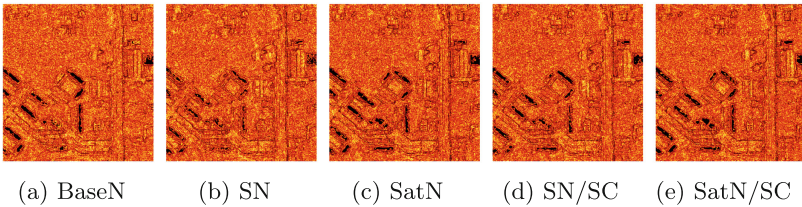
Based on the texture maps and error maps, both JAX-412 and JAX-260 conform to the general trends of better performing tree regions and error prone urban areas as depicted in Fig. 9 and 10. The texture comparison maps in Fig. 10 show a unified error distribution across the urban area of the scene, except for white regions which are directly illuminated by sunlight and have minimal local patterns (e.g., rooftops), which have lower level of texture-based errors. However, texture comparison map shows water-bodies having a higher texture error compared to the rest of the scene. The error map between DSMs show a much better comparison across the different models. According to DSM error maps, the rooftops are localized sources of errors. Moreover, water-bodies and shadow areas are also weakly reconstructed as depicted in Fig. 12 and 11. The performance of the solar correction models are visibly better compared to the non solar correction variants for both JAX-412 and JAX-260 as depicted in Fig. 11 and 9. The Solar Corrective variants have comparable performances. S-NeRF-SC is approximately 4.20% higher than base S-NeRF for the masked (tree) region. This behavior confirms the superiority of solar correction models.



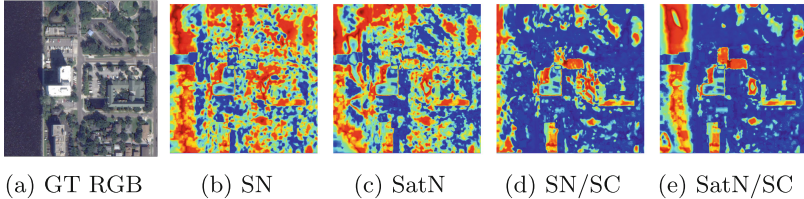
**Fig. 8. Comparative Performance Analysis of Models on tree and Non-tree urban Regions based on PSNR, SSIM, and MAE** masked (blue) = ‘tree’ region, inverse (orange) = ‘non-tree region’. (Color figure online)



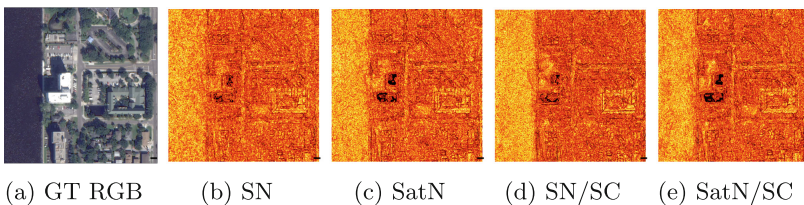
**Fig. 9. Altitude error map between ground truth DSM and predicted DSM** BaseN: Base NeRF, SN: S-NeRF, SatN: Sat-NeRF, SC: Solar Correction (warmer shades: more erroneous region)



**Fig. 10. Texture comparison map between ground truth DSM and predicted DSM** BaseN: Base NeRF, SN: S-NeRF, SatN: Sat-NeRF, SC: Solar Correction (Lighter shades: more erroneous region)



**Fig. 11. Error map between GT and DSMs showing waterbodies, rooftops and shadows to be the source of localized error** GT RGB: Ground Truth RGB, SN: S-NeRF, SatN: Sat-NeRF, SC: Solar Correction



**Fig. 12. Error map between GT and texture comparison maps showing waterbodies as the source of localized texture error (Lighter shade implies higher error)** GT RGB: Ground Truth RGB, SN: S-NeRF, SatN: Sat-NeRF, SC: Solar Correction

### 4.3 Evaluation Summary

Here is our summary of the comprehensive evaluation conducted with five NeRF variants across 24 expanded satellite datasets.

- (1) Solar correction (SC) can significantly enhance the performance of both S-NeRF and Sat-NeRF across quantitative and qualitative metrics.
- (2) Sat-NeRF with SC demonstrates superior performance compared to S-NeRF and its SC variant in scenes featuring frequent transient objects such as cars. Sat-NeRF can generate more accurate representations.
- (3) S-NeRF with SC can be the best option for scenes characterized by changing lighting conditions but lacking moving or transient objects. This is because S-NeRF with SC can ensure smooth handling and consistent scene rendering.
- (4) Sat-NeRF and its SC variant are better suited for scenes with minimal altitude changes and transient objects.
- (5) Based on our land cover study, the S-NeRF, Sat-NeRF and their variants perform better in tree regions, with lower errors and higher image quality compared to urban regions.

## 5 Conclusion

In this study, we conducted a thorough evaluation of NeRF and its variants for generating 3D views from satellite imagery, with a particular focus on their suitability and effectiveness in handling satellite data. Our assessment encompassed NeRF, Sat-NeRF, S-NeRF, and their variations incorporating SC, leveraging extensive DFC2019 datasets comprising 24 diverse scenes. Both quantitative and qualitative evaluation methods were employed to scrutinize their performance. In summary, our evaluation of NeRF and its variants, including Sat-NeRF, S-NeRF, and their solar correction variations, across 24 distinct urban scenes derived from satellite imagery, unveiled their strengths and limitations. The SC variants demonstrated superior performance compared to other NeRF models, while S-NeRF and Sat-NeRF showcased notable proficiency, especially in handling scenes characterized by shadows and transient objects.

**Acknowledgements..** This research was sponsored by the United States Army Corps of Engineers (USACE) Engineer Research and Development Center (ERDC) Geospatial Research Laboratory (GRL) and was accomplished under Cooperative Agreement Federal Award Identification Numbers W9132V-22-2-0001 and W9132V-23-2-0002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of USACE ERDC GRL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.



## References

1. Chen, H., Chen, W., Gao, T.: Ground 3D object reconstruction based on multi-view 3D occupancy network using satellite remote sensing image. In: IGARSS (2021)
2. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised NeRF: fewer views and faster training for free. In: CVPR (2022)
3. Derksen, D., Izzo, D.: Shadow neural radiance fields for multi-view satellite photogrammetry. In: CVPR (2021)
4. Enqvist, O., Kahl, F., Olsson, C.: Non-sequential structure from motion. In: ICCV Workshops (2011)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2017)
6. Le Saux, B., Yokoya, N., Hänsch, R., Brown, M.: Data fusion contest 2019 (DFC2019) (2019). <https://doi.org/10.21227/c6tm-vw12>
7. Leotta, M.J., et al.: Urban semantic 3D reconstruction from multiview satellite imagery. In: CVPR Workshops (2019)
8. Mari, R., Facciolo, G., Ehret, T.: Sat-NeRF: learning multi-view satellite photogrammetry with transient objects and shadow modeling using RPC cameras. In: CVPR Workshops (2022)
9. 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)
10. Pierrot Deseilligny, M., Clery, I.: APERO, an open source bundle adjustment software for automatic calibration and operation of set of images. *Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci.* **38**, 269–276 (2012)
11. Rupnik, E., Daakir, M., Pierrot Deseilligny, M.: MicMac - a free, open-source solution for photogrammetry. *Open Geospat. Data Softw. Stand.* **2**(1), 14 (2017)
12. Wei, X., Zhang, Y., Li, Z., Fu, Y., Xue, X.: DeepSFM: structure from motion via deep bundle adjustment. In: ECCV (2020)
13. Xie, H., Yao, H., Sun, X., Zhou, S., Zhang, S.: Pix2Vox: context-aware 3D reconstruction from single and multi-view images. In: ICCV (2019)
14. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: MVSNet: depth inference for unstructured multi-view stereo. In: ECCV (2018)





# Skeletal Triangulation for 3D Human Pose Estimation

YiHeng Jiang<sup>1</sup> , ZhiPeng Wang<sup>1</sup>, YunLong Zhao<sup>1</sup> , Yang Li<sup>2,3</sup>,  
and ChunYan Liu<sup>1</sup>

<sup>1</sup> College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

[zhaoyunlong@nuaa.edu.cn](mailto:zhaoyunlong@nuaa.edu.cn)

<sup>2</sup> Unmanned Aerial Vehicles Research Institute, Nanjing University of Aeronautics and Astronautics, Nanjing, China

<sup>3</sup> Key Laboratory of Advanced Technology for Small and Medium-sized UAV, Ministry of Industry and Information Technology, Nanjing, China

<http://cs.nuaa.edu.cn>

**Abstract.** Nowadays, many researchers have made significant progress in the field of multi-view 3D human pose estimation. However, numerous multi-view human pose estimation models based on deep learning heavily rely on data-driven training. As a 3D reconstruction method based on mathematical modeling, triangulation has shown excellent generalization ability and is widely used for 3D pose estimation and 3D pose annotation tasks in unlabeled environments. In this paper, we propose a refinement module based on graph convolution and visual fusion, and based on this, propose a triangulation-based method infused with structural information, Skeletal Algebraic Triangulation (SAT), encoding human pose prior knowledge into the model to ensure its robustness under occlusion and complex motions. Experiments show that our model outperforms algebraic methods and achieves comparable performance to state-of-the-art methods. Meanwhile, our method has better generalization performance, showing better and more robust results on different view Settings from the training dataset. Besides that, the proposed method can be applied to different backbone networks. As a core part of SAT, the graph refinement model can also be used to improve existing keypoint estimation. The volume triangulation combined with the graph refinement module, called Skeleton Volume triangulation, achieves state-of-the-art performance on the Human36M. In addition, the graph refinement module is also used for other keypoint estimation tasks that contain structural information, such as hand landmarks.

**Keywords:** 3D human pose estimation · Multi-view Image · Learnable Triangulation · Skeletal Structure · Graph Refine · GCN

---

Supported by National Science and Technology Major Project (2022ZD0115403).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15318, pp. 180–196, 2025.

[https://doi.org/10.1007/978-3-031-78456-9\\_12](https://doi.org/10.1007/978-3-031-78456-9_12)

## 1 Introduction

3D human pose estimation is a research focus in the field of computer vision, which aims to infer 3D joint coordinates from multi-view images and related information. Despite advancements in multi-view pose estimation algorithms, as seen in [28] and [35], many methods heavily rely on high-quality training data and may struggle to adapt to novel camera configurations. Acquiring ample labeled images for new scenarios is essential for achieving satisfactory performance.

A significant challenge lies in obtaining accurate 3D pose annotations, particularly in the wild, which predominantly rely on triangulation [12, 16]. Triangulation leverages mathematical modeling to compute spatial geometric relationships among keypoints across multiple views, thereby yielding 3D keypoints. Due to its mathematical underpinnings, this approach boasts strong generalization capabilities, applicable wherever image data and camera calibration parameters can be obtained. However, its reliance on mathematical methods renders it sensitive to imprecise 2D pose estimates. Humans can effortlessly discern postures in space, easily identifying incorrect pose estimates, thanks to their rich spatial contextual and biological structure priors, which play a crucial role in enhancing accuracy. Our contributions are summarized as follows:

- (1) We have investigated methods for encoding structural information into pose estimation, with experimental validation substantiating the significance of such information in enhancing pose estimation accuracy.
- (2) We propose a refinement module based on graph convolution and visual fusion, and based on this, propose a triangulation-based method infused with structural information, Skeletal Algebraic Triangulation (SAT), An approach refining 2D joint localizations across views using both visual features and skeletal prior, followed by multi-view 3D skeletal refinement after initial triangulation.

Experiments show that our framework is compatible with mainstream 2D skeletons, significantly improves pose estimation performance, and outperforms mainstream algebraic methods. Especially in the case of significant deviations in 2D pose. In addition, the Advanced volume method combined with our refinement module called Skeletal Volume triangulation, achieves state-of-the-art results.

## 2 Related Works

In the realm of human pose estimation, research is bifurcated into two primary categories: monocular and multi-view estimation [30]. While significant advancements have been made in monocular pose estimation [9, 18, 19, 23, 27, 36], Directly obtaining 3D human pose from a single 2D image is still an ill-posed problem. This is due to the ambiguity caused by the existence of multiple 3D poses with the same 2D projections. Incorporating multiple-view images alleviates these

challenges, enhancing the precision of 3D pose recovery. Works focusing on single-person scenarios have successfully exploited multi-view geometry [11], learnable triangulation techniques [16], and graphical models [20, 24], with additional inquiries into the generalizability of learned triangulation [1]. For more complex multi-person settings, matching and triangulation-based algorithms [6, 7] predict 2D keypoints followed by feature matching and application of multi-view geometry [11, 12] for 3D joint coordinate extraction. Volumetric methods [28] segment the 3D space into uniform grids and utilize probabilistic models alongside 3D CNNs for keypoint detection. Two-stage, top-down graph convolutional networks [32] tailor GNN modules based on dynamic graph convolutions [22] for specific tasks. Leveraging Transformers’ [29] powerful attention mechanisms and their profound impact in computer vision [8], single-stage approaches [35] directly predict 3D keypoints for multiple individuals. Additionally, monocular video-based methods [21, 34, 37] are a thriving area of interest, leveraging temporal information to resolve ambiguities distinct from spatial cues, offering wider applicability but often at the expense of accuracy compared to multi-view strategies. Notably, existing learning-based methods for human pose estimation are heavily reliant on the quality and coverage of training data, necessitating extensive and high-quality datasets for achieving satisfactory predictive performance. Datasets such as shelf/campus [2], Panoptic [10], Human3.6M [15], HumMAN [3], and Freeman [31] have been instrumental in providing real-world annotations, indoor/outdoor scenarios, and depth information, pushing the boundaries of pose estimation under varying conditions. These datasets commonly employ motion capture [15] or triangulation methods [3, 31], capitalizing on their generalization capabilities, although the accuracy is contingent on meticulous camera calibration and precise 2D detections. Improved triangulation algorithms [16, 24] refine these techniques for enhanced precision. Humans innately understand complex poses and occlusions, leveraging priors about body structure, spatial geometry, and multi-view integration. The inherent graph structure of human pose can enhance estimation accuracy, as seen in works addressing occlusions in 2D estimation [25] and 3D regression [5, 36], and more prominently in multi-person multi-view estimation [32]. Structured triangulation [4], assuming known bone lengths, further incorporates the natural structural relationships and functional correlations between body parts into optimization frameworks.

### 3 Skeletal Triangulation

#### 3.1 Overview of Method

Algebraic triangulation transforms the 2D pose representation into 3D joint coordinates in space using the principles of multi-view geometry and spatial geometry. This method boasts excellent generalization capabilities due to its reliance on such underlying mathematical principles. However, the inherent nature of this algebraic approach places a high premium on the precision of 2D joint detections. The utilization of confidence scores associated with 2D keypoints as weights in the 3D reconstruction process [16] can significantly ameliorate, yet reliance solely

on this mitigation strategy still yields inaccuracies, as considerable amounts of visual information are lost during the abstraction of 2D keypoints and their corresponding confidences.

We propose a structural information embedding refinement module, and based on that, propose a new learnable triangulation model, Skeletal Algebraic Triangulation (SAT), which harnesses the human skeleton structure prior information encoded into the graph structure while concurrently utilizing visual features to rectify errors, thereby mitigating the detrimental influence of erroneous intermediate outcomes on the overall estimation. The architecture of SAT is depicted in Fig. 1.

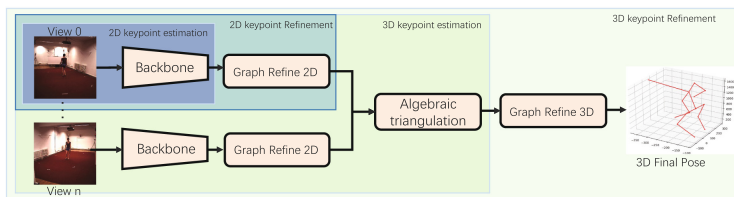
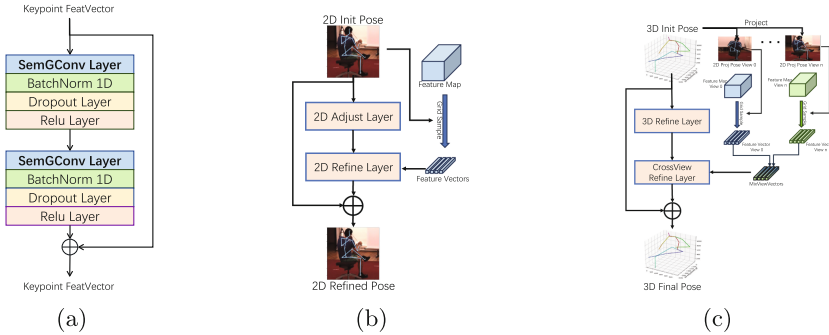


Fig. 1. The Overview of SAT

### 3.2 Graph Refinement Module

Complex poses and occlusions have long posed significant challenges for human pose estimation. The errors caused by these factors seriously affect the accuracy of pose estimation. Humans, with their extensive prior knowledge including multi-view geometry, spatial arrangements, and innate understanding of bodily structures, excel at recognizing faulty poses. This knowledge plays a decisive role in mitigating errors resulting from varying circumstances. Extensive prior research [25, 32] has validated the profound impact of such priors on enhancing pose estimation performance. In light of this, we introduce a graph refinement module for pose correction. We construct a human pose graph  $G = \{V, E\}$  to model the skeletal structure, where  $V = \{v_i \mid i=0,1,\dots,n\}$  is a set of vertices representing  $n$  keypoints in the skeleton, and  $E = \{v_i v_j \mid \text{if } v_i \text{ and } v_j \text{ are connected in the skeleton}\}$  represents the edges symbolizing body limbs. The adjacency matrix  $A = \{a_{ij}\}$  of graph  $G$  is defined such that  $a_{ij} = 1$  if vertices  $v_i$  and  $v_j$  are connected, and 0 otherwise. The graph neural network employs a network architecture akin to [25, 36] called SemanticGCN, with a layer schematic depicted in Fig. 2 (a). Each key point will be passed through the multilayer perceptron to obtain its feature vector. The feature vector is then passed through the semantic graph convolutional layer, batch normalization layer, Dropout layer and RELU activation layer, and added to the initial input to obtain the refined feature vector. The graph refinement module is divided into 2D refinement module (Figure 2 (b)) and 3D refinement module (Figure 2 (c)). The 2D refinement module uses

skeletal prior encoded in the graph with visual features at locations of keypoints to adjust incorrect predictions. The 3D refinement module utilizes skeletal priors and integrates visual features from multiple views while learning their dependencies. This strategy further improves the accuracy of pose estimation.



**Fig. 2.** shows the architecture of Graph Refinement modules, (a) shows the architecture of Graph Refinement layer, (b) shows the architecture of 2D Refinement Module, and (c) shows the architecture of 3D Refinement Module.

**2D Refinement Module.** This model is a two-layer semantic graph convolutional network, similar to [25, 36]. The first layer processes only 2D keypoints. The input is the feature vector transformed by the keypoints, integrating information through pre-established skeletal prior. It employs a skeletal structure to rectify 2D keypoints, correcting 2D pose estimation biases induced by occlusions or complex actions. Then, the visual features of each keypoint and the keypoint features are concatenated and input to the next graph convolution layer. This layer refines based on keypoints and their corresponding visual features, guided by prior knowledge of human skeleton structure, yielding a more precise 2D pose, as illustrated in Fig. 2(a).

**3D Refinement Module.** The architecture of 3D refinement Module is similar to its 2D counterpart, as illustrated in Fig. 2(b). 3D coordinates derived from triangulation are fed into a graph neural network layer for refinement with structural information, yielding refined 3D coordinates. These coordinates are then projected onto individual views, extracting 2D visual features at corresponding locations in each view’s feature map. Simple visual information concatenation proves insufficient for multi-view 3D reconstruction and fails to effectively integrate features from diverse perspectives. It also cannot accommodate different numbers of views. Inspired by [22, 32], we design a multi-view fusion method based on a dynamic graph structure. The mathematical expression of fusing multiple view visual feature vectors is as Eq. 1:

$$x_v = \max(\text{FC}(\text{concat}(x_v, x_v - x_n))) \quad (1)$$

In a triangulation process involving  $n$  views, visual feature vectors from each camera  $v$  are concatenated with differences from  $n - 1$  remaining feature vectors.  $x_v$  represents the visual feature at the projected point in  $v$ 's feature map for node  $x$ , while  $x_n$  symbolizes features from all other views. MLP neural network,  $FC$ , is employed.  $n - 1$  concatenated feature vectors traverse a fully connected layer, receiving learnable weights, and are max pooled to form a multi-view fused feature vector encapsulating multi-view information. This process is applied to all  $n$  visual feature vectors from each perspective, yielding  $n$  feature vectors with integrated information. These vectors undergo a fully connected layer, multiplication by learnable weights, and max pooling, ultimately merging into a single visual feature vector.

Influenced by the concept of residual networks introduced in [13], the offset used to refine the attitude coordinates is output instead of the final coordinates directly, similar to [25]. The offset is then added to the original input coordinates to obtain the final refined coordinates. This approach of outputting corrections fosters stability during model training, while the optimization module demonstrates strong versatility, as both the 2D and 3D refinement models can be independently applied to various backbones.

### 3.3 Architecture of SAT

The overall architecture of SAT is depicted in Fig. 3. The 2D keypoints and their visual feature maps are extracted from each view by the backbone. These keypoints are then input into the 2D refinement model, where they are adjusted in conjunction with visual feature information, resulting in corrected 2D keypoint values. After refinement, the corrected keypoints undergo a triangulation algorithm, generating preliminary 3D joint coordinates. These 3D keypoints are further input into the 3D refinement model, where they are combined with visually fused features from each perspective for targeted fine-tuning, ultimately yielding 3D keypoint correction values and precise 3D joint keypoint locations.

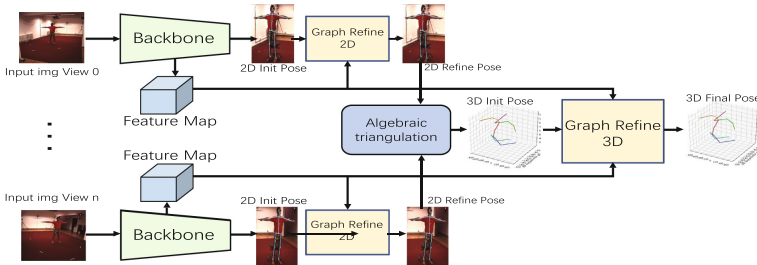


Fig. 3. The architecture of SAT.

During training, The gradients are backwarded to the corresponding 2D refinements and backbone for each view, ensuring an efficient exchange of information across views. Crucially, our model permits joint training with the 2D

backbone network, the network can also learn the dependence between different views in the multi-view joint training, which can better adapt to different situations.

Our graph refinement module can also adapt existing methods to further improve their metrics. For example, we combine 3DRM with end-to-end method volume triangulation [16], which is called Skeletal Volume Triangulation.

## 4 Experiments

### 4.1 Experiments Setting

We mainly used the Human36M [15] for evaluation, and we also used the CMU Panoptic [10] for experiments. The Human3.6M dataset, widely used in 3D human pose estimation research, was released by Ionescu et al., as the largest single-person 3D pose estimation dataset. The CMU Panoptic dataset contains multiple cameras that capture the social activities of multiple people and is the largest dataset for multi-person pose estimation. For Human36M, Adhering to the experimental approach of Isakov et al. [16]. However, the difference is that we opted to work directly with the Human3.6M dataset’s raw images instead of applying image undistortion. For CMU Panoptic, the HD images are resized to a quarter and the single-person sequences 170915 office1, 171026 cello3, 171026 pose3, 171204 pose3, 161029 car1 are selected as the training set, and 16102 tools1, 170407 office2 are selected as the test set. At the same time, the frames in which the subject cannot be seen in all views are removed. Our experiments were conducted on an Nvidia GeForce RTX3090 GPU (24GB memory) using PyTorch 1.13.1 and CUDA-12.3. The backbone [33] and the baseline method were fine-tuned on the Human3.6M dataset using the ResNet152 weight file provided by Isakov et al. [16]. For systematic comparison. We also employed PoseResNet50 initially trained on the Panoptic dataset [10] following the VoxelPose [28] framework and fine-tuned on Human36M. The metrics are mainly Mean Per Joint Position Error(MPJPE), which quantifies the absolute error between prediction and Groundtruth, and reflects the absolute accuracy assessment of the model in the world coordinate system. All methods are based on the ResNet152 backbone unless otherwise stated.

### 4.2 Evaluation of Graph Refinement Module

We investigated the core of SAT, the Graph Refinement Module (GRM). Table 1 shows the experimental results, contrasting the refinement capabilities of the 2D Refinement Module(2DRM) and 3D Refinement Module(3DRM). For the evaluation of the two GRMs, we directly used the L2Loss on the test set as the metric. SAT integrates algebraic triangulation with both 2DRM and 3DRM modules. We performed ablation experiments using only 2DRM to refine the 2D pose(-G2 suffix in the table) and only 3DRM to refine the 3D pose(-G3) compared with SAT(-G23). Table 2 shows the experimental results. The Mean

Per Joint Position Error (MPJPE) was used as the metric. Considering the GRM as a generic refinement model compatible with diverse frameworks. In order to verify the generality of GRM, we also integrated the 3DRM component with VolT. By refining the initial pose estimates yielded from this method through incorporation with Structure prior and visual features, referred to as VolT-G3 or Skeletal Volume Triangulation (SVT) in Table 2. We also investigate the role of graph convolution and skeleton prior to GRM. The keypoint connections of GRM were replaced with different structures separately and compared with SAT, using a fully connected graph (or complete graph, -FCG suffix in the table), where each keypoint connects all other points and does not represent an interpretable human skeleton prior, and using only MLP (we did not change the network structure of 2DRM and 3DRM, it just turns the body structure graph into an empty graph. Each node is independent and no two points are connected, -MLP suffix in the table).

**Table 1.** Evaluation of GRMs

	L2-Before	L2-After	Improve
2DRM	0.20	0.19	5.00%
3DRM	1.92	1.70	11.46%

**Table 2.** Effectence of GRMs

	MPJPE-Before	MPJPE-After	Improve
AlgT-G2	20.89	19.12	1.78 (8.00%)
AlgT-G3	20.89	18.66	2.23 (10.71%)
AlgT-G23(SAT)	20.89	18.65	2.24 (10.73%)
VolT-G3(SVT)	17.62	16.71	0.68 (5.16%)
AlgT-FCG	20.89	19.49	1.40(6.72%)
AlgT-MLP	20.89	19.46	1.43(6.85%)
AlgT-G23(SAT)	20.89	18.65	2.24 (10.73%)

It is evident from the results that both the 2DRM and 3DRM modules help to reduce the error between keypoints and Groundtruth. Note that the enhancement provided by using only 3DRM is better than using only 2DRM. This is because 2D refinement focuses on refining 2D pose, whereas accuracy in a single view may not necessarily improve 3D pose estimation accuracy. In contrast, 3DRM, which combines pose structure with multi-view visual information, exhibits stronger 3D pose accuracy. Integrating 3DRM with the end-to-end method volumetric triangulation also improves accuracy. Graph Refinement modules exploit interpretable prior information to further refine the results. This confirms the generality of graph refinement models that can be adapted to different frameworks and improve their performance capabilities. Based on studies on the effectiveness of graph convolutions (bottom half of Table 7), it is evident that the fusion of visual information leads to improved performance. We note that employing a fully-connected graph yields inferior performance compared to both the method only MLPs and SAT. The redundancy in connections within the fully-connected graph introduces unnecessary interference, detracting from accuracy. Relying exclusively on MLPs, which lack the integration of skeletal context and instead adjust keypointsbased solely on visual features, also falls short



of the performance achieved by SAT. This highlights the crucial importance of combining human anatomical priors with visual information for enhancing the precision of pose estimation.

### 4.3 Performance Comparison

**Comparison with Baseline.** We first compare the performance of SAT with its baselines. GRM is a general module, and SAT based on GRM can be adapted to different backbones. To verify compatibility and performance with different backbones, PoseResNet50 was adopted, initially set up on VoxelPose [28] and trained on the Panoptic dataset [10]. This was subsequently fine-tuned on H36M. The -50 suffix in the table represents that the backbone is replaced with ResNet50. We also evaluate on CMU Panoptic dataset. Similar to the configuration on the H36M, all models were tested after re-fine-tuning in Panopitc. Tables 3 and 4 illustrate the results.

**Table 3.** Experiments on the H36M

Human3.6M	MPJPE
AlgT-50	24.38
AlgT	20.89
SAT-50(ours)	<b>20.26</b>
SAT(ours)	<b>18.65</b>

**Table 4.** Experiments on the Panoptic

CMU-Panoptic	MPJPE
AlgT-50	29.59
AlgT	28.31
SAT-50(ours)	<b>28.17</b>
SAT(ours)	<b>27.31</b>

Our method SAT achieves better results with different backbones. When the backbone is replaced with ResNet50, The parameter count nearly halved in the backbone, leading to a decline in performance across all metrics. Despite these constraints, our model adjustments still yield improved outcomes. It is noteworthy that our SAT employing ResNet50 as the backbone outperforms the AlgT model utilizing ResNet152. Furthermore, considering the computational overhead in light of the performance contrasts presented in Table 6 and 7, SAT with ResNet50 backbone demonstrates significantly lesser expense compared to AlgT with a ResNet152 backbone. This underscores the efficacy of SAT’s graph refinement model in leveraging both visual and structural information to enhance pose estimation.

We notice a noticeable decrease in the metrics of Panoptic relative to H36M. This is because the Panoptic data set is different from the H36M data set in which all images can contain complete subjects, including a large number of parts that cannot shoot complete subjects. Many of the selected sequences also contain more complex occlusions Experimental results show that our method still achieves better results than the Baseline in CMU Panoptic dataset. In particular, SAT using small-scale Baseline ResNet50 surpasses the baseline using ResNet152.

**Comparison with Existing Methods.** Our method is compared with existing methods, including existing algebraic methods, similar methods and state-of-the-art methods in human pose estimation on the Human36M dataset, and Table 5 shows the comparison results. This table is divided into three sections, starting with a comparison with the single-frame algebraic methods, RANSAC is an improved method by [16]. The Lagrangian Method in the table is the iteration-based method reproduced by [4], which is used as the baseline of Structural triangulation (ST) [4], ST necessitates estimating bone lengths for pose refinement; to maintain uniformity in experimental settings, we confined our estimation of subject-specific bone lengths to the current frame’s image data(The -S suffix in the table). The second part of the table shows the comparison of similar methods. The structural triangulation (ST) method introduces a mathematical method that leverages known bone lengths to perform more accurate structure estimation and is a similar approach to SAT. Unlike ST, SAT is a data-driven algorithm that learns skeletal priors directly from training data. To compare these two methods, we conducted a detailed comparative test. SAT learns prior information directly from the training data, for ST, various strategies for bone length estimation were employed: utilizing ground truth values directly (ST-GT), estimating average bone lengths individually for each person in the test set (ST), calculating an average bone length across all frames in the test set (ST-A), estimating bone lengths per frame for testing(ST-S), and adopting the average bone length from the training set as a universal prior (ST-T-A). All bone length estimation methods are the same as in [4]. The last part of the table is a comparison with the state-of-the-art method on Human36M.

Experimental results indicate that our method achieves the best performance among single-frame algebraic methods. And SAT also surpasses the algebraic triangulation-based method AlgT as well as previous methods, is closely competitive with the current state-of-the-art. Skeletal Volume Triangulation (SVT), an advanced method of volumetric triangulation refined with our 3DRM, outperforms all previous methods and has achieved state-of-the-art performance on the Human3.6M dataset. Experimental results show that the similarity method ST achieves the state of the art when the bone length is known (estimated using groundtruth). ST is also better than SAT when the bone length is known and accurate for each individual. Nonetheless, in practical situations, estimating bone length requires subject identification and traversal of image sequences, which is not feasible in a single-frame context. It is also not easy to obtain the exact length of bones, and identification is required for scenes where different people may appear. Conversely, SAT excels in single-frame scenarios, particularly when utilizing universally scaled skeletal priors derived from a population. In addition, SAT(18.65 mm) surpasses the reported of the multi-person pose estimation model VoxelPose [28](19.0 mm) on Human3.6M and is on par with the reported of MvP [35](18.6 mm). SVT, with an even lower error of 16.71 mm, exceeds the performance of both VoxelPose and MvP.

**Table 5.** Comparison with existing methods

	Avg	Dir.	Disc.	Greet	Phone	Pose	Purch.	Smoke	Photo	Wait	Walk	WalkD.	WalkT.
Comparison of single-frame algebraic methods													
RANSAC [16]	24.39	22.01	23.96	21.71	25.26	22.83	23.31	26.33	25.69	22.07	26.52	25.30	27.70
AlgT [16]	20.89	18.55	20.09	17.70	21.61	18.57	20.84	21.58	22.37	18.59	24.35	22.33	24.14
Lagrangian [4]	19.49	18.14	20.53	18.38	20.07	18.00	19.67	20.60	19.99	19.50	19.02	21.10	18.88
ST-S [4]	19.89	17.98	20.36	18.29	19.92	17.91	<b>19.57</b>	20.27	<b>19.68</b>	19.31	18.91	20.83	<b>18.73</b>
SAT(ours)	<b>18.65</b>	<b>17.29</b>	<b>18.18</b>	<b>16.72</b>	<b>19.08</b>	<b>17.74</b>	20.13	<b>19.41</b>	20.16	<b>17.08</b>	<b>18.68</b>	<b>20.47</b>	18.88
Comparison of similar methods													
ST-GT	<b>17.53</b>	<b>16.00</b>	18.54	16.81	<b>17.79</b>	<b>16.47</b>	<b>17.43</b>	<b>18.19</b>	<b>17.82</b>	17.82	<b>17.52</b>	<b>18.67</b>	<b>17.33</b>
ST	18.52	16.81	19.67	18.05	18.61	17.43	18.58	18.84	18.51	18.97	18.84	19.78	18.20
ST-S	19.89	17.98	20.36	18.29	19.92	17.91	19.57	20.27	19.68	19.31	18.91	20.83	18.73
ST-T-A	31.08	35.19	29.17	30.60	33.40	53.96	27.22	26.22	24.82	28.30	27.33	28.89	27.82
ST-A	19.44	17.94	20.69	19.34	19.27	18.70	19.38	19.44	19.20	19.88	19.59	20.47	19.33
SAT(ours)	18.65	17.29	<b>18.18</b>	<b>16.72</b>	19.08	17.74	20.13	19.41	20.16	<b>17.08</b>	18.68	20.47	18.88
Comparison with SOTA													
CVF [24]	31.2	28.9	32.5	28.1	29.3	28.0	36.8	35.6	29.3	30.0	30.0	28.3	30.5
GHPT [1]	29.1	27.5	28.4	27.5	30.1	27.9	30.8	30.8	28.1	29.4	30.5	28.5	30.1
ET [14]	25.7	27.7	23.7	24.8	26.9	24.9	26.5	28.2	31.4	26.4	28.3	23.6	23.5
TF [17]	24.6	24.2	26.4	21.1	25.2	23.2	24.7	26.4	26.8	24.2	23.2	26.1	23.3
AlgT [16]	20.89	18.55	20.09	17.70	21.61	18.57	20.84	21.58	22.37	18.59	24.35	22.33	24.14
ST-GT [4]	17.53	16.00	18.54	16.81	17.79	16.47	<b>17.43</b>	18.19	<b>17.82</b>	17.82	17.52	18.67	17.33
MFT+ [26]	25.8	23.4	25.2	24.4	27.4	22.8	25.2	25.9	28.5	23.6	26.6	22.6	22.7
VolT [16]	17.62	16.13	18.51	15.39	17.80	16.21	18.26	17.20	19.34	16.29	18.22	18.55	19.54
SAT(ours)	18.65	17.29	18.18	16.72	19.08	17.74	20.13	19.41	20.16	17.08	18.68	20.47	18.88
SVT(ours)	<b>16.71</b>	<b>15.62</b>	<b>17.05</b>	<b>15.22</b>	<b>16.47</b>	<b>16.17</b>	17.64	<b>16.99</b>	18.11	<b>14.95</b>	<b>16.16</b>	<b>17.26</b>	<b>16.18</b>

#### 4.4 Evaluation of Computational Overhead

Due to the integration of a 2DRM and 3DRM atop algebraic triangulation, an increase in computational overhead is inevitable for SAT. Consequently, we assessed the computational overhead. All experiments were performed on a system equipped with i5-13600KF, 32 GB RAM, RTX 3060(8 GB memory), and all experiments were based on H36M. Table 6 presents an evaluation of Computational overhead. The computational overhead metric used in the table is milliseconds. We further compared SAT with other methods, and Table 7 summarizes these comparative outcomes. In the table, the time metric is milliseconds and the memory metric is MB. The -50 suffix in the table represents that the backbone is replaced with ResNet50.

It is evident that the computational cost of 2DRM exceeds that of 3DRM. Because 2DRM must handle each view, whereas 3DRM only refines the overall 3D pose. It is observed that the overall performance overhead of the graph convolutional model is minimal, requiring approximately 5 ms, which closely approximates the computational cost of the purely mathematical method, ST (4 ms under identical conditions). It can also be observed that the parameter increment introduced by SAT is minimal, showing no substantial increase compared

**Table 6.** Comparison of computational overhead of each part

Part	Pred-time
Backbone	22
2DRM	3
3DRM	2

**Table 7.** Comparison of Computational overhead with other methods

	pred-Time	Train-Time	pred-Mem	Trai-Mem	Param
AlgT-50	58	511	815	2210	44886114
SAT-50(ours)	65	670	831	2251	46470895
AlgT	89	870	853	3694	79521890
SAT(ours)	94	1040	920	3714	79555535
VolT	100	2011	1176	5220	80588050

to the identical baseline. This highlights the lightweight of SAT’s core 2DRM and 3DRM.

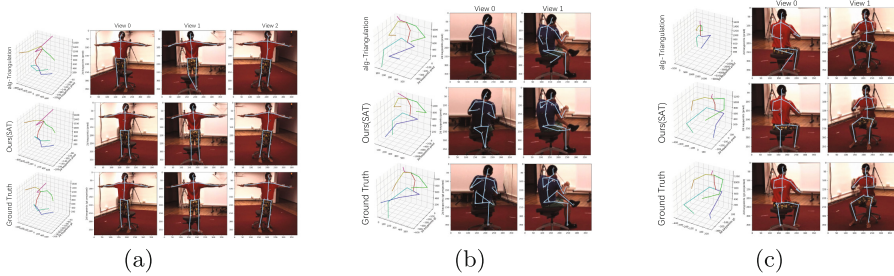
**Generalization Experiment.** Triangulation, known for its strong generalization, is used for data annotation in GT-absent environments and diverse scenarios. Iskakov et al. [16] proved its efficacy. However, real-world camera setups may differ from ideal conditions, warranting investigation of view and scene impacts on model performance. We trained on Human36M with fixed four views but tested with random 2–4 views, repeating 100 times to gauge model generalization across camera configurations. The random seed was fixed to ensure consistent view selection in each validation run. We compared the mean and standard deviation of MPJPE between the baseline and SAT in the random view test. In the table, the suffix -50 represents the replacement of the backbone with ResNet50, the prefix AVG- represents the mean and STDEV- represents the standard deviation.

**Table 8.** Experimental results of generalization performance on H36M

	Avg	Dir.	Disc.	Greet	Phone	Pose	Purch.	Smoke	Photo	Wait	WalkD.	Walking	WalkT.
AVG-AlgT	36.34	<b>40.64</b>	<b>32.70</b>	86.58	32.71	<b>32.82</b>	32.12	32.59	33.89	34.80	<b>36.40</b>	31.89	37.94
AVG-SAT(ours)	<b>33.28</b>	42.47	33.41	<b>33.86</b>	<b>29.85</b>	40.63	<b>31.62</b>	<b>30.21</b>	<b>32.75</b>	<b>30.91</b>	36.60	<b>29.10</b>	<b>32.13</b>
STDEV-AlgT	10.60	<b>16.39</b>	<b>5.79</b>	227.05	4.23	<b>8.17</b>	6.75	3.46	<b>6.03</b>	10.91	<b>13.77</b>	<b>4.26</b>	11.52
STDEV-SAT(ours)	<b>3.93</b>	23.25	11.12	<b>15.77</b>	<b>3.78</b>	41.48	<b>6.74</b>	<b>3.34</b>	<b>6.22</b>	<b>7.21</b>	40.28	4.06	<b>10.44</b>
AVG-AlgT-50	79.80	70.05	79.46	77.25	93.40	78.10	54.17	56.68	57.58	78.01	43.74	71.73	44.83
AVG-SAT-50(ours)	<b>46.93</b>	<b>52.42</b>	<b>54.04</b>	<b>45.43</b>	<b>60.72</b>	<b>58.29</b>	<b>41.09</b>	<b>39.67</b>	<b>41.48</b>	<b>53.69</b>	<b>36.42</b>	<b>35.66</b>	<b>37.56</b>
STDEV-AlgT-50	33.28	52.07	69.80	70.92	93.20	102.81	17.41	12.46	16.72	118.14	13.47	86.69	20.94
STDEV-SAT-50(ours)	<b>10.35</b>	<b>27.61</b>	<b>43.23</b>	<b>20.35</b>	<b>77.31</b>	<b>64.38</b>	<b>12.07</b>	<b>4.63</b>	<b>9.82</b>	<b>49.00</b>	<b>9.51</b>	<b>6.23</b>	<b>12.22</b>

Table 8 shows the results of the generalization experiments, using a different view order and number of views than the train set. Compared to the case where the same set of views is used for both training and testing, the error of a single view is magnified in different views. The integration of structural prior and visual feature in the refined model alleviates this inaccuracy, while the dependencies learned by the model across multiple views provide enhanced generalization capabilities and robustness. A significant decrease in performance is observed

for smaller backbones. In the case of a small number of cameras, the impact of single view error on 3D coordinates is more serious.



**Fig. 4.** shows the visualization of the results using PoseResNet50, (a) shows the results for three views, (b) and (c) show the results for two views.

Figure 4 provides a visualization of the results derived from PoseResNet50. In the case of only 3 views (Fig. 4 (a)), the wrong 3D coordinates of a single view have some impact, and the bias caused by this error is more severe in the case of only 2 views. (Figure 4 (b)) The bias due to occlusion is more severe, and (Fig. 4 (c)) the 3D joints due to occlusion have obvious errors. Our SAT initially corrects 2D poses and subsequently improves 3D reconstruction performance by integrating multi-view visual features, encoding human skeleton priors, and learning inter-view dependencies throughout training. These enhancements under abnormal conditions demonstrate the robustness of SAT in handling adversity. We also conduct generalization performance experiments on the CMU Panoptic dataset. CMU Panoptic differs from H36M by containing a large number of images from different views. We tested the trained model with a different number of views (more or less than the training views) and with views completely different from the training view. Table 9 shows the experimental results, and Fig. 5 shows the visualizations of testing and training at different views.

**Table 9.** Experimental results of generalization performance on CMU Panoptic

	Same views	Less views	More views	Different views
AlgT	28.31	49.21	30.46	30.68
SAT	<b>27.31</b>	<b>46.22</b>	<b>29.65</b>	<b>29.79</b>
AlgT-50	29.59	49.42	33.89	42.27
SAT-50	<b>28.71</b>	<b>47.94</b>	<b>31.98</b>	<b>30.28</b>

Similar to the experimental results on H36M, fewer views lead to a magnified error for a single view and a noticeable decrease in 3D pose accuracy. The dependencies between visual features and multiple views are learned by SAT, mitigating this error. The visualization also shows that SAT makes fewer false predictions.

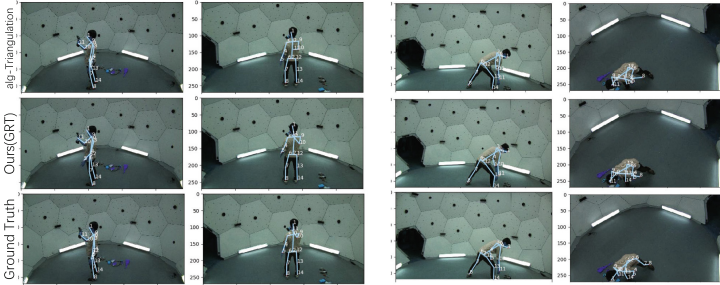


Fig. 5. shows the visualization of the results on CMU Panoptic

We also observe that SAT achieves better results with more views than trained or with completely different views. In addition, testing with more views or completely different views than training, SAT results were comparable to both training and testing with the same viewpoint. This also illustrates the generalization of SAT.

## 5 Conclusion

In this paper, we propose a refinement module encoding skeletal priors, leading to the introduction of Structured Algebraic Triangulation (SAT), a learnable algebra method that embedding structural information. Experimental results show that SAT outperforms algebraic methods and is on par with state-of-the-art methods. Additionally, enhanced generality and stability are showcased. Our method can be adapted to different backbones, with the refinement model applicable for boosting the performance of existing 2D and 3D pose estimation techniques. The integration of the graph refinement module in volumetric triangulation, called SVT, achieves state-of-the-art on the Human36M dataset. The proposed method facilitates new scene data acquisition, with minimally pre-processed data easily utilized for training other multi-view 3D pose models. SAT primarily employs lightweight graph convolution operations and triangulation processes, thereby incurring modest computational overhead. This enables fine-tuning and real-timepose estimation even on low-performance hardware. In con-

trast to various deep learning-based methods with high computational demands, SAT is better suited for deployment in resource-constrained environments like edge devices.

## References

1. Bartol, K., Bojanić, D., Petković, T., Pribanić, T.: Generalizable human pose triangulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11028–11037 (2022)
2. Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., Ilic, S.: 3d pictorial structures for multiple human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1669–1676 (2014)
3. Cai, Z., et al.: Humman: multi-modal 4d human dataset for versatile sensing and modeling. In: European Conference on Computer Vision, pp. 557–577. Springer, Heidelberg (2022)
4. Chen, Z., Zhao, X., Wan, X.: Structural triangulation: a closed-form solution to constrained 3d human pose estimation. In: European Conference on Computer Vision, pp. 695–711. Springer, Heidelberg (2022)
5. Ci, H., Wang, C., Ma, X., Wang, Y.: Optimizing network structure for 3d human pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2262–2271 (2019)
6. Dong, J., Fang, Q., Jiang, W., Yang, Y., Bao, H., Zhou, X.: Fast and robust multi-person 3d pose estimation and tracking from multiple views (2021)
7. Dong, J., Jiang, W., Huang, Q., Bao, H., Zhou, X.: Fast and robust multi-person 3d pose estimation from multiple views (2019)
8. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
9. Gong, K., Zhang, J., Feng, J.: Poseaug: a differentiable pose augmentation framework for 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8575–8584 (2021)
10. HanbyulJoo, T., XulongLi, H., LeiTan, L., SeanBanerjee, T.: Panoptic studio: a massively multiview system for social interaction capture. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(1) (2019)
11. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2003)
12. Hartley, R.I., Sturm, P.: Triangulation. *Comput. Vis. Image Underst.* **68**(2), 146–157 (1997)
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, Y., Yan, R., Fragkiadaki, K., Yu, S.I.: Epipolar transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7779–7788 (2020)
15. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(7), 1325–1339 (2013)
16. Isakov, K., Burkov, E., Lempitsky, V., Malkov, Y.: Learnable triangulation of human pose. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7718–7727 (2019)

17. Ma, H., et al.: Transfusion: cross-view fusion with transformer for 3d human pose estimation. arXiv preprint [arXiv:2110.09554](https://arxiv.org/abs/2110.09554) (2021)
18. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2640–2649 (2017)
19. Mehta, D., et al.: Vnect: real-time 3d human pose estimation with a single rgb camera. *ACM Trans. Graph. (tog)* **36**(4), 1–14 (2017)
20. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Harvesting multiple views for marker-less 3d human pose annotations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6988–6997 (2017)
21. Peng, J., Zhou, Y., Mok, P.: Ktpformer: kinematics and trajectory prior knowledge-enhanced transformer for 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1123–1132 (2024)
22. Phan, A.V., Le Nguyen, M., Nguyen, Y.L.H., Bui, L.T.: Dgcnn: a convolutional neural network over large-scale labeled graphs. *Neural Netw.* **108**, 533–543 (2018)
23. Popa, A.I., Zanfir, M., Sminchisescu, C.: Deep multitask architecture for integrated 2d and 3d human sensing. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6289–6298 (2017)
24. Qiu, H., Wang, C., Wang, J., Wang, N., Zeng, W.: Cross view fusion for 3d human pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4342–4351 (2019)
25. Qiu, L., et al.: Peeking into occluded joints: a novel framework for crowd pose estimation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 488–504. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58529-7\\_29](https://doi.org/10.1007/978-3-030-58529-7_29)
26. Shuai, H., Wu, L., Liu, Q.: Adaptive multi-view and temporal fusing transformer for 3d human pose estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(4), 4122–4135 (2022)
27. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 529–545 (2018)
28. Tu, H., Wang, C., Zeng, W.: VoxelPose: towards multi-camera 3D human pose estimation in wild environment. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 197–212. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58452-8\\_12](https://doi.org/10.1007/978-3-030-58452-8_12)
29. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
30. Wang, J., et al.: Deep 3d human pose estimation: a review. *Comput. Vis. Image Underst.* **210**, 103225 (2021)
31. Wang, J., et al.: Freeman: towards benchmarking 3d human pose estimation in the wild. arXiv preprint [arXiv:2309.05073](https://arxiv.org/abs/2309.05073) (2023)
32. Wu, S., et al.: Graph-based 3d multi-person pose estimation using multi-view images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11148–11157 (2021)
33. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: European Conference on Computer Vision (ECCV) (2018)
34. Yu, B.X., Zhang, Z., Liu, Y., Zhong, S.H., Liu, Y., Chen, C.W.: Gla-gcn: global-local adaptive graph convolutional network for 3d human pose estimation from monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8818–8829 (2023)



35. Zhang, J., Cai, Y., Yan, S., Feng, J., et al.: Direct multi-view multi-person 3d pose estimation. *Adv. Neural. Inf. Process. Syst.* **34**, 13153–13164 (2021)
36. Zhao, L., Peng, X., Tian, Y., Kapadia, M., Metaxas, D.N.: Semantic graph convolutional networks for 3d human pose regression. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3425–3435 (2019)
37. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: Motionbert: a unified perspective on learning human motion representations. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15085–15099 (2023)



# Reassembling Broken Objects Using Breaking Curves

Ali Alagrami<sup>1</sup>, Luca Palmieri<sup>1</sup>, Sinem Aslan<sup>1,2</sup>, Marcello Pelillo<sup>1</sup>,  
and Sebastiano Vascon<sup>1,2</sup>(✉)

<sup>1</sup> Department of Environmental Science, Informatics and Statistics, Ca'Foscari  
University of Venice, Via Torino 155, Mestre (Venice), Italy

<sup>2</sup> European Centre for Living Technology, Dorsoduro 3911, 30123 Venice, Italy  
sebastiano.vascon@unive.it

**Abstract.** Reassembling 3D broken objects is a challenging task. A robust solution that generalizes well must deal with diverse patterns associated with different types of broken objects. We propose a method that tackles the pairwise assembly of 3D point clouds, that is agnostic on the type of object, and that relies solely on their geometrical information, without any prior information on the shape of the reconstructed object. The method receives two point clouds as input and segments them into regions using detected closed boundary contours, known as *breaking curves*. Possible alignment combinations of the regions of each broken object are evaluated and the best one is selected as the final alignment. Experiments were carried out both on available 3D scanned objects and on a recent benchmark for synthetic broken objects. Results show that our solution performs well in reassembling different kinds of broken objects. The code is available at <https://github.com/RePAIRProject/AAFR>.

**Keywords:** Puzzle Solving · 3D Reassembly · Pairwise Geometric Reassembly

## 1 Introduction

Reconstructing three-dimensional broken objects is an important task in several fields such as computer graphics [8, 16], cultural heritage [14, 15], and robotics [4, 10, 21]. The growing interest in the community toward the 3D multi-part assembly task in recent years led to the development of a benchmark composed of realistically broken objects [16].

While there are numerous methods for the registration of 3D points, e.g., [2, 9, 18, 22], reassembling two parts of a broken object is a different task that usually requires registering only a partial subset of each part. Some registration methods address this issue by focusing on the low-overlap region [9], however accurately identifying the fractured surface region is important for performing

---

A. Alagrami, L. Palmieri and S. Aslan—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15318, pp. 197–208, 2025.  
[https://doi.org/10.1007/978-3-031-78456-9\\_13](https://doi.org/10.1007/978-3-031-78456-9_13)

pairwise matching over such point subsets. Indeed, the success of the reassembly depends highly on the precision of the segmentation process, and developing an algorithm that accurately identify fractured surface regions without making assumptions about the shape of the object is challenging. To deal with this issue, prior works [1, 8, 17] adopted extraction of *breaking curves* in an initial step, and achieve segmentation by merging vertices that are not part of the breaking curve into a single region. Other approaches adopted graph-based techniques for segmentation of point clouds, as outlined in Sect. 2. These have been successfully used for extracting spatial geometric attributes from 3D point cloud data [5, 6, 12].

We propose a modular and adaptable open-source<sup>1</sup> framework that integrates geometric-based methods to effectively reassemble pairs of 3D broken objects, without making any assumptions about their type or the nature of their damage. The proposed approach offers a significant advantage in obtaining region segmentation independent of surface characteristics. This is achieved through the guidance of *breaking curves*, which are extracted using an extension of the graph-based method in [5]. We experimentally demonstrate that, if the breaking curve extraction and the successive segmentation steps are successfully achieved, it is possible to accomplish the registration stage with a standard registration method such as the Iterative Closest Point (ICP) [2]. We evaluated the proposed approach on a state-of-the-art synthetic benchmark as well as two real-world datasets. The results demonstrate the robustness and accuracy of the proposed method, as presented in Fig. 1.



**Fig. 1.** The proposed method reconstructs accurately the mug by assembling the two parts, where the other approaches fail drastically in this case.

## 2 Related Work

### 2.1 Non-learning Based (Geometrical) Methods

A common approach for automatic reassembly of broken 3D objects relies on fractured region matching for identifying potential pairwise matches of fragments. This involves (*i*) segmentation of the broken objects into fractured and

<sup>1</sup> The code is available at <https://github.com/RePAIRProject/AAFR>.

intact regions and (ii) matching of the fractured surfaces. A conventional technique for surface segmentation is to use *region growing*, where vertices with similar attributes are combined in the same region.

The region growing segmentation relies either on the contours or on the surface characteristics. Altantsetseg et al. [1] adopted the Fourier series to approximate the boundary contour, Huang et al. [8] extracted the long closed cycles from a minimum spanning graph of the edge points that have persistent curvatures at multiple scales. Several works used breaking curves for aligning fragments after segmenting them [19, 23], yet they do not consider deteriorated fragments. Some other works adopted features computed on the fractured surfaces for their alignment, e.g., concave and convex regions were extracted on the fractured surfaces by Li et al. [11] and Son et al. [17], and Huang et al. [8] adopted clusters of multi-scale surface characteristics computed based on the integral invariants. Papaioannou et al. [14] conducted an exhaustive search of fractured surfaces of all fragments, rather than extracting features.

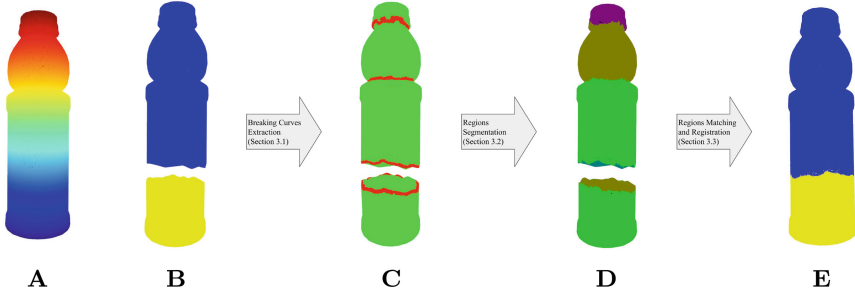
## 2.2 Learning-Based Methods

Another approach adopted by the recent literature involves learning-based techniques to estimate the transformation required for the reassembly of fragments. In this context, Chen et al. [3] created a synthetic dataset by breaking 3D meshes into pairs of fragments and employed a transformer-based network with a loss that is a combination of geometric shape-based and transformation matrix-based loss functions to learn pairwise alignment. The reported results highlight the high complexity of this task, given that synthetically generated fragments devoid of physical deterioration were only roughly aligned [16]. This trend is further validated by Sellan et al. [16], which introduced a physically realistic dataset of broken 3D meshes to serve as a benchmark for the reassembly task and demonstrated that baseline learning-based algorithms are insufficient for solving the multi-part assembly task. In this work, we follow the first approach, i.e., segment the broken surfaces as in [1, 8, 17] and register each segmented broken region with an exhaustive search as in [14]. Unlike them, we use a graph-based method for detecting the breaking curves of fragments which allows segmenting regions without prior assumptions on the surface characteristics of the object, and adopt the ICP algorithm for registration.

## 3 The Proposed Approach

The proposed method has a modular workflow depicted in Fig. 2, which is divided into three main parts:

1. Detecting breaking curves: the set of points which belong to a three-dimensional edge (Sect. 3.1),
2. Segmenting the points into a set of regions using the breaking curves (Sect. 3.2),
3. Registering the objects by selecting the best match among possible combinations of the segmented regions of each objects (Sect. 3.3).



**Fig. 2.** The pipeline of the proposed approach. The 3D object (a drinking bottle from Breaking Bad dataset [16], **A**) is broken into two parts (**B**), which are the input to our algorithm. The algorithm detects the breaking curves (**C**) and segment the regions (**D**). The registration selects the best match among the segmented regions and reassemble the two parts (**E**).

### 3.1 Breaking Curves Extraction

When dealing with the assembly of fragmented objects, it is crucial to detect borders and edges as they provide cues for the correct matching. The proposed approach starts from a 3D point cloud and detects breaking curves. A breaking curve is defined as a subset of connected points that belong to a 3D edge, as illustrated in Fig. 3b. The set of all breaking curves acts as a support for segmenting the objects into distinct regions.

We do not make any assumption regarding the type of fracture, their size or location. While this allows us to work with data coming from different sources, it introduces the possibility of having one large broken region or several small ones. The proposed approach detects breaking curves in different parts of the fragment, which may result in over-segmentation, creating more broken region (as shown in red in Fig. 2 step C). However, it is enough to obtain even a part of the correct pair of broken regions to obtain a correct registration.

Let  $P$  be the set of points in a point cloud. We represent  $P$  as an unweighted directed graph  $G = (V, E)$  where the set of vertices  $V$  corresponds to the set of points  $p \in P$  and the edges  $E \subseteq V \times V$  represents the neighbouring relations between the points. Being the density of the point cloud non-uniform, we opted for a mixed approach when adding edges: we create an  $\epsilon$ -graph [13, 20] using the average distance of the  $k$  nearest neighbours considering the entire point cloud. The  $\epsilon$  value is then computed as:

$$\epsilon = \frac{1}{|P|} \frac{1}{k} \sum_{p \in P} \sum_{q \in \mathcal{N}_p^k} |p - q| \quad (1)$$

Here  $P$  is the point cloud,  $p \in P$  is a 3D-point in  $\mathbb{R}^3$  and  $\mathcal{N}_p^k$  is the set of  $k$ -nearest neighbours of point  $p$ .

After the graph is created, we compute for each node its *corner penalty* [5]

defined as:

$$\omega_{co}(p) = \frac{\lambda_2(p) - \lambda_0(p)}{\lambda_2(p)} \quad (2)$$

where  $\lambda_0$  and  $\lambda_2$  are respectively the smallest and the largest of the three eigenvalues of the correlation matrix of the neighbours of  $p$ . The eigenvalues of the correlation matrix provide the level of skewness of the ellipsoid enclosing the points.

Intuitively, if the point  $p$  lies on a flat area (i.e. the surface), one would have  $\lambda_2 \approx \lambda_1$  and  $\lambda_2 \approx 0$ , while if the point lies on a corner, the eigenvalues should approximately be the same ( $\lambda_2 \approx \lambda_1 \approx \lambda_0$ ) [5]. If the corner penalty tends to 1, the node is likely to be on a flat area. We select all nodes whose corner penalty is less than a threshold to obtain a noisy initial version of the *breaking curves*. The final version is obtained after applying a refinement step similar to the morphological operation of opening. A pruning step is followed by a dilation to remove small isolated branches and promote the creation of closed breaking curves. Given a point cloud  $P$  we define  $\mathcal{B}^P$  as the set of points in  $P$  that are part of a breaking curve.

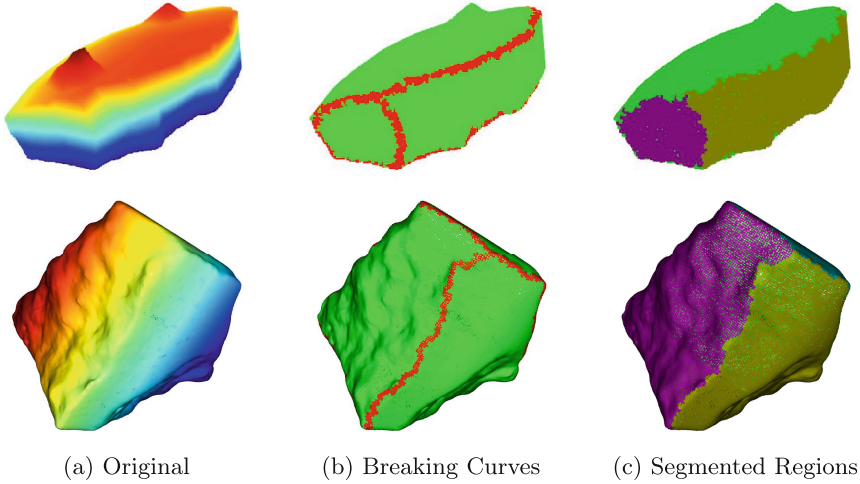
### 3.2 Regions Segmentation

Regions are extracted using a region-growing approach constrained by the previously extracted breaking curves. Given a point  $p \notin \mathcal{B}^P$  we define the  $i$ -th region  $\mathcal{R}_i^P$  and assign  $p$  to it. We consider the set of  $q \in \mathcal{N}_p$  and include each  $q$  in the region  $\mathcal{R}_i$  if  $q \notin \mathcal{B}^P$ . This procedure is iterated until all  $p \notin \mathcal{B}^P$  are considered. This results in segmenting the point cloud  $P$  into several regions  $\mathcal{R}^P$  enclosed by the breaking curves.

The only points that remain unassigned to a region are those that belong to the breaking curves. However, the breaking curve shape can also aid in the matching phase. Thus, a  $k$ -NN voting scheme is employed to assign these points to a segmented region. For each *boundary* point in the breaking curves we count the number of its neighbouring points which are not labeled as boundary. If more than  $\tau = 50\%$  of the neighbouring points belong to one segmented region, that boundary point is assigned to that segmented region. If the algorithm completes a run over all border pixels without any changes, the threshold  $\tau$  is decreased until all points are assigned to any segmented region. This straightforward heuristic yields acceptable outcomes (see Fig. 3c). This operation is applied to the *dilated* boundaries, some of which naturally reside within specific segments' regions.

### 3.3 Region Matching and Registration

The final step involves aligning the fragments using the segmented regions. Given two segmented point clouds  $P$  and  $Q$ , we attempt to register the regions in  $\mathcal{R}^P$  with the one in  $\mathcal{R}^Q$ . To this end, we first discard regions having a number of nodes below a certain threshold. This step has two beneficial effects: reducing the computational effort and making the method more robust to noisy regions. The



**Fig. 3.** An example of the pipeline on both synthetic (top) and real (bottom) data: after processing the original point cloud, the borders (in red) are detected and the regions are segmented accordingly (different colors). (Color figure online)

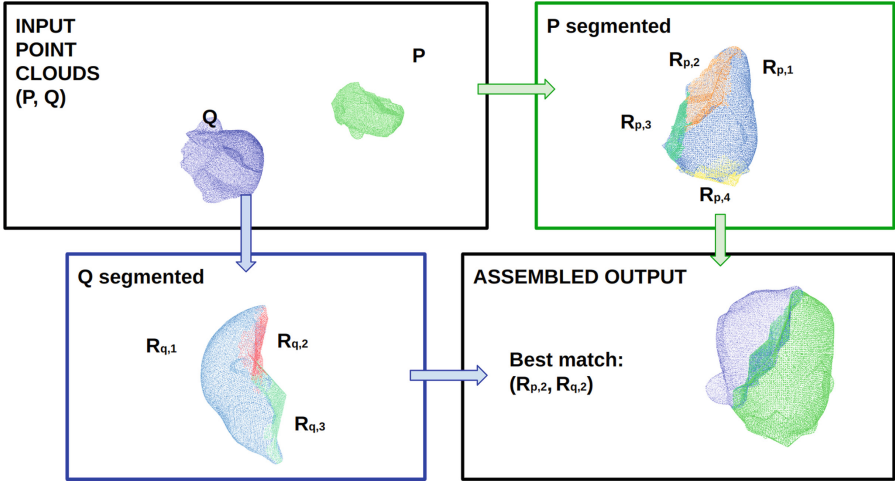
registration is achieved with an exhaustive search of all the remaining regions matches. Given a pair of regions  $\mathcal{R}_i^P$  and  $\mathcal{R}_j^Q$ , we register them with ICP [2] and compute the Chamfer Distance (CD) as their matching score. The pair with the best score is selected and their transformation is used for the final alignment.

## 4 Experiments

We evaluate our model on two available datasets of both synthetic and real scanned 3D objects and on an in-house set of scanned 3D fresco fragments from the Pompeii Archaeological Site collected under the RePAIR project<sup>2</sup>. In particular, we experimented on a subset of categories of the Breaking Bad (BBad) dataset [16] having enough variability in terms of object characteristics, and one sample of TU-Wien dataset [8] since it was sufficient to explore whether the proposed algorithm is capable of solving the reassembly task with objects coming from different data sources.

Figure 4 illustrates the experimental setup where the input consists of two point clouds,  $P$  and  $Q$ , which are randomly translated and rotated. The output of the algorithm is the transformation that aligns point cloud  $Q$  to point cloud  $P$ , enabling the assembly of the two broken parts.

<sup>2</sup> For more information, please visit <https://www.repairproject.eu/>.



**Fig. 4.** An example experiment demonstrating the *input*, *segmentation*, and *matching* outcomes of our approach. Here, the point clouds are sourced from a Toy figure in the Breaking Bad dataset [16].

We compare our method against the Generative 3D Part Assembly (DGL) method proposed in [7], which was reported as the superior method on the BBad dataset in [16]. As a baseline we also include ICP [2] into our evaluation<sup>3</sup>.

Despite other approaches for assembling 3D broken objects [1, 8, 14] exists, we do not report a comparison with them for two reasons: *i*) these algorithms have a high dependence on particular characteristics of the broken objects, and *ii*) they are complex to reproduce due to a large number of parameters. Moreover, they are not suitable for assembling synthetic objects, as they differentiate broken and intact regions of the objects based on the surface roughness [8] or use feature curves to complete the reassembly [14].

Although Neural Shape Mating (NSM) [3] reported promising results in the pairwise assembly task, we choose DGL as our competitor since we consider our work as a building block for the multi-part assembly task. Moreover, NSM is using an adversarial shape loss, which requires the complete object reconstruction after pairwise assembly, while our approach, as visible in Fig. 5, correctly assembles incomplete broken parts with no need for the complete object reconstruction, an important step towards real-cases multi-part assembly.

**Quantitative Results.** We followed [16] for the choice of the metric using the root mean square error of the translation and the relative rotation. Quantitative results are presented in Table 1. To ensure a fair comparison we list the best outcome of DGL across any pair belonging to a certain category. Our method

<sup>3</sup> We trained from scratch the DGL on only pairs of fragments following authors' implementation and used the Open3D implementation for ICP.



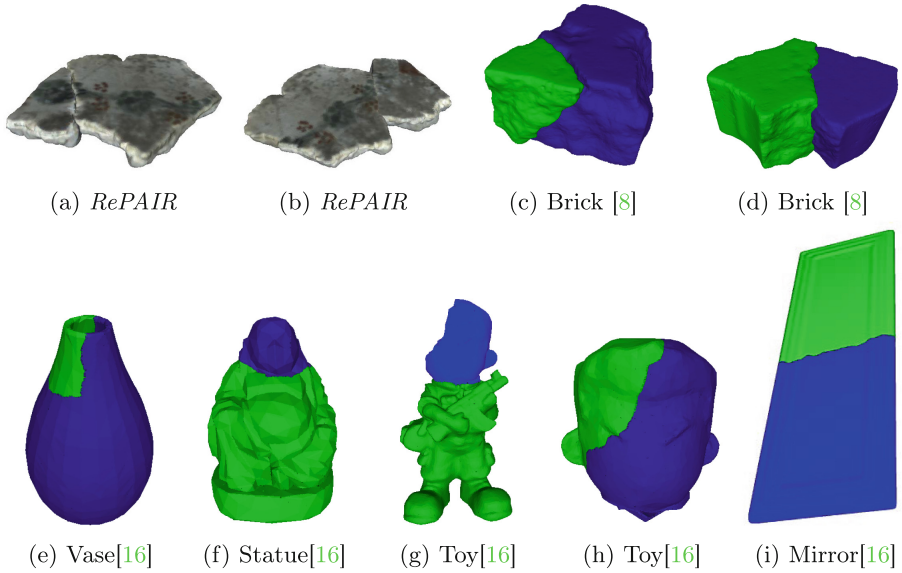
significantly outperforms DGL and ICP in all datasets in terms of relative rotation error and in the majority of datasets in terms of translation error.

We note here that for some categories (Mirror, Cup, Repair) the ICP results show very low error. This happens because the broken parts are merged and almost completely overlap, but the solution is not satisfactory (See Fig. 1.e).

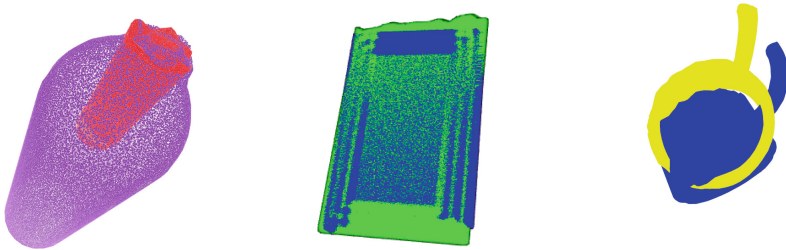
**Table 1.** Preliminary quantitative evaluation. The top rows refer to the synthetic breaking bad dataset [16] and the last two rows refer to real scanned objects.  $\spadesuit$  For DGL, we take the best value for each category.  $\clubsuit$  Scanned objects, where the solution is obtained from manual alignment (Brick from TU Wien Dataset [8] and fresco fragments from the RePAIR Project).

Category	Relative RMSE (R)			RMSE (T)		
	ICP [2]	DGL $\spadesuit$ [7]	ours	ICP [2]	DGL $\spadesuit$ [7]	ours
BeerBottle	57.028	78.933	<b>1.62</b>	1.104	0.073	<b>0.02</b>
WineBottle	54.262	84.699	<b>1.58</b>	0.743	0.024	<b>0.02</b>
DrinkBottle	60.253	70.014	<b>1.89</b>	1.288	<b>0.008</b>	0.033
Bottle	68.125	76.802	<b>1.983</b>	1.198	0.078	<b>0.077</b>
Mug	5.041	86.221	<b>1.12</b>	0.364	0.164	<b>0.025</b>
Cookie	12.594	85.707	<b>1.96</b>	0.632	0.159	<b>0.043</b>
Mirror	0.593	81.454	<b>0.111</b>	0.503	0.125	<b>0.001</b>
ToyFigure	208.333	87.972	<b>1.98</b>	4.123	0.159	<b>0.079</b>
Statue	105.582	89.605	<b>0.66</b>	2.159	0.149	<b>0.003</b>
Vase	30.756	82.218	<b>0.592</b>	1.496	0.109	<b>0.002</b>
Brick $\clubsuit$ [8]	11.577	62.820	<b>3.064</b>	2.356	1.684	<b>0.626</b>
Repair $\clubsuit^2$	7.911	87.491	<b>3.466</b>	2.525	<b>0.076</b>	0.695

**Qualitative Results.** We report qualitative results in Fig. 5 showing that our method correctly reassembles the broken parts of real and synthetic broken objects. Visual inspection of the assembled objects confirms the accuracy of the reconstruction. We show results on different kinds of objects, ranging from high-quality textured data from the RePAIR dataset (texture is not used from the proposed approach, but rather used for visualization) to the scanned brick from [8] and the synthetic data from [7]. Subfigures 5 (a), (b), (c), (d), (g) and (h) are two-parts assembly of objects which are composed of more parts, highlighting the capability of our approach to handle multiple broken regions in the input point clouds. Subfigures 5 (g) and (h) have a common broken part (the blue part of the head of the toy) and hint towards possible extensions to multi-part assembly.



**Fig. 5.** A qualitative overview of our results. In the first row, we present the reassembly of real scanned objects: (a-b) show fresco fragments from the RePAIR project and (c-d) show the scanned brick from the TU Wien dataset [8]. In the second row, we illustrate the reassembly of synthetic objects from various categories of [16].



(a) Example of our approach failing on a Bottle from BBad dataset [16]. (b) Example of ICP [2] failing on a Mirror from BBad dataset [16]. (c) Example of DGL [7] failing on a Mug from BBad dataset [16].

**Fig. 6.** Example Failure Cases. Here, we present failure cases from three algorithms, showcasing their respective limitations. While our algorithm successfully identified segmented regions, it assembled the bottle’s cork upside down. In contrast, the ICP algorithm merged two point clouds instead of assembling them, and the DGL algorithm struggled to determine a suitable assembly pose. Our algorithm’s failure, though significant, underscores its relative performance compared to these benchmarks.

**Limitations and Failure Cases.** To ensure a thorough evaluation, we discuss both the limitations and failure cases of our approach and those of our competitors, illustrated in Fig. 6.

Our algorithm fails when the broken surface is not detected and segmented or when it struggles to determine the correct transformation for aligning the two parts. The detection of breaking curves and subsequent segmentation is sensitive to noise, requiring careful parameter tuning. However, with optimal parameters, our algorithm effectively identifies broken regions across diverse datasets. Sometimes, we observed our method assembling broken parts in an *inverted orientation* (See Fig. 6a). This can occur with roughly planar surfaces where alignment may still appear plausible even after a vertical flip.

Regarding ICP, it is important to note that the algorithm was originally developed for registering sets of points rather than for assembling objects, thus its lower performance was expected. Figure 6b presents example with a mirror broken in half, where the two halves are almost completely overlapped. Explaining DGL is challenging due to its complex nature. This learning-based method was originally designed to assemble objects with semantic meaning, initially focusing on furniture objects. However, it faces difficulties in adapting to unfamiliar objects that are broken into non-standard parts, as depicted in Fig. 6c.

## 5 Conclusions

We presented a robust method for the pairwise assembly of 3D broken objects which performs well across different datasets of both real and synthetic models.

The objective of this analysis is not to discuss which algorithm works better in which case, but rather to analyze the current situation. We note that: *(i)* using an off-the-shelf approach like ICP without processing the point cloud is not a viable solution, *(ii)* it is confirmed that the DGL method, which was the best performer for the published benchmark [16], although performing well for semantic assembly, does not work for the geometric reassembly of broken objects and *(iii)* using a more principled geometrical approach is a safe way to assemble broken objects.

Concerning the limitations, the proposed pipeline is sensitive to the choice of the parameters. In our experiments, we used a different set of parameters for the synthetic objects and for the real ones. There is a margin for improvements in the robustness at different steps of the pipeline.

The proposed method is presented as a building block for reassembling objects broken into multiple parts. Extending the reassembly task to multiple broken parts following a greedy approach is under exploration. Future works include detecting non-matching surfaces and designing more principled ways of selecting the best registration among many pairs of broken objects.

**Acknowledgements.** This work is part of a project that has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 964854.




## References

1. Altantsetseg, E., Matsuyama, K., Konno, K.: Pairwise matching of 3D fragments using fast fourier transform. *Vis. Comput.* **30**, 929–938 (2014)
2. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 239–256 (1992)
3. Chen, Y.C., Li, H., Turpin, D., Jacobson, A., Garg, A.: Neural shape mating: self-supervised object assembly with adversarial shape priors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12724–12733 (2022)
4. Ghasemipour, S.K.S., Kataoka, S., David, B., Freeman, D., Gu, S.S., Mordatch, I.: Blocks assemble! learning to assemble with large-scale structured reinforcement learning. In: *International Conference on Machine Learning*, pp. 7435–7469. PMLR (2022)
5. Gumhold, S., Wang, X., MacLeod, R.: Feature extraction from point clouds. In: *Proceedings of 10th International Meshing Roundtable 2001* (2001)
6. Hao, F., Li, J., Song, R., Li, Y., Cao, K.: Mixed feature prediction on boundary learning for point cloud semantic segmentation. *Remote Sens.* **14**(19), 4757 (2022)
7. Huang, J., et al.: Generative 3D part assembly via dynamic graph learning. In: *The IEEE Conference on Neural Information Processing Systems (NeurIPS)* (2020)
8. Huang, Q., Flöry, S., Gelfand, N., Hofer, M., Pottmann, H.: Reassembling fractured objects by geometric matching. *ACM Trans. Graph.* **25**, 569–578 (2006). <https://doi.org/10.1145/1141911.1141925>
9. Huang, S., Gojcic, Z., Usvyatsov, M., Andreas Wieser, K.S.: Predator: registration of 3D point clouds with low overlap. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (2021)
10. Kataoka, S., Ghasemipour, S.K.S., Freeman, D., Mordatch, I.: Bi-manual manipulation and attachment via sim-to-real reinforcement learning. *arXiv preprint arXiv:2203.08277* (2022)
11. Li, Q., Geng, G., Zhou, M.: Pairwise matching for 3D fragment reassembly based on boundary curves and concave-convex patches. *IEEE Access* **8**, 6153–6161 (2019)
12. Loizou, M., Averkiou, M., Kalogerakis, E.: Learning part boundaries from 3D point clouds. In: *Computer Graphics Forum*, vol. 39, pp. 183–195. Wiley Online Library (2020)
13. Natali, M., Biasotti, S., Patanè, G., Falcidieno, B.: Graph-based representations of point clouds. *Graph. Models* **73**(5), 151–164 (2011)
14. Papaioannou, G., et al.: From reassembly to object completion - a complete systems pipeline. *ACM J. Comput. Cult. Heritage* **10**(2), 1–22 (2017). <https://doi.org/10.1145/3009905>
15. Pintus, R., Pal, K., Yang, Y., Weyrich, T., Gobbetti, E., Rushmeier, H.: A survey of geometric analysis in cultural heritage. In: *Computer Graphics Forum*, vol. 35, no. 1, pp. 4–31 (2016)
16. Sellán, S., Chen, Y.C., Wu, Z., Garg, A., Jacobson, A.: Breaking bad: a dataset for geometric fracture and reassembly. In: *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track* (2022). <https://openreview.net/forum?id=mJWt6pOCHNy>
17. Son, T.G., Lee, J., Lim, J., Lee, K.: Reassembly of fractured objects using surface signature. *Vis. Comput.* **34**, 1371–1381 (2018)
18. Yang, H., Shi, J., Carlone, L.: TEASER: fast and certifiable point cloud registration. *IEEE Trans. Robot.* **37**, 314–333 (2020)

19. Yang, X., Matsuyama, K., Konno, K.: Pairwise matching of stone tools based on flake-surface contour points and normals. In: GCH, pp. 125–129 (2017)
20. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: EC-Net: an edge-aware point set consolidation network. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 386–402 (2018)
21. Yu, M., et al.: RoboAssembly: learning generalizable furniture assembly policy in a novel multi-robot contact-rich simulation environment. arXiv preprint [arXiv:2112.10143](https://arxiv.org/abs/2112.10143) (2021)
22. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision* **13**(2), 119–152 (1994)
23. Zhao, F., Zhou, M., Geng, G., Zhu, L.: Rigid blocks matching method based on contour curves and feature regions. *IET Comput. Vision* **12**(1), 76–85 (2018)



# Fluent and Accurate Image Captioning with a Self-trained Reward Model

Nicholas Moratelli<sup>(✉)</sup>, Marcella Cornia<sup></sup>, Lorenzo Baraldi<sup></sup>,  
and Rita Cucchiara<sup></sup>

University of Modena and Reggio Emilia, Modena, Italy  
{nicholas.moratelli,marcella.cornia,lorenzo.baraldi,  
rita.cucchiara}@unimore.it

**Abstract.** Fine-tuning image captioning models with hand-crafted rewards like the CIDEr metric has been a classical strategy for promoting caption quality at the sequence level. This approach, however, is known to limit descriptiveness and semantic richness and tends to drive the model towards the style of ground-truth sentences, thus losing detail and specificity. On the contrary, recent attempts to employ image-text models like CLIP as reward have led to grammatically incorrect and repetitive captions. In this paper, we propose Self-Cap, a captioning approach that relies on a learnable reward model based on self-generated negatives that can discriminate captions based on their consistency with the image. Specifically, our discriminator is a fine-tuned contrastive image-text model trained to promote caption correctness while avoiding the aberrations that typically happen when training with a CLIP-based reward. To this end, our discriminator directly incorporates negative samples from a frozen captioner, which significantly improves the quality and richness of the generated captions but also reduces the fine-tuning time in comparison to using the CIDEr score as the sole metric for optimization. Experimental results demonstrate the effectiveness of our training strategy on both standard and zero-shot image captioning datasets.

**Keywords:** CLIP-based Reward · Image Captioning · Vision-and-Language Models.

## 1 Introduction

The image captioning task involves a step-by-step generation of textual descriptions, where each word is produced incrementally. During this process, contextual information is taken into account by leveraging the previously generated words while also incorporating the semantic information derived from the visual features of the input image. Over the years, researchers have made remarkable progress in developing image captioning architectures in such a way that the model strives to produce captions that effectively capture the salient aspects

of the image while maintaining linguistic fluency and relevance. In the initial stages, traditional training of early architectures involved minimizing the standard cross-entropy loss. Subsequent advancements introduced reinforcement learning techniques based on policy gradient methods, as proposed by [31, 41]. Similarly, the most adopted paradigm employs SCST (Self-Critical Sequence-Training) [43], which has demonstrated notable improvements in achieving state-of-the-art results through the optimization of the CIDEr metric [50].

Despite substantial progress, the capability to generate “human-like” descriptions remains a challenge. Recently, there has been an exploration of the large-scale CLIP model [40] for evaluating image captioning performance. This led to the development of the CLIP-Score [21], which demonstrated a considerable correlation with human judgment, thereby highlighting its effectiveness as an evaluation metric. Following this direction, other evaluation metrics based on the CLIP model have been proposed [44, 45, 52]. Among them, PAC-Score [44] stands out for its greater correlation with human evaluations, obtained thanks to a positive-augmented fine-tuning strategy that has converted the CLIP embedding space towards the style of COCO captions [30]. When employed as a reward for a captioning model, these metrics exhibit impressive ability to generate semantically rich sentences. Nonetheless, they also lead to significantly longer captions that may often contain word repetitions and grammatical errors and tend to overlook the proper word order in captions, which is an essential prerequisite in text generation.

To address these issues, we propose a novel approach based on SCST, wherein the image captioning model learns to generate captions by iteratively refining its output through a self-evaluation mechanism. Our strategy encompasses two key steps. First, we conduct a fine-tuning process for a caption discriminator using a self-supervised methodology inspired by CLIP. Specifically, alongside the usual positive image-caption pairs, we introduce a set of negative texts generated by the captioning model fine-tuned with the original CLIP-S and PAC-S as reward. The overall goal is to create a self-supervised environment that improves the correlation with human judgment, preserves syntactic accuracy, and allows the model to learn from its errors. As a second step, we integrate this discriminator as the reward used to fine-tune a captioning model, further enhancing its ability to generate high-quality and semantically richer captions.

We assess the effectiveness of the proposed approach by conducting several experiments on the COCO dataset [30], thereby showcasing its robust performance across a range of different backbones. To enhance the comprehensiveness of our analysis and validate the zero-shot capability of our approach, we expand our investigations to include out-of-domain experiments conducted on additional datasets like CC3M [46], nocaps [1], and VizWiz [20], providing insights into its potential applicability in various real-world scenarios.

## 2 Related Work

**Standard image captioning architectures.** Early captioning architectures initially involved filling in predefined templates after identifying relevant objects

within the image [48,56]. Notable advancements in this field led to the adoption of CNNs for encoding images, traditionally employed in several Computer Vision tasks [7,38,39], followed by RNNs to describe the encoded visual information into natural language [24,43,51]. This approach was further refined with the incorporation of attention mechanisms [33,54], which facilitated a shift towards enhancing the generation by focusing on key regions in the image [4], eventually enriched with spatial and semantic graphs [55,57]. Currently, in addition to shifting towards Transformer-based architectures [15,16,23], a dominant strategy involves leveraging visual features from comprehensive cross-modal architectures like CLIP [47]. In this context, several directions have been explored, such as defining memory concepts to gather information from other samples [6,16] or integrating external knowledge into the architecture [28]. More recently, the advent of large scale models like LLMs and multimodal LLMs [9,10,13,49] as significantly changed the landscape of image description leading to generated captions with increased descriptive capabilities [8,19,27].

**Training strategies.** While initial captioning models were trained with a standard cross-entropy loss [24,51,54], literature in this field soon turned towards the use of reinforcement learning paradigms. This strategy entails conceptualizing the models as agents, with the primary goal of maximizing the expected reward. On this line, notable advancements have been made by adopting a reinforcement learning strategy defining the reward as non-differentiable metrics [41,43] such as BLEU [37], ROUGE [29], CIDEr [50], SPICE [2], or a combination of them [31]. Following this principle, Dai *et al.* [17] proposed a contrastive loss method to distinguish captions based on their relationship to references, while the approach proposed in [34] exploits a reward represented by a weighted combination of the CIDEr score and a discriminability loss. Slightly different is the work proposed by Ren *et al.* [42], which relies on controlling the captioning model by mapping images and sentences into a unified semantic embedding space.

Despite the effectiveness of these training schemes, especially when employed in combination with a CIDEr-based reward, the advent of pre-trained vision-and-language models like CLIP [40] has also shed light on the limitations of the traditional criteria to evaluate caption quality. In fact, while using a CIDEr-based reward can lead to aligning with the style of ground-truth captions, it can also significantly reduce the semantic richness of predicted sentences. Following this premise, our work introduces a novel training strategy, focusing on the complete removal of all reference captions involved in calculating the reward and exploiting the supervision given by a CLIP-based model fine-tuned with additional examples. Along this line, very few approaches [14,18,36,58] closely aligned with ours refer to the CLIP model to obtain more descriptive captions.

## 3 Proposed Method

### 3.1 Preliminaries

In this section, we recap the definition of the training protocol typically used in image captioning, of Contrastive Language-Image Pre-training [40], and of



learnable image captioning metrics. Also, we introduce the terminology employed in the rest of the paper.

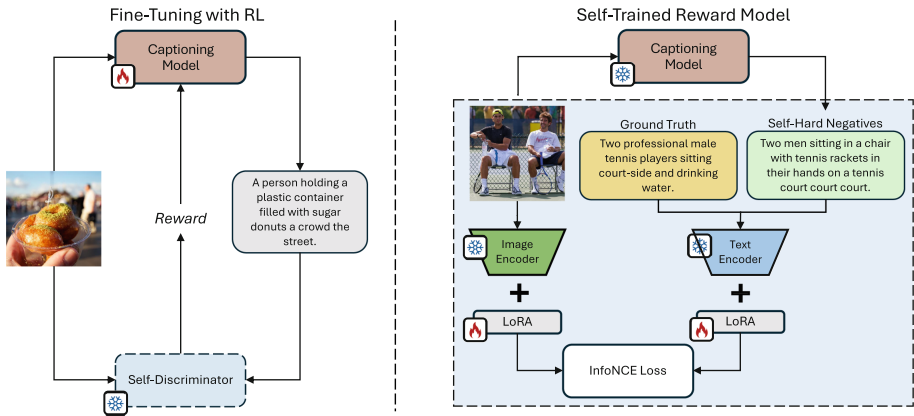
**Captioning training protocol.** Image captioning models are usually trained with a two-stage training approach. The network  $f_\theta$  is first pre-trained by encoding an image  $I_i$ , described through a sequence of  $R = (v_1, v_2, \dots, v_R)$  visual features, with a time-wise cross-entropy loss in relation to ground-truth sentences  $s_{ij} = (w_1, w_2, \dots, w_T)$ . In the second stage, the network undergoes fine-tuning through a RL strategy aimed at maximizing the CIDEr score [50] on the training dataset. During the first stage, the model is trained from scratch through a conditioning mechanism, wherein caption generation depends not only on visual features  $R$  but also on all previous ground-truth tokens up to time step  $t - 1$ , where  $w_t$  is a token belonging to a pre-defined vocabulary. During this phase,  $f_\theta$  is optimized using a cross-entropy loss (XE) as follows:

$$L_{\text{XE}}(\theta) = - \sum_{t=1}^T \log \left( P(w_t | w_{1:t-1}, R) \right). \quad (1)$$

The network then operates in an autoregressive manner, generating one token per time step. The model  $f_\theta$  outputs a discrete probability distribution, where the token  $w_t$  is chosen as the one with the highest probability, determined by preceding tokens. This selection involves passing the final network embeddings through an MLP followed by a softmax function. In the second training stage, at each time step  $t$  tokens are sampled from the probability distribution generated by the model at time step  $t - 1$ . Once the entire caption is generated, the CIDEr score is computed as reward to guide a policy-gradient RL update step [43].

**Contrastive Language-Image Pre-Training (CLIP).** CLIP [40] represents a state-of-the-art model for the computation of similarities between images and texts. In this context, the computation of matrix similarities and the training of the network through contrastive learning assume a critical role, as it serves as a fundamental step in learning the intrinsic relationships between textual and visual elements, denoted as  $T$  and  $V$  respectively. The effectiveness of the contrastive method is particularly evident when applied to large-scale datasets. Here, the matrix  $T$  is defined as comprising  $N_t$  textual instances, each characterized by a  $D$ -dimensional embedding. Likewise, the visual representation matrix  $V$  has a size of  $N_v \times D$ . To calculate the similarity matrix  $S$ , the cosine similarity function is adopted. For each textual instance  $T_i$  and visual instance  $V_j$ , the similarity score  $S_{ij}$  is computed as follows:  $S_{ij} = \text{sim}(T_i, V_j)$ , where  $\text{sim}(\cdot)$  represents the cosine similarity. This leads to a matrix  $S$ , with dimensions  $N_t \times N_v$ , where each element  $S_{ij}$  represents the similarity score between the  $i$ -th textual instance and the  $j$ -th visual instance.

**Learnable captioning metrics from human feedback.** A recent yet under-explored research direction involves leveraging a model trained with language-image pre-training as an image captioning metric, given its robust alignment capabilities between visual and textual domains. Following [21], the evaluation score of a caption  $s'_i$  can be computed with a cosine similarity  $\text{sim}(I_i, s'_i)$  between



**Fig. 1.** Overview of our approach. On the left, the training strategy of the captioner model is shown. The model acts as an agent providing rewards from a discriminator obtained with textual negatives directly derived from the model itself (right).

the visual embedding of the input image and the generated caption. In particular, in [21] a score proportional to the ReLU of the predicted similarity is employed. Additionally, to confine the score within the range of  $[0, 1]$  for convenience, the final result is scaled by a multiplicative factor denoted as  $w$ :

$$\text{Score}(I_i, s'_i) = w \cdot \text{ReLU}(\text{sim}(I_i, s'_i)). \quad (2)$$

One of the most commonly used learnable scores is CLIP-S [35], where the underlying architecture was pre-trained on 400M noisy (image, text) pairs sourced from the internet. Despite demonstrating better alignment with human judgment compared to traditional captioning metrics (*e.g.* BLEU, METEOR, CIDEr), which rely on reference captions, the use of noisy data during training leads to significant performance degradation when this score is used to directly optimize a captioning model, resulting in disparities between the score and the overall quality of captions. To mitigate this, a recent approach termed PAC-S [44] involves fine-tuning the model on cleaned data, thereby enhancing correlation with human evaluations. Specifically, PAC-S score is trained using a similarity matrix constructed from human-curated captions and machine-generated ones. Nevertheless, although these two metrics appear to yield improved correlation with humans, they tend to favor longer texts that are semantically rich yet grammatically flawed over shorter yet grammatically correct captions.

### 3.2 Self-Trained Reward Model

The SCST approach outlined in Sec. 3.1 has proven to be effective in increasing the quality of description with respect to a single XE training stage. However, it also tends to bias the model towards the “average” caption that reflects the most general mode contained in the training set [12]. This comes with some

critical disadvantages, including reduced descriptiveness, semantic richness, and discriminative power of the generated captions. What is more, one could argue that employing the CIDEr metric as a reward is an obsolete choice, as it achieves a low correlation with human judgments in comparison with recent alternatives.

Following this intuition, in this paper we propose a novel training scheme which is based on a self-supervised reward. In our approach, the classical CIDEr reward is replaced by a learnable language-image discriminator  $\mathcal{D}_r$ , which takes the form of a language-image model. Following the REINFORCE algorithm, the expected gradient of the reward function can be computed as

$$\nabla_{\theta} L_{\text{SCST}}(I_i, s'_i, \theta) = (\mathcal{D}_r(I_i, s'_i) - b) \nabla_{\theta} \log f_{\theta}(s'_i), \quad (3)$$

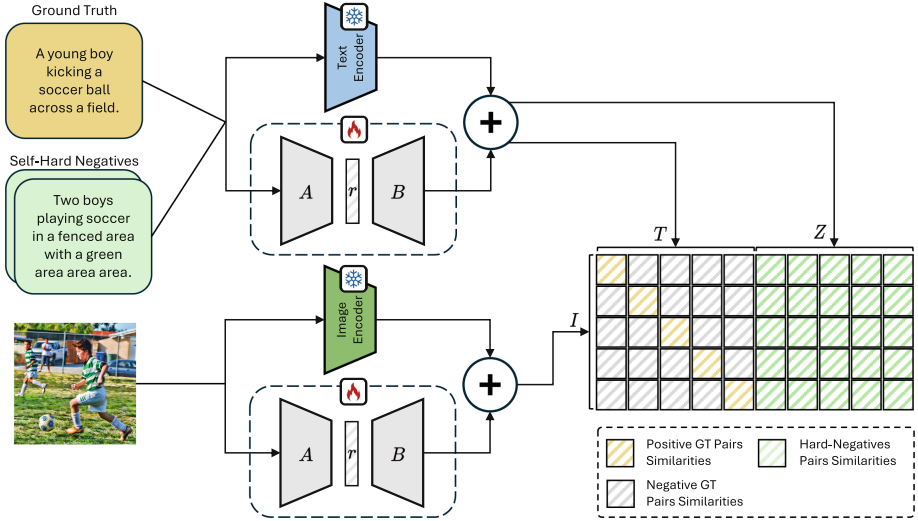
where the expected gradient has been approximated using a single Monte-Carlo sample, and  $b$  is a baseline employed to reduce the variance of the gradient estimate, which is usually computed as a function of the rewards computed inside a mini-batch. A classical choice when generating multiple descriptions for the same image through beam search is that of computing  $b$  as the average reward of all descriptions generated for  $I_i$ , so that  $b = \sum_j \mathcal{D}_r(I_i, s'_{ij})/n$ .

There are three conceptual advantages in replacing an handcrafted captioning metric with a learnable discriminator: (i) contrarily to a standard metric,  $\mathcal{D}_r$  is aware of  $I_i$  and thus can evaluate image-text alignment by “looking” at the image; (ii) being not handcrafted,  $\mathcal{D}_r$  can be trained to mimic an evaluation behavior of choice, and does not depend on the annotation style; (iii)  $\mathcal{D}_r$  is not limited to work on semantic domains on which ground-truth captions are available.

In this regard, a straightforward choice for  $\mathcal{D}_r$  would be that of employing a pre-trained CLIP model based, which also has a large semantic coverage, as explored in [14]. However, when employing learnable rewards, we observed a significant decrease of performance on reference-based metrics, which nonetheless serve as crucial benchmarks for assessing caption quality. Moreover, it is well known that CLIP-based architectures, if not properly fine-tuned, tend to focus heavily on the semantics of the caption, strongly neglecting its grammatical aspect, which is one of the most important aspects of image captioning. From a pragmatic perspective, several works have analyzed the embedding space of CLIP and consistently find that it excels in aligning object categories with images using a bag-of-words approach. This results in robustness against word swapping, rather than mere repetition of identical concepts. Therefore, we introduce a novel fine-tuning methodology grounded in self-supervised learning, which comprises two distinct stages: (i) refinement of CLIP through fine-tuning conditioned on self hard-negatives sourced from the model itself post fine-tuning with CLIP-S and PAC-S; (ii) fine-tuning of the pre-trained model employing our self-discriminator as a reward model. An overview of our training strategy is shown in Fig. 1.

### 3.3 Fine-tuning of Self-Discriminator

As mentioned above, the first stage involves refining the CLIP-based discriminator  $\mathcal{D}_r$  through generation-aware mining of hard-negatives. Initially, we employ



**Fig. 2.** Overview of our self-discriminator approach, in which both CLIP encoders are fine-tuned with low-rank adaptation (LoRA) using additional textual negatives.

captioner models trained with CLIP-based rewards to generate these negative instances, which are then exploited to fine-tune CLIP. This process aims to condition CLIP against enforcing alignment styles particularly unsuitable for image captioning. Specifically, through fine-tuning, the goal is to modify the noisy embedding space of CLIP based on the errors obtained from the captioning model. When CLIP is employed in SCST, it results in a meager grammatical reward, despite its strong semantic robustness. For this purpose, we have generated two distinct types of negatives for each sample (*i.e.*  $Z_i = \{Z_i^1, Z_i^2\}$ ) derived from the fine-tuned captioner using SCST with rewards based on CLIP-S and PAC-S in their reference-based versions, respectively. This choice allows the model to learn not only to better align the embedding space but also to provide self-supervised reward and thus learn from its own mistakes.

To fine-tune the CLIP-based discriminator  $\mathcal{D}_r$ , we propose a simple modification to the CLIP objective (see Figure 2). In particular, given a batch of  $N$  images  $\mathcal{I} = \{I_1, \dots, I_N\}$  and  $N$  captions  $\mathcal{T} = \{T_1, \dots, T_N\}$ , we concatenate the textual negatives in such a way as to obtain  $\tilde{\mathcal{T}} = \{T_1, \dots, T_N, Z_1^1, Z_1^2, \dots, Z_N^1, Z_N^2\}$ . Next, we compute the similarity matrix  $S \in \mathbb{R}^{N \times 3N}$ . Here, the row-wise and column-wise cross-entropy losses are computed as in CLIP, with the difference that we do not compute the loss for the negative captions column-wise (as there is no matching image for a negative caption). To reduce the number of trainable parameters and save memory, we employ low-rank adaptation (LoRA) [22] during the fine-tuning phase of our CLIP-based discriminator, on all layers of both visual and textual encoders.

### 3.4 Training strategy

Once the fine-tuning of the discriminator is completed, it is employed as a reward signal to fine-tune the captioner through SCST. Our fine-tuned discriminator  $\mathcal{D}_r$  is capable of providing feedback not only on semantics but it is also sensitive to grammar and syntax. Finally, the reward perceived by our agent is conditioned not only on the generated text but also on the input image and implicitly on the errors that our model would have generated without any correction and modification of the embedding space.

## 4 Experimental Evaluation

### 4.1 Datasets and Evaluation Protocol

We train our model on the COCO dataset [30] which contains around 120k images each associated to five different captions, using the splits defined in [24] where 5,000 images are used for validation, another 5,000 for testing, and the remainder for training. We then evaluate the effectiveness of our solution on the COCO test set and on the validation set of different image captioning datasets, namely nocaps [1], VizWiz [20], and CC3M [46].

To evaluate our results, we employ both standard captioning metrics, such as BLEU [37], METEOR [5], ROUGE [29], CIDEr [50], and SPICE [3], and more recent learning-based scores like CLIP-Score [21] and PAC-Score [44] in their reference-free and reference-based versions. In addition, we employ a novel measure to evaluate the grammatical correctness of the generated captions. Specifically, we define Rep- $n$  with  $n = 1, 2, 3, 4$  as the average number of  $n$ -grams which are repeated in the generated captions.

### 4.2 Implementation Details

**CLIP fine-tuning.** Regarding the fine-tuning of CLIP, we use ViT-B/32 as backbone for encoding both images and textual sentences, leveraging the original OpenAI implementation<sup>1</sup>. As positive examples, we exploit image-caption pairs from the COCO dataset. We use AdamW [32] as optimizer with a learning rate set to  $1 \cdot 10^{-4}$  and a batch size of 256. Additionally, to reduce the number of trainable parameters and make fine-tuning more efficient, we employ LoRA [22] with a rank equal to 8.

**Architecture.** As our captioning model, we employ a standard encoder-decoder Transformer with 3 layers in both encoder and decoder, a hidden size of 512, and 8 attention heads. To encode input images, we use different CLIP-based backbones, such as RN50, ViT-B/32, and ViT-L/14. To implement our model, we employ the Hugging Face library [53].

<sup>1</sup> <https://github.com/openai/CLIP>

**Training details.** We first pre-train the model with the classical cross-entropy loss for sentence generation. Next, we optimize our model using different rewards based on unsupervised and supervised metrics (*i.e.* our Self-Cap strategy, both CLIP-Score [21] and PAC-Score [44], and the CIDEr score). During cross-entropy pre-training, we train our network with the Adam optimizer [25], a batch size of 1,024, and for up to 20,000 steps. During this phase, we linearly warmup for 1,000 steps, then keep a constant learning rate of  $2.5 \cdot 10^{-4}$  until 10,000 steps, then sub-linearly decrease until 15,000 steps to  $10^{-5}$  and keep the value constant until the end of the training. For the second stage, we further optimize our model with  $1 \cdot 10^{-6}$  as learning rate using a batch size of 32. During caption generation, we employ a beam size equal to 5.

**Table 1.** Comparison between different reward signals in terms of supervised, unsupervised, and grammar-based metrics. Results are reported on the COCO test set.

Backbone	Reward	Supervised $\uparrow$					Unsupervised $\uparrow$			Grammar $\downarrow$				
		B-4	M	R	C	S	RefCLIP-S	RefPAC-S	CLIP-S	PAC-S	Rep-1	Rep-2	Rep-3	Rep-4
-	-	32.8	28.1	55.0	109.8	20.3	0.796	0.853	0.743	0.817	1.516	0.108	0.022	0.009
-	CIDEr	39.7	29.2	58.3	126.8	21.2	0.797	0.855	0.739	0.817	1.384	0.05	0.008	0.005
-	CLIP-S	14.3	24.7	34.9	3.1	21.2	0.765	0.830	0.804	0.837	11.762	5.168	2.809	1.518
-	PAC-S	18.5	26.5	42.2	32.2	21.7	0.785	0.849	0.799	<b>0.860</b>	5.453	1.588	0.645	0.288
-	CLIP-S [14]	6.3	19.7	29.5	11.2	12.3	0.786	0.823	<b>0.843</b>	0.837	5.619	1.541	0.466	0.151
-	CLIP-S+Gr [14]	16.9	25.9	45.6	71.2	19.6	<b>0.792</b>	0.849	0.779	0.839	<b>1.536</b>	<b>0.097</b>	<b>0.015</b>	<b>0.003</b>
<b>*RN50</b>	<b>Self-Cap</b>	<b>20.8</b>	<b>26.8</b>	<b>48.2</b>	<b>72.0</b>	<b>21.8</b>	<b>0.792</b>	<b>0.851</b>	0.780	0.844	2.706	0.495	0.153	0.049
-	-	33.1	28.2	55.4	112.4	20.5	0.804	0.861	0.755	0.830	1.468	0.091	0.017	0.005
-	CIDEr	39.4	29.5	58.3	129.0	22.2	0.809	0.866	0.757	0.833	1.360	0.055	0.006	0.001
-	CLIP-S	11.4	23.1	31.2	1.1	18.5	0.778	0.830	<b>0.851</b>	0.846	11.166	3.566	1.232	0.395
-	PAC-S	20.3	27.1	44.1	40.7	22.4	0.796	0.858	0.810	<b>0.870</b>	5.078	1.443	0.584	0.260
<b>*ViT-B/32</b>	<b>Self-Cap</b>	<b>23.6</b>	<b>27.3</b>	<b>49.3</b>	<b>81.4</b>	<b>22.9</b>	<b>0.808</b>	<b>0.862</b>	0.800	0.861	<b>2.626</b>	<b>0.483</b>	<b>0.156</b>	<b>0.063</b>
-	-	37.3	30.4	58.1	126.6	23.3	0.811	0.868	0.758	0.831	1.402	0.062	0.007	0.002
-	CIDEr	43.6	30.8	61.0	143.3	23.2	0.809	0.866	0.750	0.826	0.239	0.498	0.616	0.349
-	CLIP-S	10.2	23.0	30.3	1.1	15.3	0.793	0.827	<b>0.865</b>	0.834	8.788	2.113	0.716	0.248
-	PAC-S	22.3	28.4	46.2	51.1	24.6	0.801	0.861	0.805	<b>0.862</b>	4.612	1.199	0.479	0.206
<b>*ViT-L/14</b>	<b>Self-Cap</b>	<b>22.6</b>	<b>28.4</b>	<b>45.0</b>	<b>28.2</b>	<b>24.7</b>	<b>0.809</b>	<b>0.864</b>	0.787	0.853	<b>2.216</b>	<b>0.376</b>	<b>0.118</b>	<b>0.039</b>

### 4.3 Experimental Results

**Results on COCO test set.** We start by comparing our solution against other CLIP-based rewards (*i.e.* CLIP-S and PAC-S) using different visual backbones to encode input images. Results are reported in Table 1 in terms of supervised, unsupervised, and grammar-based metrics. For completeness, we also include the results of the model trained after cross-entropy loss and using a standard CIDEr score as reward. In all experiments, we employ the same Transformer-based architecture with three layers in both the encoder and decoder. Regarding a comparison with previous works, it is important to note that the only work within the same settings is proposed by Cho *et al.* [14] which however only

**Table 2.** Descriptiveness analysis of generated captions in terms of unsupervised scores and retrieval-based metrics. Results are reported on the COCO test set.

Backbone	Strategy	Unsupervised		Recall			
		CLIP-S	PAC-S	R@1	R@5	R@10	MRR
	XE	0.743	0.817	21.2	44.2	57.6	31.2
	SCST (CIDEr)	0.739	0.817	19.8	43.4	55.7	29.8
<b>*RN50</b>	<b>Self-Cap</b>	<b>0.780</b>	<b>0.844</b>	<b>37.7</b>	<b>67.3</b>	<b>78.6</b>	<b>50.3</b>
	XE	0.755	0.830	24.8	50.8	62.8	35.7
	SCST (CIDEr)	0.757	0.833	25.7	51.7	64.4	36.7
<b>*ViT-B/32</b>	<b>Self-Cap</b>	<b>0.800</b>	<b>0.861</b>	<b>47.1</b>	<b>74.6</b>	<b>84.9</b>	<b>58.9</b>
	XE	0.758	0.831	27.7	52.6	64.2	38.5
	SCST (CIDEr)	0.750	0.826	23.9	49.8	61.6	34.9
<b>*ViT-L/14</b>	<b>Self-Cap</b>	<b>0.787</b>	<b>0.853</b>	<b>44.7</b>	<b>77.1</b>	<b>82.6</b>	<b>56.5</b>

adopts CLIP RN50 backbone as visual encoder. Specifically, two variants both optimized using CLIP-S are proposed, where the former only employs CLIP-S as reward while the latter combines CLIP-S with a grammar-based reward.

From the results, we can notice that adopting a reward relying on CLIP-based models significantly alters the performance of the model, leading to word repetitions and a lack of logical or grammatical structure within the caption. Indeed, within a few steps, the model appears to hack the metric by finding alternative ways to boost the semantics and consequently the value of the metric itself (*i.e.* CLIP-S or PAC-S), completely disregarding the syntactic structure of the caption. In particular, considering the results of our proposal (*i.e.* Self-Cap) with ViT-B/32 as visual backbone, it can be seen that our reward strategy can significantly improve the results on standard supervised metrics (*e.g.* 81.4 CIDEr points compared to 40.7 and 1.1 achieved with PAC-S and CLIP-S rewards respectively). This demonstrates the effectiveness of Self-Cap in better preserving the coherence of the predicted caption with the image and the ability to generate “human-like” and thus structurally correct captions. As expected, directly optimizing a specific metric leads to the best results on that metric, as showed by the results of the models trained with CLIP-S or PAC-S as reward. Nonetheless, this is not confirmed on the reference-based versions of CLIP-S and PAC-S for which Self-Cap achieves the best performance according to all employed backbones, further confirming a better correlation with human-written captions.

To further clarify the problems associated with unsupervised metrics when used as rewards, we also report the average number of repeated  $n$ -grams for each caption (*i.e.* Rep- $n$  with  $n = 1, 2, 3, 4$ ). Notably, Self-Cap significantly reduces the number of repetitions within the generated sentences, decreasing the 1-gram repetitions from 11.166 and 5.078 respectively using CLIP-S and PAC-S to 2.626, always when employing visual features from ViT-B/32. These results are confirmed also considering a larger number of  $n$ -grams and across all considered

visual backbones, further demonstrating the effectiveness of our training strategy in reducing the grammatical incorrectness of captions generated by captioners optimized using standard CLIP-based rewards.

When instead comparing our model with the one proposed in [14] using RN50 visual features, we can notice that the model optimized only with CLIP-S version yields a high value of CLIP-S, while totally degrading the reference-free metrics (*i.e.* 11.2 CIDEr points with respect to 72.0 of Self-Cap) and producing numerous repetitions (*i.e.* 5.619 and 1.541 of Rep-1 and Rep-2 compared to 2.706 and 0.495 of our approach). The scenario is different when considering the second variant, which is optimized with a combination of CLIP-S and a grammar-based reward. Specifically, while Self-Cap still achieves higher results in terms of all supervised metrics, it presents slightly higher values of repetitions. Nevertheless, it is noteworthy that Self-Cap does not exploit any explicit grammatical reward, as it is learned directly within the embedding space of the discriminator itself during the refinement process.

**Table 3.** Ablation study on COCO test set, using different negative textual sentences and CLIP ViT-B/32 as image encoder.

Negatives			Supervised					Unsupervised			
Manual	CLIP-S	PAC-S	B-4	M	R	C	S	RefCLIP-S	RefPAC-S	CLIP-S	PAC-S
✓			19.7	27.4	44.0	41.2	<b>22.3</b>	0.799	0.856	<b>0.812</b>	<b>0.865</b>
	✓		21.6	<b>27.5</b>	46.2	57.3	22.3	0.801	0.858	0.808	0.865
		✓	23.1	27.4	48.5	78.9	21.9	0.805	0.861	0.803	0.864
✓		✓	21.3	27.1	47.5	70.0	21.8	0.807	0.862	0.798	0.861
✓	✓	✓	21.0	27.3	46.0	60.4	21.7	<b>0.808</b>	<b>0.862</b>	0.802	0.862
	✓	✓	<b>23.6</b>	27.3	<b>49.3</b>	<b>81.4</b>	21.9	<b>0.808</b>	<b>0.862</b>	0.800	0.861

**Analysis on the descriptiveness of generated captions.** To effectively compare the captions generated by Self-Cap with those generated by a captioning model trained with a standard training paradigm (*i.e.* cross-entropy loss followed by SCST with CIDEr reward), we complement the results shown in Table 1 with retrieval-based metrics reported in Table 2. Retrieval-based metrics are generally used to measure the discriminative degree of the generated captions, which is usually a viable strategy to estimate their descriptiveness and semantic richness.

In particular, following recent works [11, 26], we measure the quality of generated captions in distinguishing images in a dataset and compute the percentage of the times the image corresponding to each generated caption is retrieved among the first  $k$  retrieved items. This is done by ranking the images in terms of CLIP similarity between visual and textual embeddings, using the CLIP ViT-B/32 model, and computing recall at  $K$  with  $k = 1, 5, 10$ . We also compute the mean reciprocal rank (MRR) for each generated caption: higher MRR scores indicate that captions are more discriminative and therefore usually more detailed.



Notably, Self-Cap can significantly increase the results obtained with a standard training paradigm (*i.e.* 24.8 and 25.7 achieved by XE and SCST (CIDEr) in terms of R@1 vs. 47.1 achieved by Self-Cap with ViT-B/32), highlighting a higher degree of descriptiveness in generated captions.

**Ablation study on negative examples.** As mentioned in Sec. 3, to compute the reward during the RL-based optimization, we employ a CLIP-based discriminator fine-tuned using a combination of self-generated negative samples obtained by two different captioners, one trained with CLIP-S reward and the other trained with PAC-S reward. In Table 3, we evaluate the effectiveness of the chosen negative samples. In particular, we consider negative samples generated by a single captioning model (*i.e.* either trained with CLIP-S or PAC-S) and manually-constructed negative samples, or a combination of them. When generating manual negatives, we consider the failure cases typically produced by a captioner fine-tuned with CLIP-based rewards: (i) premature termination of captions (*e.g.* “a man playing with a cat in”); (ii) redundancy of the final term (*e.g.* “a man with an umbrella in the background background background”); and (iii) duplication of concepts within captions (*e.g.* “a cat in the garden and a cat in the garden”). We therefore manually corrupt COCO captions either manually repeating or removing one or more random words, performing a random swap of two words, or substituting one word with a randomly selected word from the entire vocabulary of the COCO dataset.

**Table 4.** Out-of-domain performance analysis on nocaps, VizWiz, and CC3M validation sets in terms of supervised and unsupervised metrics.

Backbone	Reward	nocaps				VizWiz				CC3M									
		B-4R	C	S	CLIP-SPAC-S B-4 R	C	S	CLIP-SPAC-S B-4R	C	S	CLIP-SPAC-S								
	CLIP-S	3.7	23.2	4.6	12.9	0.738	0.799	8.70	29.8	6.7	8.8	0.667	0.78	1.0	13.9	4.3	6.5	0.678	0.78
	PAC-S	4.0	25.3	20.9	<b>14.1</b>	<b>0.741</b>	<b>0.850</b>	9.22	31.6	13.0	10.3	<b>0.688</b>	<b>0.816</b>	0.8	12.4	5.8	6.5	<b>0.699</b>	<b>0.814</b>
<b>*RN50</b>	<b>Self-Cap</b>	<b>4.9</b>	<b>27.1</b>	<b>30.4</b>	13.9	0.737	0.844	<b>10.1</b>	<b>35.4</b>	<b>19.7</b>	8.1	0.667	0.795	<b>1.2</b>	<b>14.9</b>	<b>15.9</b>	<b>7.7</b>	0.686	0.798
	CLIP-S	4.0	27.1	9.8	13.2	<b>0.754</b>	0.810	5.5	23.8	1.3	8.5	<b>0.737</b>	0.814	0.8	11.4	0.6	6.0	<b>0.718</b>	0.784
	PAC-S	5.2	28.5	35.7	<b>16.2</b>	0.750	<b>0.854</b>	11.0	34.3	20.1	<b>9.8</b>	0.715	<b>0.837</b>	1.2	14.1	9.8	7.6	0.698	<b>0.809</b>
<b>*ViT-B/32</b>	<b>Self-Cap</b>	<b>6.2</b>	<b>29.8</b>	<b>46.3</b>	16.0	0.751	<b>0.854</b>	<b>13.0</b>	<b>37.8</b>	<b>27.0</b>	9.1	0.702	0.828	<b>1.3</b>	<b>15.2</b>	<b>19.4</b>	<b>8.5</b>	0.688	0.803
	CLIP-S	5.2	28.9	10.2	17.3	<b>0.750</b>	0.819	4.1	21.8	1.2	7.0	<b>0.766</b>	0.775	0.6	10.2	0.6	4.4	<b>0.747</b>	0.765
	PAC-S	5.7	30.0	44.8	<b>18.1</b>	0.746	<b>0.850</b>	11.2	36.0	26.8	<b>12.2</b>	0.701	<b>0.820</b>	1.4	15.1	13.2	8.6	0.701	<b>0.811</b>
<b>*ViT-L/14</b>	<b>Self-Cap</b>	<b>6.9</b>	<b>31.3</b>	<b>62.8</b>	<b>18.1</b>	0.742	0.839	<b>11.4</b>	<b>37.4</b>	<b>28.5</b>	10.2	0.690	0.809	<b>1.6</b>	<b>16.7</b>	<b>21.9</b>	<b>9.6</b>	0.696	0.809

As it can be seen, the best results are obtained using a combination of negative samples deriving from the combination of CLIP-S and PAC-S, which achieves significantly higher CIDEr values compared to the manually created negatives (*i.e.* 81.4 vs. 41.2) and all other alternatives. Overall, the use of manual negatives does not prove effective also when used in combination with other considered negative samples, leading to performance degradation on all supervised metrics.

**Out-of-domain evaluation.** To assess the out-of-domain capabilities of our model, we evaluated Self-Cap on three distinct datasets, namely nocaps [1],

CC3M [46], and VizWiz [20]. While nocaps is specifically tailored for the novel object captioning task encompassing object classes absent in COCO, CC3M and VizWiz respectively comprises images sourced from the web and captured by visually impaired people. Except for captions from CC3M which are automatically generated, all other datasets are composed of manually-curated textual sentences. Table 4 shows the results obtained using three different backbones, comparing our approach with models fine-tuned using CLIP-S and PAC-S rewards. Also in this setting, Self-Cap achieves significantly higher results in terms of standard evaluation metrics, demonstrating the effectiveness and generalization capabilities of our approach even in out-of-domain scenarios.

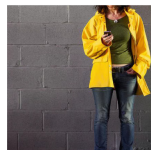
#### 4.4 Qualitative Analysis

To validate the quality of captions generated by our approach, Figure 3 shows some qualitative samples from the COCO test set. In this case, we compare captions generated by Self-Cap with those generated by a captioning model trained with PAC-S reward. As it can be seen, Self-Cap can generate more descriptive and complex captions while minimizing repetitions and grammatical errors often encountered when combining SCST with CLIP-based rewards.



**PAC-S:** A group of people sitting at a dinner table with a wine glass in the background of a boat setting of wine in the background of a restaurant.

**Self-Cap (Ours):** A group of men and women sitting around a dinner table at restaurant.



**PAC-S:** A woman in a yellow raincoat checking her cell phone against a grey wall with yellow raincoat in the background.

**Self-Cap (Ours):** A woman in a yellow jacket looking at her cell phone against a brick wall.



**PAC-S:** A young man sitting on a couch holding a wii remote control in his hand while playing video games in the living room area area.

**Self-Cap (Ours):** A young man laying in a black leather couch holding a wii remote.



**PAC-S:** The big ben clock tower towering over the city of London at night at night time with cars driving past it at night.

**Self-Cap (Ours):** The big ben clock tower towering over the city of London at night.



**PAC-S:** Two boys playing soccer in a fenced area with a green soccer ball in the background of a home area setting.

**Self-Cap (Ours):** A young boy kicking a soccer ball in a field with other players.



**PAC-S:** A herd of sheep grazing on a lush green field with a baby sheep grazing in the background of the background area area of a country.

**Self-Cap (Ours):** Three sheep are grazing in a grassy field and one is looking at the camera.

**Fig. 3.** Qualitative results on COCO sample images, comparing Self-Cap with a model trained using PAC-S as reward.

## 5 Conclusion

We present Self-Cap, a novel fine-tuning method for image captioning which entails a two-phase training procedure. It leverages a discriminator to provide feedback by learning directly from the errors of the captioner. In a setting utilizing a CLIP-based reward, the proposed solution demonstrates state-of-the-art performance in supervised metrics. Additionally, we showcase the out-of-domain capabilities of our approach on three different datasets. Self-Cap generates captions that are not only more complex and semantically richer but also yield superior grammatical accuracy compared to competitors.

**Acknowledgements.** We acknowledge the CINECA award under the IS CRA initiative, for the availability of high-performance computing resources and support. This work has been conducted under a research grant co-funded by Altilia s.r.l. and supported by the PRIN 2022 project “MUSMA” (CUP G53D23002930006) and by the PRIN 2022-PNRR project “MUCES” (CUP E53D23016290001), both funded by EU - Next-Generation EU - M4 C2 I1.1.

## References

1. Agrawal, H., Desai, K., Chen, X., Jain, R., Batra, D., Parikh, D., Lee, S., Anderson, P.: nocaps: novel object captioning at scale. In: ICCV (2019)
2. Anderson, P., Fernando, B., Johnson, M., Gould, S.: SPICE: Semantic Propositional Image Caption Evaluation. In: ECCV (2016)
3. Anderson, P., Fernando, B., Johnson, M., Gould, S.: SPICE: Semantic Propositional Image Caption Evaluation. In: ECCV (2016)
4. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR (2018)
5. Banerjee, S., Lavie, A.: METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: ACL Workshops (2005)
6. Barraco, M., Sarto, S., Cornia, M., Baraldi, L., Cucchiara, R.: With a Little Help from your own Past: Prototypical Memory Networks for Image Captioning. In: ICCV (2023)
7. Bolelli, F., Borghi, G., Grana, C.: XDOCS: an Application to Index Historical Documents. In: Digital Libraries and Multimedia Archives (2018)
8. Bucciarelli, D., Moratelli, N., Cornia, M., Baraldi, L., Cucchiara, R., et al.: Personalizing Multimodal Large Language Models for Image Captioning: An Experimental Analysis. In: ECCV Workshops (2024)
9. Caffagni, D., Cocchi, F., Barsellotti, L., Moratelli, N., Sarto, S., Baraldi, L., Baraldi, L., Cornia, M., Cucchiara, R.: The Revolution of Multimodal Large Language Models: A Survey. In: ACL Findings (2024)
10. Caffagni, D., Cocchi, F., Moratelli, N., Sarto, S., Cornia, M., Baraldi, L., Cucchiara, R.: Wiki-LLaVA: Hierarchical Retrieval-Augmented Generation for Multimodal LLMs. In: CVPR Workshops (2024)
11. Chan, D.M., Myers, A., Vijayanarasimhan, S., Ross, D.A., Canny, J.: IC<sup>3</sup>: Image Captioning by Committee Consensus. In: EMNLP (2023)
12. Chen, Q., Deng, C., Wu, Q.: Learning distinct and representative modes for image captioning. In: NeurIPS (2022)

13. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%\* ChatGPT Quality (2023)
14. Cho, J., Yoon, S., Kale, A., Dernoncourt, F., Bui, T., Bansal, M.: Fine-grained image captioning with clip reward. In: NAACL (2022)
15. Cornia, M., Baraldi, L., Cucchiara, R.: Explaining Transformer-based Image Captioning Models: An Empirical Analysis. *AI Commun.* **35**(2), 111–129 (2022)
16. Cornia, M., Stefanini, M., Baraldi, L., Cucchiara, R.: Meshed-Memory Transformer for Image Captioning. In: CVPR (2020)
17. Dai, B., Lin, D.: Contrastive learning for image captioning. *NeurIPS* (2017)
18. Dessì, R., Bevilacqua, M., Gualdoni, E., Rakotonirina, N.C., Franzon, F., Baroni, M.: Cross-Domain Image Captioning with Discriminative Finetuning. In: CVPR (2023)
19. Dong, H., Li, J., Wu, B., Wang, J., Zhang, Y., Guo, H.: Benchmarking and Improving Detail Image Caption. *arXiv preprint [arXiv:2405.19092](https://arxiv.org/abs/2405.19092)* (2024)
20. Gurari, D., Zhao, Y., Zhang, M., Bhattacharya, N.: Captioning Images Taken by People Who Are Blind. In: ECCV (2020)
21. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In: EMNLP (2021)
22. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. *arXiv preprint [arXiv:2106.09685](https://arxiv.org/abs/2106.09685)* (2021)
23. Huang, L., Wang, W., Chen, J., Wei, X.Y.: Attention on Attention for Image Captioning. In: ICCV (2019)
24. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
25. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: ICLR (2015)
26. Kornblith, S., Li, L., Wang, Z., Nguyen, T.: Guiding Image Captioning Models Toward More Specific Captions. In: ICCV (2023)
27. Li, X., Tu, H., Hui, M., Wang, Z., Zhao, B., Xiao, J., Ren, S., Mei, J., Liu, Q., Zheng, H., et al.: What If We Recaption Billions of Web Images with LLaMA-3? *arXiv preprint [arXiv:2406.08478](https://arxiv.org/abs/2406.08478)* (2024)
28. Li, Y., Pan, Y., Yao, T., Mei, T.: Comprehending and ordering semantics for image captioning. In: CVPR (2022)
29. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: ACL Workshops (2004)
30. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: ECCV (2014)
31. Liu, S., Zhu, Z., Ye, N., Guadarrama, S., Murphy, K.: Improved image captioning via policy gradient optimization of spider. In: ICCV (2017)
32. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. *arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101)* (2019)
33. Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In: CVPR (2017)
34. Luo, R., Price, B., Cohen, S., Shakhnarovich, G.: Discriminability objective for training descriptive captions. In: CVPR (2018)
35. Mokady, R., Hertz, A., Bermano, A.H.: ClipCap: CLIP Prefix for Image Captioning. *arXiv preprint [arXiv:2111.09734](https://arxiv.org/abs/2111.09734)* (2021)

36. Moratelli, N., Caffagni, D., Cornia, M., Baraldi, L., Cucchiara, R.: Revisiting Image Captioning Training Paradigm via Direct CLIP-based Optimization. In: *BMVC (2024)*
37. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: *ACL (2002)*
38. Pollastri, F., Maronas, J., Bolelli, F., Ligabue, G., Paredes, R., Magistroni, R., Grana, C.: Confidence calibration for deep renal biopsy immunofluorescence image classification. In: *ICPR (2021)*
39. Pollastri, F., Parreño, M., Maroñas, J., Bolelli, F., Paredes, R., Ramos, D., Grana, C.: A deep analysis on high-resolution dermoscopic image classification. *IET Comput. Vision* **15**(7), 514–526 (2021)
40. 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: *ICML (2021)*
41. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. In: *ICLR (2016)*
42. Ren, Z., Wang, X., Zhang, N., Lv, X., Li, L.J.: Deep reinforcement learning-based image captioning with embedding reward. In: *CVPR (2017)*
43. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-Critical Sequence Training for Image Captioning. In: *CVPR (2017)*
44. Sarto, S., Barraco, M., Cornia, M., Baraldi, L., Cucchiara, R.: Positive-augmented contrastive learning for image and video captioning evaluation. In: *CVPR (2023)*
45. Sarto, S., Cornia, M., Baraldi, L., Cucchiara, R.: BRIDGE: Bridging Gaps in Image Captioning Evaluation with Stronger Visual Cues. In: *ECCV (2024)*
46. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual Captions: A Cleaned, Hypernymed. *ACL, Image Alt-text Dataset For Automatic Image Captioning*. In (2018)
47. Shen, S., Li, L.H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.W., Yao, Z., Keutzer, K.: How Much Can CLIP Benefit Vision-and-Language Tasks? In: *ICLR (2022)*
48. Socher, R., Fei-Fei, L.: Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In: *CVPR (2010)*
49. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint [arXiv:2302.13971](https://arxiv.org/abs/2302.13971)* (2023)
50. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: CIDEr: Consensus-based Image Description Evaluation. In: *CVPR (2015)*
51. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: *CVPR (2015)*
52. Wada, Y., Kaneda, K., Saito, D., Sugiura, K.: Polos: Multimodal Metric Learning from Human Feedback for Image Captioning. In: *CVPR (2024)*
53. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-Art Natural Language Processing. In: *EMNLP (2020)*
54. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: *ICML (2015)*
55. Yang, X., Tang, K., Zhang, H., Cai, J.: Auto-Encoding Scene Graphs for Image Captioning. In: *CVPR (2019)*

56. Yao, B.Z., Yang, X., Lin, L., Lee, M.W., Zhu, S.C.: I2t: Image parsing to text description. *Proceedings of the IEEE* **98**(8) (2010)
57. Yao, T., Pan, Y., Li, Y., Mei, T.: Exploring Visual Relationship for Image Captioning. In: *ECCV* (2018)
58. Yu, Y., Chung, J., Yun, H., Hessel, J., Park, J., Lu, X., Ammanabrolu, P., Zellers, R., Bras, R.L., Kim, G., Choi, Y.: Multimodal knowledge alignment with reinforcement learning. arXiv preprint [arXiv:2205.12630](https://arxiv.org/abs/2205.12630) (2022)



# A Benchmark and Chain-of-Thought Prompting Strategy for Large Multimodal Models with Multiple Image Inputs

Daoan Zhang<sup>1</sup>(✉), Junming Yang<sup>2</sup>, Hanjia Lyu<sup>1</sup>, Zijian Jin<sup>3</sup>, Yuan Yao<sup>1</sup>, Mingkai Chen<sup>4</sup>, and Jiebo Luo<sup>1</sup>

<sup>1</sup> University of Rochester, Rochester, USA

[daoan.zhang@rochester.edu](mailto:daoan.zhang@rochester.edu)

<sup>2</sup> Nanjing University of Posts and Telecommunications, Nanjing, China

<sup>3</sup> New York University, New York, USA

<sup>4</sup> Stony Brook University, Stony Brook, USA

**Abstract.** When exploring the development of Artificial General Intelligence (AGI), a critical task for these models involves interpreting and processing information from multiple image inputs. However, Large Multimodal Models (LMMs) encounter two issues in such scenarios: (1) a lack of fine-grained perception, and (2) a tendency to blend information across multiple images. To better investigate the capability of LMMs to perceive fine-grained visual details when dealing with multiple input images, we built a benchmark for evaluating LMM with multiple image inputs - **MIMU** (Muti-Image Inputs Multimodal Understanding Benchmark). The benchmark focuses on two scenarios: first, image-to-image matching (to evaluate whether LMMs can effectively reason and pair relevant images), and second, multi-image-to-text matching (to assess whether LMMs can accurately capture and summarize detailed image information). We conduct evaluations on a range of both open-source and closed-source large models, including GPT-4V, GEMINI, OPEN-FLAMINGO, and MMICL. Although GPT-4V achieves the best results in all metrics, it still has a significant gap from Human Evaluation. To enhance model performance, we further develop a Contrastive Chain-of-Thought (CoCoT) prompting approach based on multi-input multimodal models. This method requires LMMs to compare the *similarities and differences* among multiple image inputs, and then guide the models to answer detailed questions about multi-image inputs based on the identified similarities and differences. Our experimental results showcase CoCoT's proficiency in enhancing the multi-image comprehension capabilities of large multimodal models.

**Keywords:** Large Multimodal Model · Multimodal Prompting · Large Language Model

---

D. Zhang, J. Yang and H. Lyu—These authors contributed equally to this work.

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

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15318, pp. 226–241, 2025.

[https://doi.org/10.1007/978-3-031-78456-9\\_15](https://doi.org/10.1007/978-3-031-78456-9_15)

# 1 Introduction

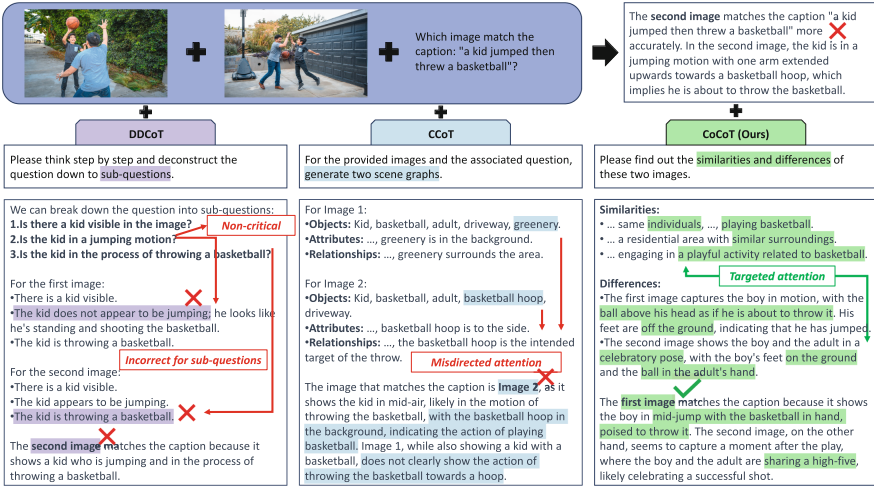
Recent advancements in Large Language Models (LLMs) [3, 4, 19, 24, 32] have sparked optimism in the pursuit of Artificial General Intelligence (AGI). Given the pivotal role of vision in human information acquisition, its integration is crucial for AGI’s perceptual capabilities. To bridge the gap between textual and visual modalities, researchers are experimenting with aligning language with vision [7, 13, 21, 26] and directly encoding visual inputs into discrete tokens [5, 6]. These efforts have demonstrated the substantial potential of large multimodal models in processing multimodal content. However, they still fall short of human-like perception of the world [14, 16, 29]. One significant challenge is understanding the *relationship between multiple image inputs*. Language-based descriptions of relationships and interactions within and across images can become challenging, necessitating explanations of both individual elements and their spatial and contextual ties. Another hurdle is the *loss of image detail* when using natural language, a medium less precise than visual data. Complex visual information, such as subtle lighting shifts or intricate patterns, often requires comprehensive verbal description. These issues result in LMM’s inability to accurately perceive information in images when dealing with multiple image inputs, especially in cases where the multiple image inputs are quite similar.

To better evaluate the capabilities of LMMs with multiple image inputs, we introduce **MIMU**: a comprehensive benchmark designed for the understanding capability of LMMs under multiple image inputs. Our benchmark includes two general scenarios: (1) image-to-image matching and (2) multi-image-to-text matching, where image-to-image matching primarily focuses on whether LMMs can perform deep reasoning on the information in images and build relationship between multiple image inputs to match them accordingly. Meanwhile, multi-image-to-text matching requires LMMs to find the subtle differences between two images and match them with text, thus can test whether LMMs lose image details during encoding. We evaluate four different large models capable of handling multiple image inputs, such as GPT-4V, MMICL, *etc.* We find that **MIMU** faces significant challenges; for example, GPT-4V achieves an accuracy of 85.30% in the image-to-image matching task, but this is still far from the human accuracy of 98.60%, indicating substantial room for improvement.

In this case, we have further developed a multimodal prompting strategy called Contrastive Chain-of-Thought (CoCoT) to enhance LMMs’ performance in multi-image tasks. CoCoT prompts LMMs to discern and articulate the **similarities and differences** among various inputs, laying the groundwork for answering detailed, multi-image-based questions (Fig. 1). Compared to previous multimodal prompting strategies [17, 36], This method sharpens the models’ focus, particularly on the distinctions between inputs, ensuring comprehensive capture of nuanced, question-relevant information during summarization. We rigorously evaluate CoCoT on the **MIMU** benchmark and our method has shown improvements across a variety of models.

To summarize, our main contributions are:





**Fig. 1.** Comparison between different multimodal prompting strategies. The unique components in each prompting strategy’s corresponding response are highlighted in varied colors. Note that GPT-4V is used in this example.

- We establish a benchmark-MIMU for LMM with multiple image inputs, comprising two scenarios: (1) image-to-image matching and (2) multi-image-to-text matching and find that most current models do *not* perform well on MIMU.
- To address the issues with existing methods, we propose a novel Contrastive Chain-of-Thought (CoCoT) prompting strategy to enhance models’ understanding of the relationships between multiple image inputs.
- Our proposed method produces significant improvement for both open-source and closed-source models on MIMU.

## 2 Related Work

**Large Multimodal Models.** Inspired by the advancements of LLMs (*e.g.*, LLAMA [24]), LMMs offer a promising way towards AGI with multimodal information. These models blend the textual reasoning prowess of LLMs with the image and video comprehension of Vision-and-Language models. This fusion enables LMMs to handle complex tasks requiring both a profound understanding and expressive generation across various modalities. Several open-source LMMs like LLAVA [13] have emerged, demonstrating competence in tasks such as image captioning and visual question-answering. However, their architectural limitations restrict their understanding and reasoning to a single image. Conversely, models like OPENFLAMINGO [1], and MMICL [34] employ specialized architectures enabling the processing of multiple image features, which better mirrors real-world scenarios. Closed-source LMMs such as GPT-4V [18] and GEMINI [22]

go beyond basic object descriptions to capture the scene’s context [17], emotions [34], and relationships [23]. A common technique to enhance performance is fine-tuning, but applying similar methods to LMMs presents computation challenges [15]. To overcome this, we propose a novel approach to directly enable detailed analysis and reasoning on images without additional training data.

**LMM Benchmarks.** These performance measures span a broad array of LMMs’ specific capabilities, including adversarial robustness [35] and hallucination [2], exemplified by initiatives like POPE [12] and HaELM [25]. Comprehensive assessments have also been carried out, featuring benchmarks like LAMM [27], MMMU [30], SEED [10], MMBench [15], and MM-Vet [28]. However, these benchmarks predominantly focus on basic sensory skills, bypassing the need for evaluating details across multiple images and critical thinking. Diverging from these, MIMU starts from the perspective of input from multiple images, constructing two types of tasks based on common scenes.

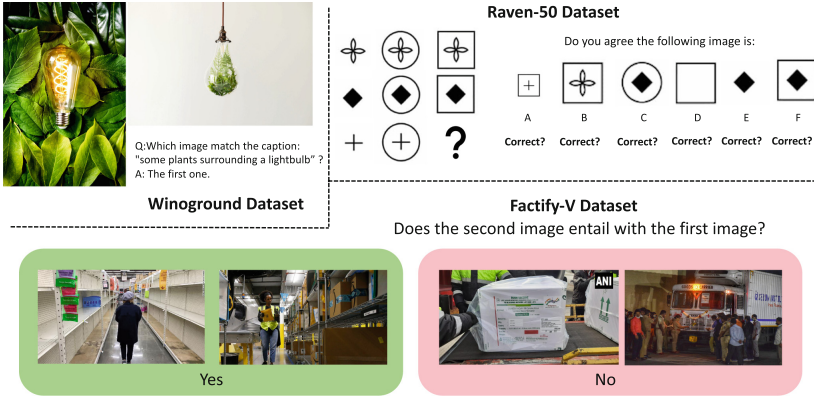
**Multimodal Prompting Methods.** Within the domain of LLMs, several language prompt methods have been established to enhance inference capabilities and ensure accurate results during prediction. These include zero-shot [9], few-shot [33], and Chain-of-Thought (CoT) [17,36] approaches. Recently, research has begun exploring the application of prompting techniques in the multimodal domain to improve the comprehension and reasoning abilities of LMMs for image data. Current multimodal prompts employed in LMMs often exhibit limitations in capturing the intricate interrelationships between visual and language information, particularly when faced with multi-image inputs. As shown in the example in Fig. 1, they are not able to identify the critical action of the boy throwing the ball. To overcome this challenge, we propose a novel prompting method that directs LMMs to extract and analyze essential information, requiring a holistic consideration of all the input images.

### 3 The MIMU Benchmark

The MIMU benchmark is built on two fine-grained multi-image tasks: (1) image-to-image matching and (2) multi-image-to-text matching. Both tasks are well-suited for assessing whether LMMs can acquire more fine-grained information from multiple image inputs.

#### 3.1 Image-to-image Matching

The image-to-image matching task employs the Raven-50 [8,31] and Factify2 [20] datasets. This task tests the models’ ability to identify and interpret visual details, requiring them to determine the degree of match between different images. The Raven-50 [8,31] test is a common tool for assessing the nonverbal reasoning capabilities of LMMs. This test demands both visual acuity and logical reasoning to decipher the connections between images. In each scenario, participants are presented with either 3 or 8 images as inputs, alongside 6 potential answer images, each with a distinct solution. The goal is to correctly



**Fig. 2.** Sampled questions from the Raven-50, Factify-V, and Winoground datasets.

identify the appropriate image. Example questions are shown in Fig. 2. Note that the evaluation metric for OPENFLAMINGO and MMICL on Raven-50 dataset is to calculate the logits of the output for each image pair; while for GPT-4V and GEMINI, we directly let the model choose the correct result and calculate the accuracy.

The Factify2 [20] dataset features 35,000 data pairs for training, and 7,500 pairs each for validation and testing. Every data pair includes a claim and a corresponding document, both of which are made up of an image, text, and OCR-generated text from the image. These pairs are categorized into one of five labels: “support multimodal”, “support text”, “refute”, “insufficient multimodal”, or “insufficient text”. Specifically, we randomly sample 500 cases in the test set, 100 for each of the 5 categories. We only use the images in the dataset in our experiments where the labels are reorganized into “support image” and “refute”. The generated subset is called Factify-V. Example questions are shown in Fig. 2. The task involves prompting the model to determine whether the pair of input images are contextually entailed.

### 3.2 Multi-image-to-text Matching

For the multi-image-to-text matching task, we use Winoground [23]. This task requires LMMs to effectively pair similar images with their corresponding textual descriptions, or alternatively, to align similar texts with the corresponding images. The Winoground [23] task involves matching images and captions which contains 400 groups of image-caption pairs. Each group contains two similar image-caption pairs. This task is challenging because the captions have the same words but in different sequences. LMMs must analyze both images and texts to identify subtle differences and understand the implied references. The Winoground is chosen to test if LMMs can comprehend fine-grained image information to text. Example questions are shown in Fig. 2. There are two tasks in

the Winoground dataset: 1) given two images, the model is required to find out which image can match the given caption; 2) given two pieces of text, the model is required to find out which text can match the given image.

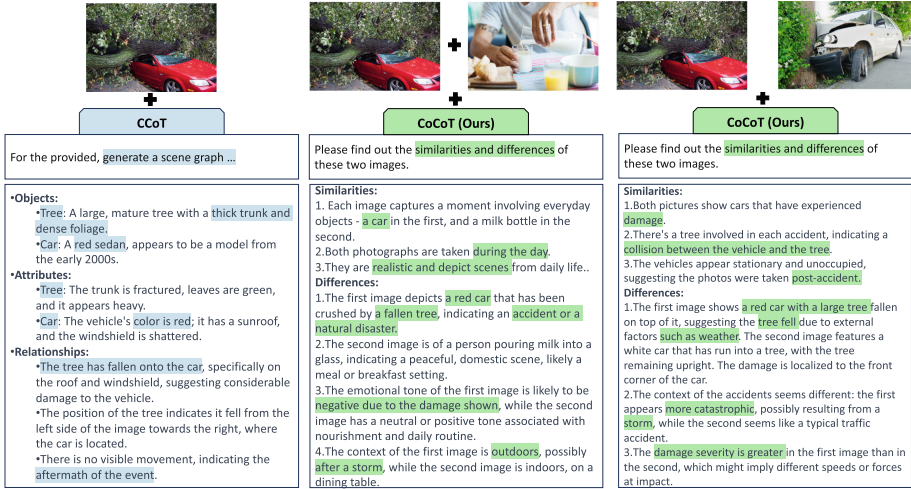
## 4 Contrastive Chain-of-Thought

### 4.1 Motivation Analysis

Traditional CoT-based prompting methods for LMMs can be categorized into two types. The first type is based on text understanding, such as DDCoT (*i.e.*, Duty-Distinct Chain-of-Thought) [36], which decomposes a question into sub-questions for a step-by-step response. The second type is based on image understanding, like CCoT (*i.e.*, Compositional Chain-of-Thought) [17], which generates a scene graph of the image to provide answers. However, while processing images, the text-based CoT does not enable LMMs to directly acquire and comprehend the detailed information in images. As shown in Fig. 1, DDCoT does not enable the LMM to recognize that the kid in the second image is **not** throwing a basketball. The image-based CCoT merely extracts basic information about the main objects in the image, also overlooking significant details. As shown in Fig. 1, CCoT generates a series of scene graphs unrelated to the question. Existing CoT-based prompting methods struggle to notice the details when answering questions about images rich in detail. Therefore, an effective prompting method should enable LMMs to discern and understand the details in images, and subsequently answer questions based on this understanding.

### 4.2 Methodology

We focus on how to enable LMMs to extract more detailed information from images, especially when the images are very similar. Initially, we examine the extent to which LMMs based on CCoT can extract information from images, as illustrated in Fig. 3. GPT-4V, utilizing CCoT, is limited to identifying entities, their characteristics, and straightforward details like events and relationships between entities. Drawing inspiration from contrastive learning [11], our approach encourages LMMs to discern similarities and differences within images. We discover that these models are capable of engaging with more complex information, such as reasoning, even when there is a considerable difference in the domain between the images being compared and the original. For instance, they might deduce that an image’s scene likely follows a storm and recognize a negative emotional tone in it. When comparing similar images, focusing on the similarities and differences of images effectively highlights the contrasts, such as recognizing more severe damage in one image compared to another, or differentiating the causes of car damage between two images, thereby effectively facilitating causal reasoning. Consequently, we develop the Contrastive Chain-of-Thought prompting. As shown in Fig. 1, this approach, similarly starting from an image perspective, initially compares the similarities and differences between



**Fig. 3.** Different CoT-based methods and their performance in extracting information from images under various conditions, with GPT-4V being used in the experiments. Left: Utilizing CCoT to generate image information; Middle: CoCoT prompting between images with a big domain gap; Right: CoCoT prompting between images with a small domain gap.

various image inputs. It then directs LMMs to answer questions based on the insights gathered from such comparisons.

Specifically, for the comparison between two images, we directly allow the LMM to perform the comparison between them. For situations involving multiple-choice questions with multiple images, we combine the images in each option with the prompt information and input them together, allowing the LMM to compare the differences between the input images.

## 5 Experiments and Results

### 5.1 Experiment Setup

**Language Models.** We evaluate two open-source LMMs: OPENFLAMINGO [1] and MMICL [34], as well as two proprietary models including GPT-4V [18] and GEMINI [22]. Due to API restrictions of GPT4-V, we only evaluate the standard and CoCoT prompting for it. For the setting of generation, we use the default configuration for each model. We use beam search with beam width of 3 for OPENFLAMINGO. In the case of MMICL, the beam width is set to 8. For GEMINI, we opt for the API of *Gemini Pro Vision* under the default settings which include a temperature of 0.4, TopK set to 32, TopP at 1, and a maximum length of 4,096. For GPT-4V, we use the default settings of the web version as of December 30, 2023. Our evaluation is conducted under a zero-shot setting to assess the capability of models to generate accurate answers without fine-tuning

or few-shot demonstrations on our benchmark. For those open-source models, all experiments are conducted with NVIDIA A6000 GPUs.

**CoT Baselines.** We compare CoCoT prompting to two state-of-the-art methods in CoT-based multimodal prompting. This includes DDCoT [36] and CCoT [17]. Additionally, we benchmark CoCoT against the standard prompting baseline, which does not incorporate any CoT instructions. All the experiments are conducted under the zero-shot setting. Example prompts and answers can be found in Fig. 1.

## 5.2 Main Results

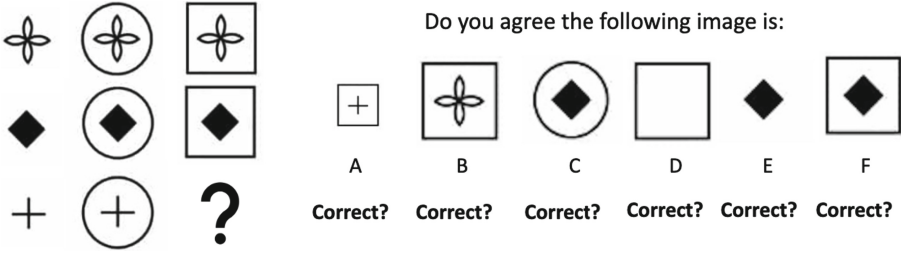
**MIMU is a challenging benchmark for LMMs.** As shown in Table 1, all LMMs have a significant gap compared to the results of human evaluation. The reasons for this situation could include several possibilities: 1) The visual encoder’s capabilities are insufficient; 2) Large models are unable to extract detailed information from the visual encoder, thereby failing in reasoning; and 3) Large models themselves are inadequate, leading to incorrect reasoning. Based on current experimental results, the visual encoder of LMMs is actually capable of recognizing some detailed information. However, due to the differences in the latent spaces between the visual encoder and the large language models, as well as the generalization issues of LLMs, LMMs are unable to fully understand images, resulting in mediocre performance across various tasks.

**Table 1.** Accuracy of LMMs in the MIMU benchmark.

	Raven-50	Factify-V	Winoground-group
Human Evaluation	98.00	99.20	85.50
OPENFLAMINGO	24.00	54.00	33.25
MMICL	22.00	64.60	37.75
GEMINI	18.00	58.00	25.00
GPT-4V	30.00	74.00	33.75

**CoCoT in Image-to-image Matching task** The task of image-to-image matching requires the model to extract information from two images simultaneously and then determine under a prompt whether the information from both images matches, as exemplified in Fig. 4. LLMs are expected to select the correct answer from the given choices. In addition to the aforementioned methods, we include another random choice baseline for comparative reference. Accuracy of LMMs with different prompting methods is shown in Table 2.

**CoCoT significantly improves LMMs’ performance in the image-to-image matching task.** Most models show improved performance when DDCoT and CCoT are employed, but the extent of improvement is not as significant as



**Fig. 4.** An example question from the image-to-image matching task, sourced from the Raven-50 [8, 31] dataset.

**Table 2.** Accuracy of LMMs employing different prompting strategies in the image-to-image matching task. The best performance within each LMM is highlighted in **bold**.

	Raven-50	Factify-V	Avg
Random Choice	17.00	50.00	42.00
Human Evaluation	98.00	99.20	98.60
OPENFLAMINGO	24.00	54.00	39.00
OPENFLAMINGO + DDCoT	24.00	58.40	41.20
OPENFLAMINGO + CCoT	24.00	63.20	43.60
OPENFLAMINGO + CoCoT	<b>26.00</b>	<b>65.00</b>	<b>45.50</b>
MMICL	22.00	64.60	43.30
MMICL + DDCoT	10.00	68.40	39.20
MMICL + CCoT	<b>26.00</b>	73.20	49.60
MMICL + CoCoT	<b>26.00</b>	<b>77.00</b>	<b>51.50</b>
GEMINI	18.00	58.00	38.00
GEMINI + DDCoT	12.00	65.40	38.70
GEMINI + CCoT	20.00	<b>80.20</b>	<b>50.10</b>
GEMINI + CoCoT	<b>22.00</b>	77.80	49.90
GPT-4V	30.00	74.00	52.00
GPT-4V + CoCoT	<b>45.00</b>	<b>80.60</b>	<b>85.30</b>

with CoCoT in most cases. Furthermore, regarding the Raven-50 dataset, which comprises non-natural images made up of various shapes, surprisingly, GEMINI emerges as the model with the poorest performance in our evaluations when GPT-4V performs the best which surpasses all models, including the open-source ones like OPENFLAMINGO and MMICL.

For the Factify-V dataset featuring natural images, GEMINI without CoT outperforms OPENFLAMINGO in similar conditions. However, when CoT is incorporated, GEMINI’s performance is almost on par with that of GPT-4V under similar conditions. This outcome differs from the results on the Raven-50

dataset, suggesting that GEMINI inherently possesses the capability to extract detailed information from natural images. Its full potential in this aspect is not fully demonstrated without the use of prompts.

In summary, our analysis of the image-to-image matching task reveals a consistent enhancement in performance across most models upon integrating various types of CoT-based prompting. This improvement underscores the ability of the visual components within LMMs to concentrate on details in terms of the task at hand. These details are subsequently processed by the LMMs for in-depth analysis, following the CoT-based prompting approach. Notably, in a majority of cases, CoCoT prompting elicits LMMs to achieve state-of-the-art performance on both natural and artificial datasets, surpassing other CoT-based strategies. This showcases the efficacy of CoCoT in guiding LMMs to accurately extract and analyze task-relevant information from images, facilitating enhanced comparative and analytical reasoning within these models.

**CoCoT in Multi-image-to-text Matching task** Compared to the image-to-image matching task, the multi-image-to-text matching task requires models to precisely extract information from images and match it with text. An example question can be found in Fig. 1. In particular, Winoground dataset is used for this task. Performance on Winoground (shown in Table 3) is assessed using three distinct metrics, each examining a different facet of the models' abilities to reason with both vision and language. The first metric, known as the **text** score, evaluates the model's capability to accurately choose the right caption when provided with an image. The second metric is the **image** score, assessing a model's ability to correctly identify the appropriate image when presented with a caption. The last metric is a composite score that integrates the first two metrics. In this **group** score, a case is considered correct if the model successfully achieves both the accurate text score and image score.

**CoCoT boosts LMMs' performance in the multi-image-to-text matching task, achieving substantial gains.** It outperforms other CoT-based methods in the majority of scenarios. This indicates that when comparing the similarities and differences of images, LMMs can better match with the text by identifying subtle differences in the input image pairs. The example in Fig. 1 also shows that methods like DDCoT and CCoT may miss key information, possibly as a result of misdirected focus.

GEMINI's performance is still the worst, indicating that although GEMINI's visual encoder can extract detailed information from the image, the model is not able to effectively summarize the information in the image, resulting in a poor match with the text. GPT-4V's performance on this task is also inferior to MMICL, indicating that GPT-4V also struggles to effectively summarize detailed information within images, particularly when the input image pairs are very similar.



**Table 3.** Accuracy of LMMs employing different prompting strategies in the multi-image-to-text matching task. The best performance within each LMM is highlighted in **bold**.

	Text	Image	Group
MTurk Human	89.50	88.50	85.50
Random Choice	25.00	25.00	16.67
OPENFLAMINGO	39.00	41.25	33.25
OPENFLAMINGO + DDCoT	47.50	47.25	39.00
OPENFLAMINGO + CCoT	42.50	27.50	20.00
OPENFLAMINGO + CoCoT	<b>58.25</b>	<b>55.25</b>	<b>41.50</b>
MMICL	46.50	40.75	37.75
MMICL + DDCoT	46.75	45.00	36.75
MMICL + CCoT	51.00	48.00	47.50
MMICL + CoCoT	<b>64.25</b>	<b>52.50</b>	<b>50.75</b>
GEMINI	30.75	26.00	25.00
GEMINI + DDCoT	<b>45.00</b>	25.00	23.75
GEMINI + CCoT	22.50	<b>33.00</b>	20.75
GEMINI + CoCoT	40.00	32.50	<b>27.75</b>
GPT-4V	54.50	42.50	37.75
GPT-4V + CoCoT	<b>58.50</b>	<b>49.50</b>	<b>44.50</b>

**Table 4.** Ablation study of the similarities and differences variants of CoCoT on the Factify-V dataset.

	MMICL	GEMINI
No prompt	64.60	58.00
+ Similarities	75.60	60.80
+ Differences	63.40	65.40
+ CoCoT	<b>77.00</b>	<b>77.80</b>

### 5.3 Ablation Study for CoCoT

CoCoT instructs LMMs to identify the similarities and differences across multiple image inputs first before providing an answer. In our ablation study, we break down the prompts into two distinct components: 1) a prompt that only requests the identification of similarities, and 2) a prompt that solely focuses on extracting the differences. As shown in Table. 4, we can observe that for GEMINI, the performance improves to some extent with the addition of either similarities or differences prompts alone, but not as much as when all prompts are included. For MMICL, adding only the differences prompts leads to a minimal decrease in performance, but the best results are achieved when both prompts are incorporated.

## 6 Discussions and Conclusions

In this study, we address the challenges faced by Large Multimodal Models in processing detailed visual information from multiple images. We first establish a benchmark for LMMs with multiple image inputs, named **MIMU**, which includes tasks under two major scenarios: (1) image-to-image matching and (2) multi-image-to-text matching. We evaluate several commonly used models capable of processing multiple input images and find that their performance on **MIMU** are far inferior to human evaluation results. Consequently, we develop the Contrastive Chain-of-Thought (CoCoT) approach, a novel multimodal prompting strategy that significantly enhances LMMs’ ability to discern fine-grained details in multi-image tasks. Our experiments with various models, demonstrate that CoCoT improves performance in image-to-image matching and multi-image-to-text tasks. This study contributes to the field of Artificial General Intelligence (AGI), offering new possibilities in areas requiring precise image interpretation. Future research should focus on refining CoCoT for more complex scenarios and integrating it with other AI technologies to further advance multimodal understanding and AGI development.

**Acknowledgement.** This work is supported by NSF award #2038208.

### A Additional Discussions of Motivation

As shown in Fig. 1, DDCoT, leaning towards a language perspective in handling images, first decouples the original question and image information into sub-questions. It then prompts LMMs to answer these sub-questions, generating sub-answers, and finally, LMMs use these sub-questions and sub-answers to respond to the original question. CCoT, more image-oriented, initially directs LMMs to generate a Scene Graph (SG) based on image information. LMMs then use the SG’s image information in conjunction with the user’s question to find an answer. Given that the above methods are not effective in catching detailed information, we focus on how to enable LMMs to extract more detailed information from images, especially when the images are very similar. To address this, CoCoT is designed to guide LMMs in identifying both the similarities and the nuanced differences between images, facilitating a more in-depth and accurate interpretation of visual content.

### B More Examples

We further provide example responses of CoCoT based on GPT-4V on different datasets, as shown in Fig. 5 and Fig. 6.

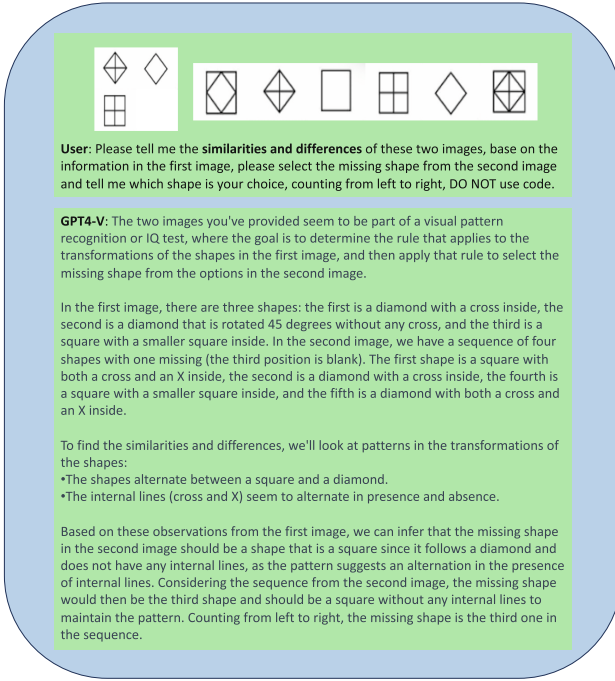


Fig. 5. An example response generated by GPT-4V via CoCoT on the Raven-50 dataset.

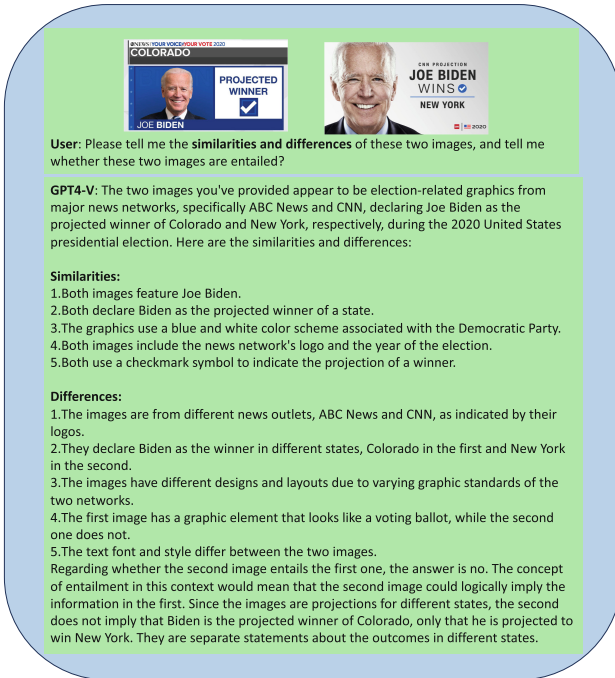


Fig. 6. An example response generated by GPT-4V via CoCoT on the Factify-V dataset.

## References

1. Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., et al.: Openflamingo: An open-source framework for training large autoregressive vision-language models. arXiv preprint [arXiv:2308.01390](https://arxiv.org/abs/2308.01390) (2023)
2. Cui, C., Zhou, Y., Yang, X., Wu, S., Zhang, L., Zou, J., Yao, H.: Holistic analysis of hallucination in gpt-4v (ision): Bias and interference challenges. arXiv preprint [arXiv:2311.03287](https://arxiv.org/abs/2311.03287) (2023)
3. Hu, Y., Ganter, T., Deilamsalehy, H., Dernoncourt, F., Foroosh, H., Liu, F.: Meetingbank: A benchmark dataset for meeting summarization. arXiv preprint [arXiv:2305.17529](https://arxiv.org/abs/2305.17529) (2023)
4. Hu, Y., Song, K., Cho, S., Wang, X., Foroosh, H., Liu, F.: Decipherpref: Analyzing influential factors in human preference judgments via gpt-4. arXiv preprint [arXiv:2305.14702](https://arxiv.org/abs/2305.14702) (2023)
5. Hu, Y., Hua, H., Yang, Z., Shi, W., Smith, N.A., Luo, J.: Promptcap: Prompt-guided task-aware image captioning. arXiv preprint [arXiv:2211.09699](https://arxiv.org/abs/2211.09699) (2022)
6. Hua, H., Li, X., Dou, D., Xu, C.Z., Luo, J.: Fine-tuning pre-trained language models with noise stability regularization. arXiv preprint [arXiv:2206.05658](https://arxiv.org/abs/2206.05658) (2022)
7. Hua, H., Shi, J., Kafle, K., Jenni, S., Zhang, D., Collomosse, J., Cohen, S., Luo, J.: Finematch: Aspect-based fine-grained image and text mismatch detection and correction. arXiv preprint [arXiv:2404.14715](https://arxiv.org/abs/2404.14715) (2024)
8. Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O.K., Liu, Q., et al.: Language is not all you need: Aligning perception with language models. arXiv preprint [arXiv:2302.14045](https://arxiv.org/abs/2302.14045) (2023)
9. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. *Adv. Neural. Inf. Process. Syst.* **35**, 22199–22213 (2022)
10. Li, B., Wang, R., Wang, G., Ge, Y., Ge, Y., Shan, Y.: Seed-bench: Benchmarking multimodal llms with generative comprehension. arXiv preprint [arXiv:2307.16125](https://arxiv.org/abs/2307.16125) (2023)
11. Li, C., Zhang, D., Huang, W., Zhang, J.: Cross contrasting feature perturbation for domain generalization. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 1327–1337 (2023)
12. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. arXiv preprint [arXiv:2305.10355](https://arxiv.org/abs/2305.10355) (2023)
13. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint [arXiv:2304.08485](https://arxiv.org/abs/2304.08485) (2023)
14. Liu, X., Liu, P., He, H.: An empirical analysis on large language models in debate evaluation. arXiv preprint [arXiv:2406.00050](https://arxiv.org/abs/2406.00050) (2024)
15. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv preprint [arXiv:2307.06281](https://arxiv.org/abs/2307.06281) (2023)
16. Lyu, H., Huang, J., Zhang, D., Yu, Y., Mou, X., Pan, J., Yang, Z., Wei, Z., Luo, J.: Gpt-4v (ision) as a social media analysis engine. arXiv preprint [arXiv:2311.07547](https://arxiv.org/abs/2311.07547) (2023)
17. Mitra, C., Huang, B., Darrell, T., Herzig, R.: Compositional chain-of-thought prompting for large multimodal models. arXiv preprint [arXiv:2311.17076](https://arxiv.org/abs/2311.17076) (2023)
18. OpenAI: GPT-4 technical report. *CoRR* **abs/2303.08774** (2023). <https://doi.org/10.48550/ARXIV.2303.08774>, <https://doi.org/10.48550/arXiv.2303.08774>

19. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. *Adv. Neural. Inf. Process. Syst.* **35**, 27730–27744 (2022)
20. Suryavardan, S., Mishra, S., Patwa, P., Chakraborty, M., Rani, A., Reganti, A., Chadha, A., Das, A., Sheth, A., Chinnakotla, M., et al.: Factify 2: A multimodal fake news and satire news dataset. *arXiv preprint [arXiv:2304.03897](https://arxiv.org/abs/2304.03897)* (2023)
21. Tang, Y., Zhang, J., Wang, X., Wang, T., Zheng, F.: Llmva-gebc: Large language model with video adapter for generic event boundary captioning. *arXiv preprint [arXiv:2306.10354](https://arxiv.org/abs/2306.10354)* (2023)
22. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: A family of highly capable multimodal models. *arXiv preprint [arXiv:2312.11805](https://arxiv.org/abs/2312.11805)* (2023)
23. Thrush, T., Jiang, R., Bartolo, M., Singh, A., Williams, A., Kiela, D., Ross, C.: Winoground: Probing vision and language models for visio-linguistic compositionality. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5238–5248 (2022)
24. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. *arXiv preprint [arXiv:2302.13971](https://arxiv.org/abs/2302.13971)* (2023)
25. Wang, J., Zhou, Y., Xu, G., Shi, P., Zhao, C., Xu, H., Ye, Q., Yan, M., Zhang, J., Zhu, J., et al.: Evaluation and analysis of hallucination in large vision-language models. *arXiv preprint [arXiv:2308.15126](https://arxiv.org/abs/2308.15126)* (2023)
26. Xie, Z., Deng, S., Liu, P., Lou, X., Xu, C., Li, D.: Characterizing anti-vaping posts for effective communication on instagram using multimodal deep learning. *Nicotine and Tobacco Research* **26**(Supplement\_1), S43–S48 (2024)
27. Yin, Z., Wang, J., Cao, J., Shi, Z., Liu, D., Li, M., Huang, X., Wang, Z., Sheng, L., Bai, L., et al.: Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *Advances in Neural Information Processing Systems* **36** (2024)
28. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mmvet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint [arXiv:2308.02490](https://arxiv.org/abs/2308.02490)* (2023)
29. Yu, Y., Du, D., Zhang, L., Luo, T.: Unbiased multi-modality guidance for image inpainting. In: *European Conference on Computer Vision*. pp. 668–684. Springer (2022)
30. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint [arXiv:2311.16502](https://arxiv.org/abs/2311.16502)* (2023)
31. Zhang, C., Gao, F., Jia, B., Zhu, Y., Zhu, S.C.: Raven: A dataset for relational and analogical visual reasoning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 5317–5327 (2019)
32. Zhang, D., Zhang, W., He, B., Zhang, J., Qin, C., Yao, J.: Dnagpt: A generalized pretrained tool for multiple dna sequence analysis tasks. *bioRxiv* pp. 2023–07 (2023)
33. Zhang, R., Hu, X., Li, B., Huang, S., Deng, H., Qiao, Y., Gao, P., Li, H.: Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 15211–15222 (2023)

34. Zhao, H., Cai, Z., Si, S., Ma, X., An, K., Chen, L., Liu, Z., Wang, S., Han, W., Chang, B.: Mmicl: Empowering vision-language model with multi-modal in-context learning. arXiv preprint [arXiv:2309.07915](https://arxiv.org/abs/2309.07915) (2023)
35. Zhao, Y., Pang, T., Du, C., Yang, X., Li, C., Cheung, N.M.M., Lin, M.: On evaluating adversarial robustness of large vision-language models. *Advances in Neural Information Processing Systems* **36** (2024)
36. Zheng, G., Yang, B., Tang, J., Zhou, H.Y., Yang, S.: Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. arXiv preprint [arXiv:2310.16436](https://arxiv.org/abs/2310.16436) (2023)



# Improving Multimodal Rumor Detection via Dynamic Graph Modeling

Xinyu Wu<sup>1</sup>, Xiaoxu Hu<sup>2</sup>, Xugong Qin<sup>1</sup>(✉), Peng Zhang<sup>1,3</sup>(✉), Gangyan Zeng<sup>1</sup>,  
Yu Guo<sup>1</sup>, Runbo Zhao<sup>1</sup>, and Xinjian Huang<sup>1</sup>

<sup>1</sup> School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

{sinyu, qinxugong, zhang\_peng, gyzen, guoyu3918, zhaorunbo, huangxinjian}@njjust.edu.cn

<sup>2</sup> National Computer Network Response Technical Team/Coordination Center Technology, Beijing, China

hxx@cert.org.cn

<sup>3</sup> Laboratory for Advanced Computing and Intelligence Engineering, Wuxi, China

**Abstract.** Due to the proliferation of rumors in social networks, automatic rumor detection has evoked increasing attention in recent years. Despite great progress achieved by exploiting multimodal features, existing works suffer from false discrimination issues due to insufficient multimodal modeling, mainly from two aspects: 1) neglect of the dynamism of social networks. 2) misaligned multimodal features. To alleviate the issues, we propose DGM, **D**ynamic **G**raph **M**odeling for rumor detection. Firstly, dynamic graph attention is devised to exploit message propagation's structural and temporal features. Secondly, we propose a modality-shared adapter to learn better multimodal representation. Thirdly, well-aligned visual-textual features are introduced to achieve better multimodality alignment and fusion, together with cross-modal attention and alignment supervision. We conduct extensive experiments on two public datasets, demonstrating the effectiveness and superiority of DGM.

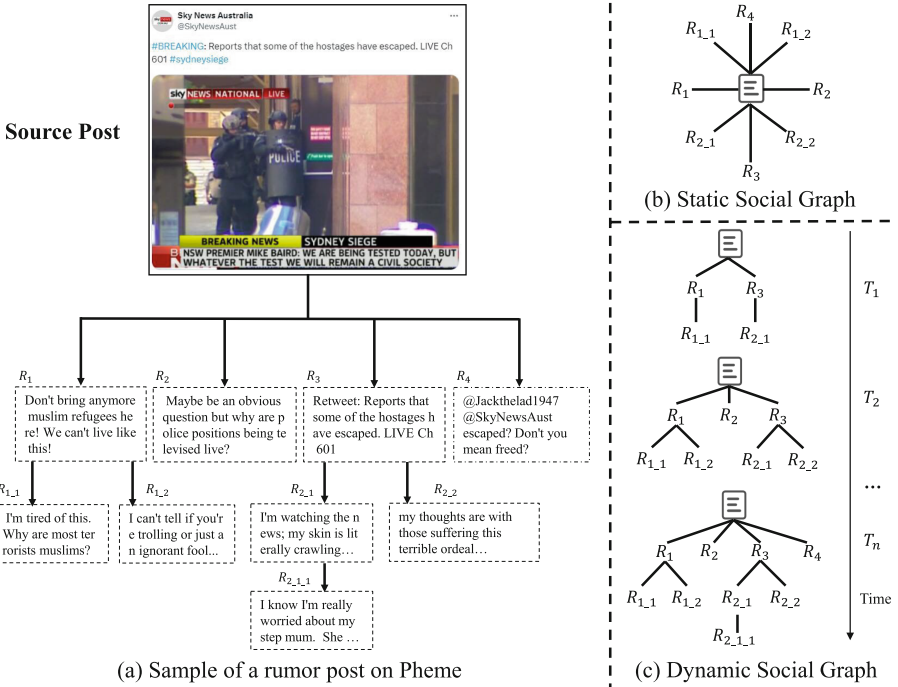
**Keywords:** Rumor detection · Multimodal learning · Dynamic graph network.

## 1 Introduction

In recent years, the rapid development of the internet has transformed social network platforms, such as Twitter and Weibo, into pivotal sources for news and interactions. In this evolving landscape, multimodal tasks [23, 24] have emerged as crucial for analyzing and understanding multimodal data. As one of the multimodal tasks, rumor detection has sparked increasing interest in recent years. The proliferation of rumors on social networks may lead to severe repercussions and sometimes even threaten public safety. For instance, a misguided claim asserting

that “5G was the cause behind the coronavirus pandemic” resulted in a 5G base station being incinerated in the UK. Given these potential consequences, developing automatic rumor detection techniques for information on social networks has become paramount.

Targeting the rumor detection task, early works mainly focus on unimodal information [4, 20]. Traditional learning models such as decision trees are employed to classify the given text or image content. Considering rumors in social networks often contain a wealth of multimodal information, recent research has attempted to integrate cross-modal features and make decisions from multiple perspectives. For example, Zhou et al. [46] compute the similarity between text and image and use the similarity to guide the cross-modal feature fusion. Another study on intra-modality and inter-modality relationships is proposed to improve performance with finer-grained features [28].



**Fig. 1.** Illustration of a rumor propagation on PHEME. (a) A multimodal post with its replies, (b) a static social graph diagram where all replies nodes are directly connected under the post node [45], and (c) a dynamic propagation diagram showcasing variations in propagation over time, including replies to the post and subsequent nested replies.

Despite promising results that have been achieved, we observe that most existing methods ignore the social context, leading to a lack of comprehensive understanding of the posts. Specifically, the social context of a source post refers



to the retweets and replies information from social users, which plays a critical role in rumor detection. To deal with it, a straightforward solution is to construct a heterogeneous graph based on this context information and utilize it as an additional model input [45]. Nevertheless, as shown in Fig.1, several limitations could be involved: 1) The heterogeneous graph is static and thus cannot reflect the dynamic evolution of posts. Yet, the timeline cues could provide strong evidence for detecting rumors. For example, the “R<sub>4</sub>” reply gives a skeptical opinion based on past comments, suggesting that the post might exhibit obvious flaws. 2) All retweets and replies nodes are simply connected under the post node, which neglects the propagation structure characteristics. To be concrete, the static social graph treats “R<sub>3</sub>” and “R<sub>2\_1</sub>” as being on the same level of the propagation chain, which misleads the learning process of rumor detection models. 3) How to effectively integrate the dynamic graph feature with other multimodal features is an open question. As the graph is structured while other modalities (text and image) are unstructured, simply concatenating them results in inferior performance.

To address the above issues, we propose a dynamic graph modeling (DGM) method for rumor detection. Specifically, we propose dynamic graph attention to generate dynamic graph representations through self-attention along structural neighborhoods and temporal dynamics. On the one hand, structural attention captures features from the local node neighborhoods in each snapshot through self-attention aggregation. On the other hand, temporal attention uses flexible weighting of historical representations to capture the evolutionary features of the graph. To learn better multimodal representation, we devise a modality-shared adapter to project different modalities into a multimodal shared semantic space. For better multimodality alignment and fusion, well-aligned visual-textual features [29] are introduced, together with cross-modal attention and alignment supervision. Integrated with these modules, DGM can produce more accurate predictions on discriminating rumors.

The main contributions of our work are summarized as follows:

- We design a dynamic graph modeling method for multimodal rumor detection. With this method, both structural and temporal information in social networks are well captured.
- A progressive modality alignment mechanism is elaborated to project the graph features to a shared multimodal space, facilitating effective interaction between different modalities.
- Extensive experiments are conducted on two public datasets, verifying the effectiveness and superiority of the proposed method.

## 2 Related Work

Considering the richness of the modality information utilized, we divide existing methods into unimodal and multimodal rumor detection.

## 2.1 Unimodal Rumor Detection

Traditional rumor detection relies on hand-crafted features from text or image content in the posts. Castillo et al. [4] first propose a learning-based method for classification. Ma et al. [26] utilize RNN to extract text representations, setting a precedent for automatic rumor detection using deep learning. To verify the relationships between emotional information and rumors, convolutional neural networks are used in CAMI [42] to extract hidden emotional features from text content. Bhattarai et al. [2] exploit lexical and semantic attributes to detect rumors. To obtain more fine-grained emotional features, FakeFlow [15] extracts affective words in the text and combines them with theme words to obtain feature representations. Zhang et al. [44] excavate the relationships between the publisher's emotion and social emotion (i.e., dual emotion) for fake news detection. For images, Jin et al. [20] assert that image content has distinct features between non-rumors and rumors. MVNN [27] learns effective visual features by combining the information of frequency and pixel domains. However, with the proliferation of multimodal content in social networks, these methods cannot capture the inconsistency from multiple modalities, causing poor generalization performance on rumor detection.

## 2.2 Multimodal Rumor Detection

With the spread of multimodal information in social networks, rumors tend to be presented multimodally. To solve the gap that existing methods focus on single mode, Hu et al. [18] construct a multimodal data set, i.e., MR2, and propose a detection method based on multimodal retrieval. To learn multimodal features, Jin et al. [19] first incorporate merging textual and visual features to enhance the accuracy of rumor detectors. SAFE [46] calculates the relevance of cross-modal features for rumor detection. A multimodal contextual attention network called HMCAN [28] is proposed to obtain the intra-modality and inter-modality relationships. Khattar et al. [21] use a bimodal variational autoencoder coupled with a binary classifier for fake news detection. Besides, to enhance the generalization of rumor detectors in new event news, EANN [36] employs an adversarial neural network, extracting event invariable features to fit new events. Zhang et al. [11] apply meta multi-task learning to detect rumors. QSAN [32] combines quantum-based text encoding with an innovative signed attention mechanism to enhance false information detection. To measure the consistency between textual and visual representations, CAFE [5] evaluates the Kullback-Leibler (KL) Divergence between the distributions of unimodal features as a cross-modal ambiguity measurement. Wu et al. [39] stack multiple co-attention layers to fuse multimodal features. To achieve more accurate multimodal alignment, Wang et al. [35] introduce cross-modal contrastive learning for fake news detection. An adaptive co-attention contrastive learning network [41] has been introduced to integrate multimodal features effectively. Chen et al. [6] analyze and identify the psycholinguistic bias in text and images, and propose a CCD framework to remove the latent data bias. However, these multimodal methods

don't consider the propagation structure features in social networks, which have been proven to be beneficial for rumor detection [40].

**Graph Modeling.** The graphical social network also reserves abundant information, which includes retweeted users and their comments. Based on it, a heterogeneous graph can be built to extract structure features. Bian et al. [3] propose a BiGCN network using GCN [22], which obtains the propagation and dispersion features via top-down and bottom-up graph convolution networks. Inspired by the graph attention network [34], GLAN [43] integrates global structure and local semantic features based on heterogeneous social networks to detect rumors. Lu et al. [25] predict fake news based on the source tweet and its propagation-based users. Wei et al. [37] rethink the reliability of latent relations by adopting a Bayesian approach. MFAN [45] simultaneously considers textual, visual, and static graph features to improve the discrimination ability of rumor detection models. Inspired by computer vision, Wu et al. [38] regard the rumor conversation thread as a color image and each node as a pixel.

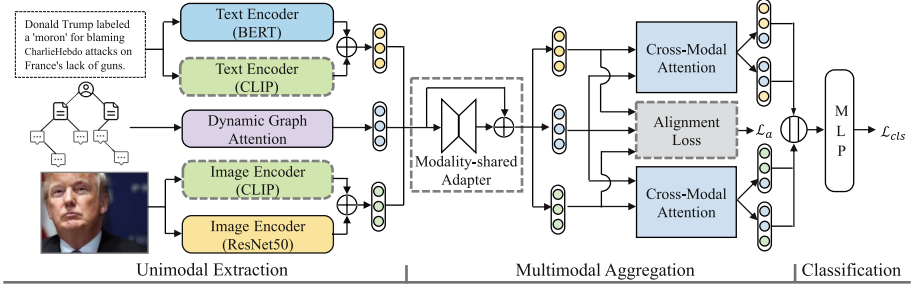
While these methods have demonstrated strong performance, few consider the dynamic nature of social networks. Different from existing methods, we propose dynamic graph modeling which jointly captures information from temporal and node dimensions as well as progressively aligns graph features to the other unstructured modalities.

### 3 Problem Definition

Let  $P = \{p_1, p_2, \dots, p_N\}$  be a set of multimodal posts with texts and images, where  $N$  is the number of posts. For each post  $p_i \in P$ ,  $p_i = \{t_i, v_i, u_i, c_i\}$ , where  $t_i$ ,  $v_i$ , and,  $u_i$  refer to the text, image, and author of the post.  $c_i = \{c_i^1, c_i^2, \dots, c_i^{N_c}\}$  signifies all comments of  $p_i$ . In social network graphs, let  $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_T\}$  be a series of static graph snapshots, where  $T$  is the number of time steps. Each snapshot  $\mathcal{G}_t = \{V, A_t, \mathcal{E}_t\}$  is a weighted undirected graph with a set of shared nodes  $V$ , links set  $\mathcal{E}_t$ , an adjacency matrix  $A_t \in \{0, 1\}^{|V| \times |V|}$  at time  $t$ . Dynamic graph learning seeks to learn the potential embeddings  $e_v^t \in \mathcal{R}^d$  for each node  $v$  at time  $t$ , which reserve both local graph structures and its temporal evolutionary behaviors. Following existing works, we define rumor detection as a binary classification task.  $\hat{y} \in \{0, 1\}$  refers to binary label, where  $\hat{y} = 0$  indicates non-rumor, and  $\hat{y} = 1$  otherwise. Given a set of posts  $P$ , our goal aims to find a function  $F : F(P) \rightarrow \hat{y}$  to predict the label.

### 4 Methodology

As shown in Figure 2, the overall pipeline can be separated into three parts. Given a multimodal input including text, image, and social graph, unimodal features are obtained by the corresponding encoders. After that, we project the unimodal features to a shared multimodal semantic space and then aggregate them to generate a rich multimodal representation for the final classification task.



**Fig. 2.** The architecture of the proposed method. We first derive the three modal representations of textual, visual, and graphical for each post on social media through unimodal extractors. Then a modality-shared adapter is utilized for unified multimodal learning. We perform cross-modal attention to obtain the enhanced features between graphic features and visual/text features. All generated cross-modal features are integrated for rumor detection.

#### 4.1 Unimodal Extraction

BERT [14] and ResNet50 [16] are utilized as the text encoder and visual encoder to generate corresponding unimodal features, which are denoted as  $m_{BERT}$  and  $m_{ResNet}$ . To reduce the burden of multimodal alignment, we also introduce pretrained CLIP encoders [29] to get well-aligned features as  $m_{CLIP-T}$  and  $m_{CLIP-V}$ . The overall textual and visual features can be obtained by weighting the outputs from BERT/ResNet50 and the CLIP encoder:

$$\begin{cases} m_t = W_{BERT} * m_{BERT} + W_{CLIP-T} * m_{CLIP-T}, \\ m_v = W_{ResNet} * m_{ResNet} + W_{CLIP-V} * m_{CLIP-V}, \end{cases} \quad (1)$$

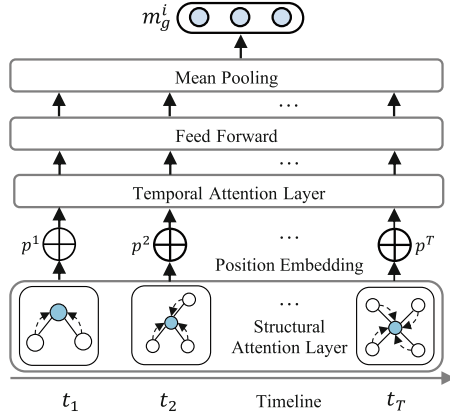
where  $W_{BERT}$ ,  $W_{CLIP-T}$ ,  $W_{ResNet}$ , and  $W_{CLIP-V}$  are learnable parameters to project the features to the same dimension.

For social graphs at different snapshots, we designate the initial node embeddings as  $X \in \mathcal{R}^{|V| \times d}$ , where  $V$  is a set of shared nodes with dimension  $d$ . The textual features are used as initial embeddings for posts and comments nodes. For user nodes, we average the post-node embeddings posted by the user as the initial embedding. The graphs feature  $m_g$  can be obtained by aggregating the information from nodes and time snapshots, detailed in section 4.2.

#### 4.2 Dynamic Graph Attention

In this part, we attempt to capture the graph structure and temporal representations at every snapshot with dynamic graph attention (DGA), illustrated in Figure 3.

**Structure Attention.** The input to this layer consists of a set of snapshots and an embedding matrix of nodes  $\{x_v^t \in \mathcal{R}^d, \forall v \in V\}$ , where  $x_v^t$  represents the node representation at time  $t$  and is initialized from  $X$ . The output is a new set



**Fig. 3.** Illustration of the proposed dynamic graph attention.

of node embeddings with local structure properties. The design of the structure layer follows GAT [34]. The operation is defined as:

$$\begin{aligned}
 x_v^t &= \sigma\left(\sum_{u \in N_v^t} \alpha_{uv} W^s x_u^t\right), & \alpha_{uv} &= \frac{\exp(e_{uv})}{\sum_{w \in N_v^t} \exp(e_{vw})}, \\
 e_{uv} &= \sigma(A_{uv}^t \alpha^T [W^s x_u^t || W^s x_v^t]) \quad \forall (u, v) \in \varepsilon_t,
 \end{aligned}
 \tag{2}$$

where  $N_v^t = \{u \in V : (u, v) \in \varepsilon_t\}$  is a set of neighbors of node  $v$  in snapshot  $\mathcal{G}_t$  at time step  $t$ ;  $\alpha_{uv}$  is a set of learnable coefficients, obtained by a softmax over the neighbors of each node in  $V$ ;  $W^s$  is shared trainable weight applied to each node;  $\alpha$  is an attention value between node  $u$  and  $v$ ;  $A_{uv}^t$  is weight of link  $(u, v)$  at time step  $t$ ;  $||$  is concatenation operation and  $\sigma(\cdot)$  is a non-linear activation function. We apply the LeakyReLU [8] function as an activation function to compute attention weights.

**Temporal Attention.** Considering the dynamicity of social networks, we design a temporal attention layer to capture the evolving behaviors of graphs further. Specifically, we capture the ordering information by using position embeddings,  $\{p^1, p^2, \dots, p^T\}$ , which encode the absolute temporal position of each snapshot. The position embeddings are combined with the output of the structural attention layer to obtain a sequence of input representations of this layer:  $\{x_v^1 + p^1, x_v^2 + p^2, \dots, x_v^T + p^T\}$ . The final layer outputs are fed into a feed-forward layer to get the final node representations  $\{e_v^1, e_v^2, \dots, e_v^T\}, \forall v \in V$  with the dimension of  $d$ .

In this layer, we apply self-attention to capture the temporal variation of the graph structure. The queries, keys, and values are transformed to a different space through linear projection matrices  $W_q, W_k, W_v$  respectively. Then the

temporal attention function is defined as:

$$\begin{aligned} e_v^t &= \beta_v(X_v W_v), & \beta_v^{ij} &= \frac{\exp(e_v^{ij})}{\sum_{k=1}^T \exp(e_v^{ik})}, \\ e_v^{ij} &= \left( \frac{((X_v W_q)(X_v W_k)^T)_{ij}}{\sqrt{d}} + M_{ij} \right), \end{aligned} \quad (3)$$

where  $\beta_v \in \mathcal{R}^{T \times T}$  is the attention weight matrix and  $M \in \mathcal{R}^{T \times T}$  is a mask matrix with  $M_{ij} \in \{0, -\infty\}$ , which ensures that future node information doesn't leakage to past. To encode the temporal information, the mask matrix  $M$  is defined as:

$$M_{ij} = \begin{cases} 0, & i \leq j \\ -\infty, & \text{otherwise.} \end{cases} \quad (4)$$

When  $M_{ij} = -\infty$ , the softmax returns a zero attention vector, which means the temporal attention layer only pays attention time step  $i$  to  $j$  [30]. We apply mean pooling to get the final dynamic graphs feature  $m_g$ , which contains structural information and temporal variation at every time step:

$$m_g = \text{MEAN}(e_v^1, e_v^2, \dots, e_v^T). \quad (5)$$

### 4.3 Multimodal Aggregation

In the multimodal aggregation stage, for better multimodal representation, we propose to maintain multimodal alignment during the aggregation process: 1) A modality-shared adapter that unifies to process the text, visual, and graph modalities to a multimodal semantic space efficiently. 2) Cross-modal attention mechanism which introduces an interaction between different modalities to generate semantic-rich multimodal representation.

**Modality-shared Adapter.** Hu et al. [17] show that the updates to the weight have a low ‘‘intrinsic rank’’ during adaptation. Besides, the pre-trained model can still learn efficiently despite a random projection to a smaller subspace [1]. Inspired by this, we propose a modality-shared adapter to learn the features of different modalities in a low-rank shared semantic space. In this part, all parameters are shared. Specifically, each modal feature is first projected to the low-rank semantic space then mapped back to the high-dimensional space, and finally summed with the original feature to get a better quality feature:

$$m'_* = W_{up} W_{down} m_* + m_*, \quad (6)$$

where  $W_{down}$  and  $W_{up}$  denote the learnable parameters in the shared linear layers, which aim to project features into different embedding spaces.  $m_*$  ( $* \in \{t, v, g\}$ ) and  $m'_*$  denote the three modal features of input and output.

**Cross-modal Attention.** In this part, attention mechanism [33] is employed to learn the interaction between different modalities. To obtain the textual-graphical enhanced feature  $m_{tg}$ , we use  $Q_g = m'_g W_q$ ,  $K_t = m'_t W_K$ ,  $V_t = m'_t W_v$

to get query, key and value matrices. Then we calculate the multi-head cross-modal attention textual-graphical feature  $m_{tg}$  as:

$$m_{tg} = \left( \parallel_{h=1}^H \text{softmax} \left( \frac{Q_g K_t^T}{\sqrt{d}} \right) V_t \right) W_{tg}^o, \quad (7)$$

where  $h$  denotes the  $h$ -th head,  $W_{tg}^o$  is the output linear transformation. Similarly, we can obtain graphical-textual feature  $m_{gt}$ , visual-graphical feature  $m_{vg}$ , and graphical-visual feature  $m_{gv}$ .

#### 4.4 Optimization

**Classification.** Given the enhanced features after cross-modal interaction, the final multimodal feature  $m$  is obtained by concatenating all cross-modal features. Then we feed the final multimodal feature  $m_i$  to a multilayer perceptron to generate the prediction  $y$ . Standard cross-entropy loss is adopted to supervise the classification task:

$$\mathcal{L}_{cls} = -\hat{y} \log(y) - (1 - \hat{y}) \log(1 - y). \quad (8)$$

**Modality Alignment.** Besides the classification supervision, we introduce a modal alignment module by enforcing dynamic graph features close to textual-visual features to refine the representation. Specifically, the textual features are summed to the visual features and then transformed to get textual-visual features  $Align_{tv}$ . The dynamic graph features are also projected into the same feature space to get  $Align_g$ :

$$\begin{aligned} Align_{tv} &= W_{tv}(m'_t + m'_v), \\ Align_g &= W_g m'_g, \end{aligned} \quad (9)$$

where  $W_{tv}$  and  $W_g$  are learnable parameters. Then the alignment loss is defined as:

$$\mathcal{L}_a = \frac{1}{d} \sum_{i=1}^d (Align_{tv} - Align_g)^2. \quad (10)$$

The overall loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_a \mathcal{L}_a, \quad (11)$$

where  $\lambda_a$  denotes the balanced factor for the alignment loss.

## 5 Experiments

### 5.1 Datasets

We evaluate our model on two real-world datasets: Weibo [31] and PHEME [47]. Weibo is collected from the biggest social media in China. PHEME contains a collection of Twitter rumors and non-rumors posted during five breaking news.

Except for the accuracy, the evaluation metrics including the precision, recall, and F1-score are weighted considering the class imbalance, which follows the official implementation of MFAN<sup>1</sup>. Each dataset contains texts, images, and social behaviors. The statistics of the two datasets are presented in Table 1.

**Table 1.** Statistics of the Weibo and PHEME datasets.

Dataset	Real	Fake	Text-Image pairs	Nodes	Edges
Weibo	877	590	1467	8505	7544
PHEME	1428	590	2018	8650	7756

## 5.2 Implementation Details

Following existing work [45], the datasets are divided into a training set, validation set, and test set in the ratio of 7:1:2. We set up 5 snapshots in both datasets. In the Weibo dataset, the number of low-rank layers in the modality-shared adapter is 16, while in the PHEME it is 64. To ensure the validity of the comment information, we retain comments with a length of more than 150 (Weibo) and 130 (PHEME).  $\lambda_a$  is set to 1 in both datasets. We use BERT to obtain text representation with the dimension of 768 (bert-base-uncased for English text and bert-base-chinese for Chinese text), and ResNet50 to extract image features with the dimension of 1000. For the CLIP model, the clip-ViT-B-32 version is utilized to derive text and image representations with the dimensions of 512. All these pre-trained models are frozen. We train our model for 20 epochs and report the best testing accuracy.

## 5.3 Experimental Results

**Comparison with state-of-the-art.** In this section, we evaluate DGM and compare it with other alternatives including graph-free and graph-based methods. Specifically, graph-free methods only take the textual and visual features of posts as input, such as EANN [36] and SAFE [46]. In contrast, graph-based methods take the social context into account which is usually constructed as a graph. As reported in Table 2, it can be seen that graph-based methods are generally better than graph-free ones, suggesting the importance of social context in rumor detection. Moreover, the proposed DGM model leads the performance among graph-based approaches. To further elaborate, all previous graph-based methods are implemented on static graphs, having difficulty capturing structure and temporal information hidden in social context. Benefiting from the dynamic graph modeling and progressive modality alignment mechanism, our DGM achieves an accuracy of 90.85% and F1-score of 90.67% on the Weibo

<sup>1</sup> <https://github.com/drivsaf/MFAN>



**Table 2.** Performance comparison of different methods on the Weibo and PHEME datasets. Acc, P, R, and F1 are short for Accuracy, Precision, Recall, and F1-score. \* indicates results reproduced by us with the official implementation. The bold (underlined) text indicates the best (second best) performances on each dataset.

Method		Weibo				PHEME			
		Acc	P	R	F1	Acc	P	R	F1
Graph-free	EANN [36]	80.96	80.19	79.68	79.87	77.13	71.39	70.07	70.44
	MVAE [21]	71.67	70.52	70.21	70.34	77.62	73.49	72.25	72.77
	QSAN [32]	71.01	71.02	67.54	67.58	75.13	69.97	65.80	66.87
	SAFE [46]	84.95	84.98	84.95	84.96	81.49	79.88	79.50	79.68
	MM-MTL[11]	-	-	-	-	82.21	78.84	85.45	82.02
	PVCG[13]	-	-	-	-	<u>88.50</u>	86.80	86.10	86.40
Graph-based	EBGCN [37]	83.14	85.46	81.76	81.45	82.99	81.31	79.29	79.82
	GLAN [43]	82.44	82.45	80.86	81.26	83.32	81.25	77.13	78.51
	KhiCL[7]	-	-	-	-	87.40	84.90	84.20	84.60
	MGIN-AG[10]	-	-	-	-	87.60	86.80	<b>88.90</b>	<u>87.40</u>
	MFAN [45]	<u>88.95</u>	<u>88.91</u>	<u>88.13</u>	<u>88.33</u>	<b>88.73</b>	87.07	85.61	86.16
	MFAN* [45]	86.10	87.77	86.10	85.49	87.27	<u>87.09</u>	87.27	87.15
	DGM	<b>90.85</b>	<b>91.02</b>	<b>90.85</b>	<b>90.67</b>	87.53	<b>87.42</b>	<u>87.53</u>	<b>87.47</b>

dataset, setting a new state-of-the-art result. Note that we do not list results on the Weibo dataset due to inconsistent dataset splits used in some SOTA methods, which may lead to unfair comparison. On the PHEME dataset, DGM still reaches better or more competitive performance compared to these methods. More specifically, as PVCG employs an additional powerful language model (T5) and MGIN-AG introduces OCR recognition to extract image embeddings at a finer granularity, they achieve a slightly higher accuracy than DGM. Nevertheless, DGM notably outperforms them in terms of F1-score. Considering the category imbalance in the PHEME dataset, the performance advantage in the F1-score can better demonstrate the superiority of our method.

To validate the impacts of key components in DGM, we perform an extensive ablation study, which is shown in Table 3. In the following analysis, we mainly focus on the accuracy metric for the sake of simplicity, as the performance of the F1 measure across different variants follows a similar trend as that in accuracy.

**Effectiveness of DGA and SF.** We start with training a model without DGA (#1), in which only textual and visual content are utilized. As can be seen, compared to DGM (#8), direct accuracy decreases of 4.07% and 4.15% on Weibo and PHEME datasets are achieved, which illustrates the importance of consideration of dynamic temporal information. Replacing the dynamic graph with a static graph (#2) results in decreases of 2.37% and 1.56% in terms of accuracy on Weibo and PHEME datasets, indicating the superiority of dynamic graph modeling. When all reply nodes are connected under the post node without structure

**Table 3.** Ablation analysis on the Weibo and PHEME datasets. Abbreviations notations: Dynamic Graph Attention (DGA), Structure Feature (SF), Temporal Index (TI), Modality-shared Adapter (MSA), Modality Alignment (MA), Static Graph (SG), Separate Adapter (SA).

#	DGA	SF	MSA	MA	Weibo		PHEME	
					Acc	F1	Acc	F1
1	✗	✗	✓	✓	86.78	85.40	83.38	79.85
2	SG	✓	✓	✓	88.48	87.63	85.97	85.21
3	✓	✗	✓	✓	90.17	90.22	85.71	84.90
4	TI	✓	✓	✓	86.10	85.58	84.94	83.76
5	✓	✓	✗	✓	86.44	86.50	85.20	85.53
5	✓	✓	SA	✓	89.15	89.10	85.71	85.53
7	✓	✓	✓	✗	88.17	88.28	85.97	84.95
8	✓	✓	✓	✓	<b>90.85</b>	<b>90.67</b>	<b>87.53</b>	<b>87.47</b>

feature (#3), the accuracy can decrease by 0.68% and 1.82%, indicating that the propagation structure feature plays an important role on both datasets and has a greater impact on the PHEME dataset. We speculate that the higher number of active users on Twitter and the more complex network structure may contribute to the ability to assess the authenticity of information.

**Impacts of dynamic temporal features.** In Table 3, we study the effect of different ways of introducing temporal information. In the naïve version, the temporal index is simply appended to the last graph node embeddings just as position embeddings do. Compared to the baseline without temporal information (#2), the naïve version (#4) even degrades performance. We suspect that dynamic temporal evolution is difficult to capture by encoding temporal indexes. In contrast, our complete version brings significant improvements.

**Effectiveness of MSA and MA.** We train variant models without the Modality-shared Adapter (#5) and with separate adapters (#6) for comparison. We can observe that the modality-shared adapter can effectively learn the multimodal representation to achieve more precise detection. Specifically, when substituting the modality-shared adapter with three separate adapters, the accuracy decreased by 1.7% and 1.82%. Due to the shared low-rank semantic modeling, the model can better learn unified semantic representation across different modalities. Besides, the comparison of experimental results #7 and #8 can also indicate that modal alignment supervision is profitable for multimodal alignment. Overall, via constructing dynamic temporal networks with propagation structure features and introducing modality-shared adapter and modality alignment, the proposed DGM achieved the best accuracy of 90.85% and 87.53% on Weibo and PHEME datasets.

**Impacts of low-rank dimensions.** Table 4 shows the effect of using different low-rank dimensions in the modality-shared adapter. Compared to the variants without MSA, adapters with different dimensions exhibit consistent improve-

ments, further demonstrating the effectiveness of the modality-shared adapter module. Specifically, we choose 16 dimensions for Weibo and 64 dimensions for PHEME as the low-rank dimensions in the adapter to achieve the best detection results.

**Table 4.** Performance comparison of variants with different low-rank dimensions in MSA on the Weibo and PHEME datasets. “*w/o MSA*” denotes a variant model without the Modality-shared Adapter.

Dimension	Weibo		PHEME	
	Acc	F1	Acc	F1
<i>w/o MSA</i>	86.44	86.50	85.20	85.53
4	88.14	88.07	86.23	85.78
8	89.83	89.89	85.45	84.67
16	<b>90.85</b>	<b>90.67</b>	85.52	84.61
32	88.81	88.87	86.49	86.22
64	90.17	90.16	<b>87.53</b>	<b>87.47</b>

#### 5.4 Validation on other benchmarks

We evaluate the generalization of our method on the FakeNewsNet dataset, which is a comprehensive dataset with diverse features in news content and social context, and it can be divided into PolitiFact and GossipCop subsets according to the data source. We report the comparison results of DGM with existing competitors in Table 5. Compared to MFAN [45], DGM has better performance on a broad range of benchmarks, indicating the effectiveness of dynamic graph attention in detecting rumors in social networks. Besides, DGM significantly outperforms other competitors and achieves new state-of-the-art on these two subsets, well demonstrating the generalization of our proposed method.

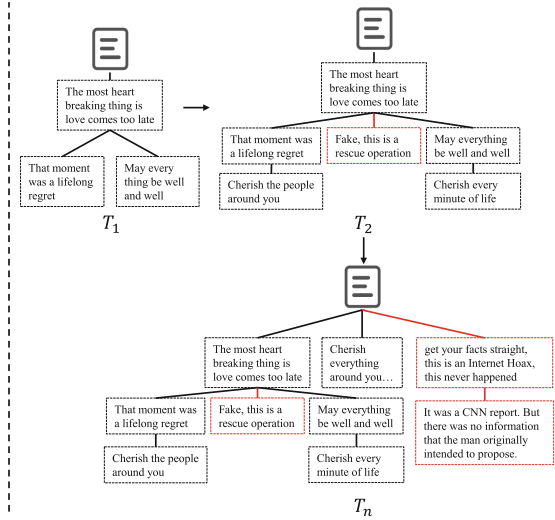
**Table 5.** Results in FakeNewsNet. The bold (underlined) text indicates the best (second best) performances on each dataset.

Method	FakeNewsNet							
	PolitiFact				GossipCop			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
MFAN [45]	78.26	78.55	78.26	76.74	74.21	74.24	74.21	74.20
MDFEND [9]	84.73	82.12	82.09	<u>82.08</u>	80.80	81.32	<u>79.70</u>	80.46
M <sup>3</sup> FEND [12]	<u>84.78</u>	<u>84.49</u>	<u>82.16</u>	81.79	<b>82.37</b>	<b>85.76</b>	79.32	<u>81.86</u>
DGM	<b>86.36</b>	<b>88.02</b>	<b>82.59</b>	<b>84.29</b>	<u>82.11</u>	<u>83.04</u>	<b>82.11</b>	<b>81.97</b>

## 5.5 Case Study

Source Post:

Someone wanted to propose to his girlfriend after running a marathon, but she was tragically killed. Cherish the people around you, you never know which one will come first tomorrow or an accident.



**Fig. 4.** A rumor case detected through capturing the temporal dynamic information of social networks.

We provide a case study from the Weibo dataset to demonstrate the effectiveness of DGM. As shown in Figure 4, a post claimed that a man was planning to propose to his girlfriend after a marathon, but she was tragically killed. At time step  $T_1$ , all comments expressed regret for this post. However, as time went by, doubts about the post emerged, which were marked with red boxes. Someone raised this was a CNN report about a man comforting and caring for an injured woman near the finish line, without any information about the man’s intention to propose. Simply connecting all replies under the post node (like Figure 1 (b)) will not notice the dynamic temporal changes of social networks and will ignore the replies that hold doubts and opposing views. Yet, our model DGM can capture the opinion variances and evolution of posts along the temporal dimension in social networks, which plays a crucial role in identifying the rumor. This case study helps to demonstrate the effectiveness of the proposed method.

## 6 Conclusion

In this work, we propose a novel multimodal rumor detection framework named DGM in which both structural and temporal information in the propagation graphs are well modeled and captured with dynamic graph attention. Additionally, we design a modality-shared adapter to learn the semantic representations of different modalities in low-rank space. The multimodal features are further

aligned and aggregated with cross-modal attention and alignment supervision. Extensive experiments on two public datasets show that our method can outperform state-of-the-art baselines for rumor detection.

**Acknowledgements.** This work is supported by the fund of the Laboratory for Advanced Computing and Intelligence Engineering.

## References


1. Aghajanyan, A., Gupta, S., Zettlemoyer, L.: Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In: *ACL*. pp. 7319–7328 (2021)
2. Bhattarai, B., Granmo, O., Jiao, L.: Explainable tsetlin machine framework for fake news detection with credibility score assessment. In: *LREC*. pp. 4894–4903 (2022)
3. Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., Huang, J.: Rumor detection on social media with bi-directional graph convolutional networks. In: *AAAI*. pp. 549–556 (2020)
4. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: *WWW*. pp. 675–684 (2011)
5. Chen, Y., Li, D., Zhang, P., Sui, J., Lv, Q., Tun, L., Shang, L.: Cross-modal ambiguity learning for multimodal fake news detection. In: *Proceedings of the ACM web conference 2022*. pp. 2897–2905 (2022)
6. Chen, Z., Hu, L., Li, W., Shao, Y., Nie, L.: Causal intervention and counterfactual reasoning for multi-modal fake news detection. In: *ACL*. pp. 627–638 (2023)
7. Liu, J., Xie, J., Zhang, F., Zhang, Q., Zha, Z.j.: Knowledge-enhanced hierarchical information correlation learning for multi-modal rumor detection. *arXiv preprint arXiv:2306.15946* (2023)
8. Maas, A.L., Hannun, A.Y., Ng, A.Y., et al.: Rectifier nonlinearities improve neural network acoustic models. In: *ICML*. vol. 30, p. 3. Atlanta, GA (2013)
9. Nan, Q., Cao, J., Zhu, Y., Wang, Y., Li, J.: Mdfend: Multi-domain fake news detection. In: *CIKM*. pp. 3343–3347 (2021)
10. Sun, T., Qian, Z., Li, P., Zhu, Q.: Graph interactive network with adaptive gradient for multi-modal rumor detection. In: *ICMR*. pp. 316–324 (2023)
11. Zhang, H., Qian, S., Fang, Q., Xu, C.: Multi-modal meta multi-task learning for social media rumor detection. *TMM* pp. 1449–1459 (2021)
12. Zhu, Y., Sheng, Q., Cao, J., Nan, Q., Shu, K., Wu, M., Wang, J., Zhuang, F.: Memory-guided multi-view multi-domain fake news detection. *TKDE* **35**(7), 7178–7191 (2022)
13. Zou, T., Qian, Z., Li, P., Zhu, Q.: Pvcg: Prompt-based vision-aware classification and generation for multi-modal rumor detection. In: *ICASSP. IEEE* (2024)
14. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: *NAACL*. pp. 4171–4186 (2019)
15. Ghanem, B., Ponzetto, S.P., Rosso, P., Rangel, F.: Fakeflow: Fake news detection by modeling the flow of affective information. In: *EACL*. pp. 679–689 (2021)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR*. pp. 770–778 (2016)
17. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: *ICLR* (2022)

18. Hu, X., Guo, Z., Chen, J., Wen, L., Yu, P.S.: Mr2: A benchmark for multimodal retrieval-augmented rumor detection in social media. In: SIGIR. pp. 2901–2912 (2023)
19. Jin, Z., Cao, J., Guo, H., Zhang, Y., Luo, J.: Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In: ACM MM. pp. 795–816 (2017)
20. Jin, Z., Cao, J., Zhang, Y., Zhou, J., Tian, Q.: Novel visual and statistical image features for microblogs news verification. TMM pp. 598–608 (2016)
21. Khattar, D., Goud, J.S., Gupta, M., Varma, V.: MVAE: Multimodal variational autoencoder for fake news detection. In: WWW. pp. 2915–2921 (2019)
22. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
23. Li, W., Ma, Z., Deng, L.J., Wang, P., Shi, J., Fan, X.: Reservoir computing transformer for image-text retrieval. In: ACM MM. pp. 5605–5613 (2023)
24. Li, W., Wang, P., Xiong, R., Fan, X.: Spiking tucker fusion transformer for audio-visual zero-shot learning. IEEE Transactions on Image Processing (2024)
25. Lu, Y., Li, C.: GCAN: graph-aware co-attention networks for explainable fake news detection on social media. In: ACL. pp. 505–514 (2020)
26. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks. In: IJCAI. pp. 3818–3824 (2016)
27. Qi, P., Cao, J., Yang, T., Guo, J., Li, J.: Exploiting multi-domain visual information for fake news detection. In: ICDM. pp. 518–527. IEEE (2019)
28. Qian, S., Wang, J., Hu, J., Fang, Q., Xu, C.: Hierarchical multi-modal contextual attention network for fake news detection. In: SIGIR. pp. 153–162 (2021)
29. 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: ICML. pp. 8748–8763 (2021)
30. Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H.: DySAT: Deep neural representation learning on dynamic graphs via self-attention networks. In: WSDM. pp. 519–527 (2020)
31. Song, C., Yang, C., Chen, H., Tu, C., Liu, Z., Sun, M.: Ced: credible early detection of social media rumors. TKDE pp. 3035–3047 (2019)
32. Tian, T., Liu, Y., Yang, X., Lyu, Y., Zhang, X., Fang, B.: Qsan: A quantum-probability based signed attention network for explainable false information detection. In: CIKM. pp. 1445–1454 (2020)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NIPS pp. 5998–6008 (2017)
34. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint [arXiv:1710.10903](https://arxiv.org/abs/1710.10903) (2017)
35. Wang, L., Zhang, C., Xu, H., Xu, Y., Xu, X., Wang, S.: Cross-modal contrastive learning for multimodal fake news detection. In: MM. pp. 5696–5704 (2023)
36. Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., Gao, J.: EANN: Event adversarial neural networks for multi-modal fake news detection. In: KDD. pp. 849–857 (2018)
37. Wei, L., Hu, D., Zhou, W., Yue, Z., Hu, S.: Towards propagation uncertainty: Edge-enhanced bayesian graph convolutional networks for rumor detection. In: ACL/IJCNLP. pp. 3845–3854 (2021)
38. Wu, Y., Yang, J., Wang, L., Xu, Z.: Graph-aware multi-view fusion for rumor detection on social media. In: ICASSP. pp. 9961–9965. IEEE (2024)

39. Wu, Y., Zhan, P., Zhang, Y., Wang, L., Xu, Z.: Multimodal fusion with co-attention networks for fake news detection. In: *ACL/IJCNLP 2021*. pp. 2560–2569 (2021)
40. Yang, X., Lyu, Y., Tian, T., Liu, Y., Liu, Y., Zhang, X.: Rumor detection on social media with graph structured adversarial learning. In: *IJCAI*. pp. 1417–1423 (2021)
41. Yi, Z., Lu, S., Tang, X., Wu, J., Zhu, J.: Maccn: Multi-modal adaptive co-attention fusion contrastive learning networks for fake news detection. In: *ICASSP*. pp. 6045–6049 (2024)
42. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A convolutional approach for misinformation identification. In: *IJCAI*. pp. 3901–3907 (2017)
43. Yuan, C., Ma, Q., Zhou, W., Han, J., Hu, S.: Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In: *ICDM*. pp. 796–805 (2019)
44. Zhang, X., Cao, J., Li, X., Sheng, Q., Zhong, L., Shu, K.: Mining dual emotion for fake news detection. In: *WWW*. pp. 3465–3476 (2021)
45. Zheng, J., Zhang, X., Guo, S., Wang, Q., Zang, W., Zhang, Y.: MFAN: Multi-modal feature-enhanced attention networks for rumor detection. In: *IJCAI*. pp. 2413–2419 (2022)
46. Zhou, X., Wu, J., Zafarani, R.: SAFE: similarity-aware multi-modal fake news detection. In: *PAKDD*. pp. 354–367 (2020)
47. Zubiaga, A., Liakata, M., Procter, R.: Exploiting context for rumour detection in social media. In: *SocInfo*. pp. 109–123 (2017)



# Learning to Synthesize Graphics Programs for Geometric Artworks

Qi Bing<sup>(✉)</sup>, Chaoyi Zhang, and Weidong Cai

School of Computer Science, The University of Sydney, Camperdown, NSW 2006,  
Australia

qbin4920@uni.sydney.edu.au

**Abstract.** Creating and understanding art has long been a hallmark of human ability. When presented with finished digital artwork, professional graphic artists can intuitively deconstruct and replicate it using various drawing tools, such as the line tool, paint bucket, and layer features, including opacity and blending modes. While most recent research in this field has focused on art generation, proposing a range of methods, these often rely on the concept of artwork being represented as a final image. To bridge the gap between pixel-level results and the actual drawing process, we present an approach that treats a set of drawing tools as executable programs. This method predicts a sequence of steps to achieve the final image, allowing for understandable and resolution-independent reproductions under the usage of a set of drawing commands. Our experiments demonstrate that our program synthesizer, Art2Prog, can comprehensively understand complex input images and reproduce them using high-quality executable programs. The experimental results evidence the potential of machines to grasp higher-level information from images and generate compact program-level descriptions.

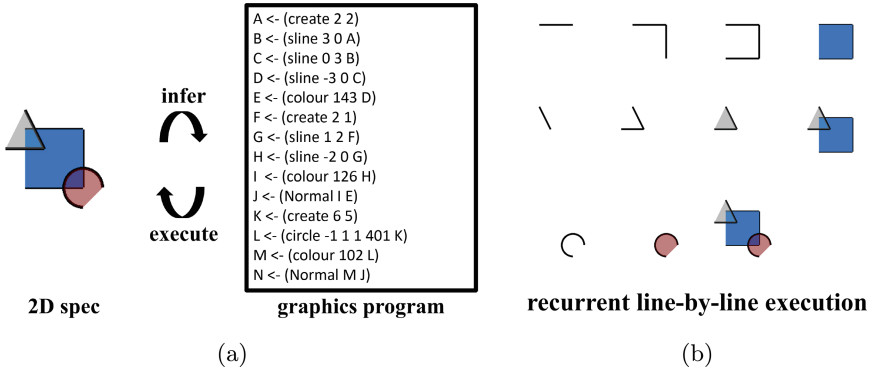
**Keywords:** Program synthesis · Vector graphics · Image vectorization · Image reasoning

## 1 Introduction

Humans can easily understand the procedure that generates an image, no matter the drawing or characters, which necessitates understanding its underlying structure. However, inferring the drawing process from only the final image presents significant challenges. These challenges stem primarily from the occlusion of shapes and inherent ambiguity, as multiple interpretations can often be equally valid. Recent research has proposed various definitions for the process leading to the final image, including sketch colorization [13, 23, 26, 28], color segmentation [1] and time-lapse video generation [29]. Though these methods accomplished their tasks, their results still suffer from low resolution, distortion and noise to different degrees. There are also similar methods that aim to bridge the gap between images and other forms of description, such as vector graphics, by



utilizing different types of parametric primitives, such as closed paths [16, 18] or strokes [14, 30]. Though these works produce promising vector-based results, they do not target to reason their generations. Instead, they approach the process more akin to vector-level segmentation. To address these issues, we design a graphics program that can comprehensively represent the drawing process, mirroring the methods used by artists with digital drawing tools (e.g., straight or curved lines, paint buckets and layer blendings). By introducing an executable graphics program, images can be represented as structured drawing commands, enabling their reconstruction at any resolution (see Fig. 1 as an example). Beyond its reconstruction capabilities, our proposed graphics program offers a representation of graphics that is not only readable and editable but also semantically meaningful. This makes it an ideal candidate for further applications, including drawing instruction.



**Fig. 1.** (a): A complex 2D graphic composed of lines and colors can be represented and reproduced by an executable program. (b): The program comprises individual lines of code, each corresponding to a specific drawing command. Executing these commands sequentially reveals the process of constructing the graphic, closely mirroring an artist’s workflow.

Our work builds upon recent developments in graphics program synthesis [5, 8, 19], which have demonstrated the potential of program synthesis in decomposing complex shapes into a series of commands. However, the recurrent inference of the drawing process for colored images remains largely unexplored, and is the main focus of this paper. Unlike most vectorization-based methods [12, 14, 16, 18, 30], our approach does not rely on a differentiable rasterizer for path optimization or supervision in the pixel space. Instead, we train a program synthesizer to generate codes directly from a single image input. Our experimental results indicate that Art2Prog outperforms state-of-the-art optimization-based vectorization approaches in reconstruction accuracy while executing the inferred programs. Additionally, our method is capable of representing more complex graphics compared to existing graphics program synthesizers.

In summary, the contributions of this paper are threefold:

- As shown in Fig. 1, we define a colored graphics program that emulates an artist’s workflow. This program is comprehensible to humans and capable of generating images at any resolution, thus efficiently bridging the bidirectional gap between programs and images.
- We develop Art2Prog, a novel GPT-2 [17] based program synthesizer for generating colored graphics programs from a single input image. This architecture demonstrates the feasibility of capturing high-level information such as the number of layers, enclosed shapes, layer overlaps, and color blending modes, all through the inference of a complex 2D graphics program.
- We evaluate our performance with state-of-the-art program synthesizers and optimization-based approaches in image vectorization. The experimental results show that Art2Prog outperforms existing works, achieving the highest reconstruction accuracy while also producing high quality program-level explanation.

## 2 Related Works

### 2.1 Graphics program synthesis

The task of learning to synthesize 2D graphics programs is not novel, with numerous recent papers being focused on reproducing 2D binary shapes. [5, 8, 19] primarily focused on reproducing CSG-based shapes, which are binary representations of solid shapes formed by applying boolean operations to simple shapes like circles and rectangles on the canvas. These methods successfully reconstructed solid shapes comprising up to 20 objects. Similarly, [4] constructed complex shapes by stacking a pre-defined binary shape (bricks of different lengths) built on the idea of Lego bricks. Written in a subset of L<sup>A</sup>T<sub>E</sub>X, [6] defined programs as line shapes (e.g., circle, rectangle and straight line) rendered on an empty canvas. Building on this concept, [7] conceptualized the program as controlling a ‘pen’ that draws binary lines on an empty canvas, and [27] followed a similar approach for reconstructing CAD sketches by sequentially drawing lines. Additionally, [9] introduced parameterized brushstrokes as programs and generated blurry paintings from photos.

To the best of our knowledge, our proposed Art2Prog is the first work that explicitly targets the inference of complex 2D graphics programs that include lines, colored surfaces, and overlapping layers with different blending modes.

### 2.2 Image vectorization

Different from image rasterization, vectorization is inherently more complex due to the potential non-uniqueness of its results. Traditional methods [3, 10, 11, 21, 22, 25] normally build specific algorithm-based methods that rely on image segmentation to conduct vectorization. To address this issue, recent research has tried to leverage the power of learning-based approaches. Studies [12, 14, 16, 30]

approached vectorization by optimizing a fixed number of parametric strokes relying on differentiable rasterizers. However, these methods needed to fully account for shape semantics, leading to redundant and inaccurate vectorization. Meanwhile, [18] trained an encoder-decoder model without supervision from vector ground-truth. However, it still relied on a differentiable rasterizer for loss backpropagation. Similarly, [2, 15] attempted vector reconstruction using Variational Autoencoders (VAEs) but failed to reach accurate results. [20] implemented category-conditioned image vectorization through a two-module network, adding one layer of solid color at a time. Their recent works highlight the potential of deep architectures to capture higher-level structural information in image vectorization.

Different from most existing vectorization approaches, Art2Prog does not rely on differentiable rasterization. Instead, we directly synthesize a program from the image input and execute it to accurately reconstruct the image. Furthermore, our method not only generates resolution-independent vector graphics but also comprehensively describes the image through human-understandable programs.

**Table 1.** The domain-specific language (DSL) for our 2D graphics.

Program $P$	$\rightarrow O \mid O(O) \mid L(O, O)$
Operation $O$	$\rightarrow \text{Create}(x = N, y = N)$ $\quad \mid \text{Straight}(x = N, y = N, O)$ $\quad \mid \text{Circle}(x = N, y = N, r = R,$ $\quad \quad \quad \text{dir} = D, O)$ $\quad \mid \text{Fillcolor}(\text{color} = C, O)$
Layer $L$	$\rightarrow \text{Normal}(O1 = O, O2 = O)$ $\quad \mid \text{Multiply}(O1 = O, O2 = O)$
Position $N$	$\rightarrow$ integers within range of $[-8 : 8]$
Radius $R$	$\rightarrow$ integers within range of $[1 : 4]$
Direction $D$	$\rightarrow \text{True} \mid \text{False}$
color $C$	$\rightarrow$ integers within range of $[1 : 54]$

### 3 Graphics Programs

This section defines the domain-specific language (DSL) used for our 2D graphics program. As depicted in Table 1, our graphics program is structured hierarchically. A final image comprises several overlapping layers, each utilizing one of two distinct color blending modes. To construct each layer, multiple drawing commands must be executed sequentially:

1. A *Create* ( $x, y$ ) command that initiates a new layer on the current canvas and sets the starting position at coordinates ( $x, y$ ).

2. Multiple *Straight* ( $x, y, O$ ) commands that draw continuous straight lines from the last position to a relative position ( $x, y$ ). These commands also help define enclosed shapes that can be filled with color.
3. Multiple *Circle* ( $x, y, r, dir, O$ ) commands that draw circular arcs. These arcs extend from the current position to a specified relative position ( $x, y$ ), defined by a radius  $r$  and a direction  $dir$ , which can be either clockwise or counterclockwise.
4. A *Fillcolor* ( $C, O$ ) command that applies the color  $C$  to an enclosed shape. This enclosed shape is determined by the line path resulting from an earlier operation  $O$ , which is defined by the arrangement of lines in the current layer.

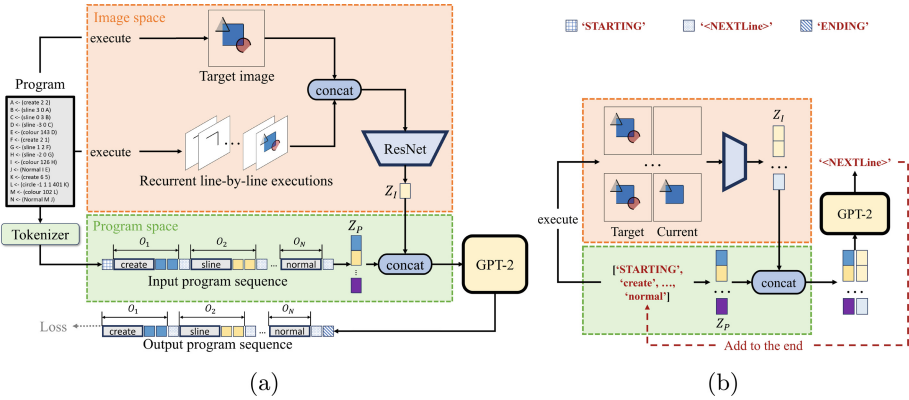
*Normal* ( $O, O$ ) or *Multiply* ( $O, O$ ) employs two distinct layer blending modes to connect a pair of object layers, denoted as  $O$ . All of the variables  $N, R, D$  and  $C$  in Table 1 are defined as tokens in this paper. We incorporate two types of layer blending modes—‘normal’ and ‘multiply’—to enhance visual variety in the output (controlled by corresponding tokens). To further constrain the search space, we divide the positions on a canvas into an  $8 \times 8$  grid. Additionally, instead of using separate tokens in color channels (RGBA), we build a color list for  $C$  that contains 54 different colors to choose from. This allows a single token to correspond to a wide range of colors, akin to a palette commonly used in modern drawing software and traditional paintings. In summary, we define our goal of 2D graphics program inference as follows: reconstructing an input image (at a resolution of  $64 \times 64$ ) by executing an inferred graphics program.

## 4 Methodology

In this section, we begin with a detailed description of the model architecture in Section 4.1 and introduce the tokenization strategy in Section 4.2 that enables our language model to interpret graphics programs as a sequence of tokens. Subsequently, we will discuss the training process in Section 4.3, followed by an explanation of the inference pipeline in Section 4.4.

### 4.1 Model architecture

We developed a deep architecture capable of efficiently inferring an executable graphics program from a randomly drawn image under our defined DSL. The comprehensive neural design of our network is illustrated in Fig. 2(a), including two trainable modules: an image encoder (ResNet) to encode the executed image results and a program decoder (GPT-2) to learn the probability distribution of the tokenized program sequence. To bridge the gap between program syntax and semantics effectively, our approach integrates information from both image and program spaces. Inspired by REPL [5], the graphics program is executed sequentially, line-by-line, to generate intermediate images through a non-differentiable, off-the-shelf rasterizer. These images are then concatenated with the target image, creating an 8-channel input image to be fed into an image



**Fig. 2.** The overall architecture of Art2Prog, which contains two trainable modules: an ResNet-based image encoder and a program decoder built upon the architecture of GPT-2 [17]. (a): To reconstruct a target image into an executable graphics program, Art2Prog treats the program as a flattened sequence of tokens. It also utilizes image embeddings derived from the target image and partially executed programs as conditioning inputs. (b): Art2Prog infers a graphics program directly from the target image. By predicting the ‘ENDING’ token, Art2Prog is capable of inferring programs of various lengths.

encoder. The encoder follows the architecture of ResNet without pretraining on other datasets.

To process the graphics program, we flatten it into a sequence of tokens, appending a special token ‘<NEXTLine>’ at the end of each line to signify its termination. This tokenized program sequence is then embedded and concatenated with the image embeddings on a token-wise basis, as illustrated in Fig. 2(b). Given that each line of code can only be executed following the prediction of a ‘<NEXTLine>’ token, each token within the same line of code is associated with the same executed image. Therefore, to enable the concatenation of images and programs in a practical manner, we duplicate the images to match the token length for each executed line of code. Only when a ‘<NEXTLine>’ token is predicted is the execution result of the current program updated. We build our program decoder on the basis of an existing language model (GPT-2 [17]) to decode the concatenated embeddings of the current state into the program sequence for the next state. Additionally, we train our decoder from scratch without pretraining on other datasets. Consequently, we simplify the program inference problem by predicting the next token based on the current program sequence. The prediction distribution for a graphics program can thus be factorized as follows:

$$p(S|I_T) = \prod_{k=1}^K p(t_k | g_{\theta} \{f_{\theta}(I_T, I_j), t_j\}_{j=1}^{k-1}), \quad (1)$$

where  $t_1..t_K$  are the tokens in the target flattened program sequence  $S$ ,  $K$  is the length of program sequence that varies across different programs,  $g_\theta$  is the sequence decoder (GPT-2), and  $f_\theta$  is the image feature extractor (we use ResNet-18 in this paper).  $I_T$  and  $I_j$  are the target image and canvas rendering at token  $t_j$  respectively.

## 4.2 Tokenization

Art2Prog employs a unique tokenization strategy to convert a graphics program into a sequence of tokens, facilitating feature concatenation and sequence decoding. The graphics program  $P$  can be represented as  $P = (O_1, \dots, O_N)$ , where  $O_i$  indicates the  $i^{\text{th}}$  line of code in the program. As mentioned in Table 1, we first quantize arguments  $X_i \in \{N, R, D, C\}$  into distinct intervals as tokens, and similarly, assign tokens to command classes  $C_i$  (such as *create*, *sline*, *circle*, *color*, *normal*, and *mul*). Therefore, a single line of code  $O_i$  may contain 2 types of tokens:  $O_i = (C_i, v_i^1, \dots, v_i^k)$ . Here, we do not tokenize the pointer to the former line  $O_{i-1}$  because we execute our code line-by-line by default so that it can be simplified. For the layer combination commands *normal* and *mul*, our detokenizer refers to  $O_{i-1}$  and the second last *color* command, which should indicate the end of a layer. Additionally, we introduce 3 special tokens {'STARTING', 'ENDING', '<NEXTLine>'} that indicate the start of a sequence, the end of a sequence, and the end of a line respectively.

## 4.3 Training

The primary training objective of our model is to minimize the cross-entropy loss for the predicted tokens at each position in the program sequence. Given the target program sequence  $S$  and a corresponding target image  $I_T$ , we train our model  $\Theta$  to minimize:

$$l(\hat{S}, S) = CE(\hat{S}, S|I_T; \Theta), \quad (2)$$

where  $CE()$  refers to the cross-entropy function, and  $\hat{S}$  is the output program sequence of our model.

Inspired by recent transformer-based language models, we shift the input sequence  $s$  to the right by one position, as shown in Fig. 2(a). Thus, the input sequence starts with a special token 'STARTING' and the output sequence ends with a special token 'ENDING'. For each line of code  $O_n$ , we assume that the preceding lines ( $O_1, \dots, O_{n-1}$ ) have been correctly successfully inferred; thus, we execute and render these partial lines to produce  $n - 1$  intermediate images ( $I_1, \dots, I_{n-1}$ ). As illustrated in Fig. 2(b), these intermediate images, concatenated with the target image, are fed to ResNet to generate  $n - 1$  image embeddings. Since the lines are flattened to sequence before being fed into the GPT-2, the number of image embeddings should match the token length. Thus, we repeat each image embedding  $I_i$  to the length  $L_O + 1$ , where  $L_O$  is the token length of the code to which these embeddings pertain. The addition of 1 accounts for

---

**Algorithm 1** The inference process of Art2Prog.
 

---

**Input:**

the target RGBA image (spec)

**Output:**Programs  $P_{best}$ **Initialisation:** Start an empty program  $P$ .Start an empty sequence of tokens  $O$  indicating current line of code.Set the maximum number of token in each line as  $N_t$ .Set the maximum number of lines in  $P$  as  $N_O$ . $P \leftarrow$  ‘STARTING’**repeat**  **if**  $\text{len}(O) > N_t$  **then**    Reset  $O$   **end if**  Samples the next token  $t$  from current  $P$  and  $O$    $O \leftarrow t$   **if**  $t$  is ‘<NEXTLine>’ **then**     $P \leftarrow O$     Reset  $O$   **end if**  **if**  $P_{best}$  **then**    **if**  $\text{Loss}(P) < \text{Loss}(P_{best})$  **then**       $P_{best} = P$     **end if**  **else**     $P_{best} = P$   **end if****until**  $\text{len}(P) > N_O$  or  $t$  is ‘ENDING’**return**  $P_{best}$ 

the additional special token ‘<NEXTLine>’. We then concatenate the extended image embeddings and program embedding in a token-wise manner prior to being fed into GPT-2.

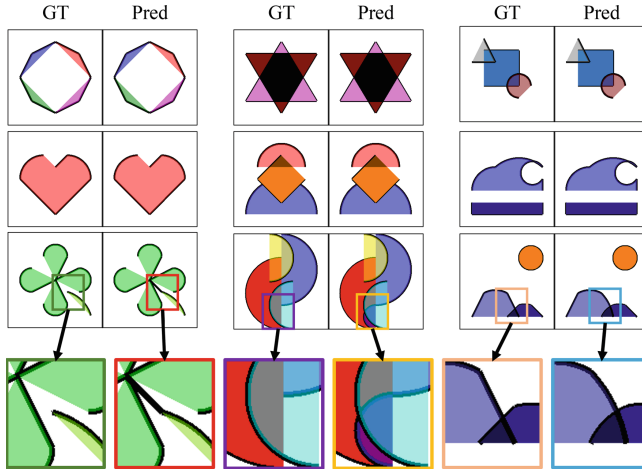
#### 4.4 Inference

As shown in Fig. 2(b), our model infers a single token at a time, beginning with the initial input sequence [‘STARTING’] and terminating with the prediction of ‘ENDING’. This design allows our model to infer graphics programs of varying lengths based solely on a target image as input. Our approach does not rely on specific search algorithms, such as beam search or Sequential Monte Carlo (SMC), which can significantly slow down the inference process. Instead, we employ a simple greedy search strategy while supporting early stopping if the program has already been correctly inferred (indicated by  $MSE = 0$ ).

The implementation details of our graphics program inference scheme are shown in Algorithm 1. For each inference, we repeatedly conduct inference from empty until timeout. After that, we compare all of the generated programs by

their  $IoU_{rgba}$  distance from the target raster image to find the best match. The metric  $IoU_{rgba}$  is defined as a modified version from  $IoU$ , which aims to measure the similarity of RGBA graphics with objects of different sizes.

Experimental results demonstrate that Art2Prog is capable of producing high-quality inferences within a short time frame (approximately 15 seconds), surpassing current state-of-the-art techniques. We present examples of our inference results in Fig. 3. The comparative analysis of results and a detailed implementation will be presented in the subsequent section.



**Fig. 3.** Inference results on hand-drawn geometric artworks. We render all the images at resolution  $300 \times 300$  for better visualization. The input image size to the model is still  $64 \times 64$ . The bottom column shows examples of failure cases.

## 5 Experiments and Results

### 5.1 Data preparation and experiment settings

We collect data for training by randomly generating graphics programs with up to 10 layers with reference to the defined DSL as described in Section 3. During training, our model was exposed to approximately 6 million examples. We utilize the Adam optimizer with a learning rate of  $1 \times 10^{-3}$ , using a batch size of 32 across one RTX 3090 GPU for all settings in Table 2. For evaluation, we built an eval set with 1000 generated data up to 13 layers in each program. Also, inspired by the design of  $IoU$ , we define a modified version to fairly compare the similarity among RGBA images with objects of different sizes:

$$IoU_{rgba}(\hat{I}, I) = \frac{\sum_{p=1}^P (\hat{I}_p = I_p)}{\sum_{p=1}^P (\hat{I}_p(A) > 0 \text{ and } I_p(A) > 0)}, \quad (3)$$



where  $\hat{I}$  and  $I$  denote the predicted image and the target image, respectively.  $\hat{I}_p$  and  $I_p$  indicate the pixel value of  $\hat{I}$  and  $I$  at position  $p$ .  $\hat{I}_p(A)$  and  $I_p(A)$  are the pixel values at the alpha channel of images  $\hat{I}$  and  $I$ , which indicate the transparency in the RGBA color space.

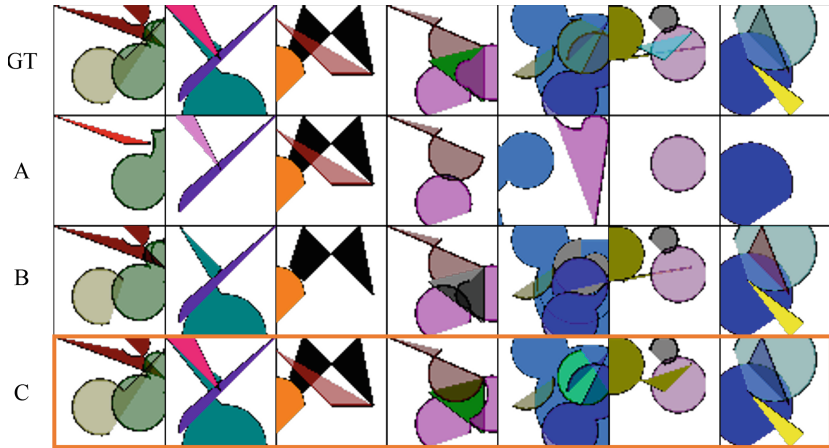
**Table 2.** Quantitative results under different architecture designs of Arg2Prog. #Layers denotes the maximum number of program layers each model was exposed to during training. SR denotes the rate of correct inferences ( $IoU_{rgba} = 1.0$ ).  $Max_{PL}$  denotes the maximum program lengths. These scores are all derived from the same eval set, which comprises 1,000 data entries, each containing programs with up to 13 layers. We have equipped Monte Carlo Tree Search (MCTS) with models that utilize Pointer Network (PtrNet) [24] as the decoder. The inference timeout for all models is set at 125 seconds in this table.

Model	Encoder	Decoder	#Layers (L)	$IoU_{rgba}$ ( $\uparrow$ )	$IoU$ ( $\uparrow$ )	MSE ( $\downarrow$ )	SR ( $\uparrow$ )	$Max_{PL}$ ( $\uparrow$ )
$A_{L=5}$	CNNs	PtrNet	5	0.7535	0.8471	0.1739	0.593	26
$B_{L=5}$	RN	PtrNet	5	0.9330	0.9692	0.0447	0.907	26
$C_{L=5}$	RN	GPT-2	5	0.9420	0.9858	0.0304	0.919	26
$A_{L=10}$	CNNs	PtrNet	10	0.5529	0.6722	0.2968	0.275	17
$B_{L=10}$	RN	PtrNet	10	0.9363	0.9719	0.0399	0.920	26
$C_{L=10}$	RN	GPT-2	10	0.9631	0.9923	0.0182	0.951	27

## 5.2 Ablations

In order to ablate our architecture, we assessed the impacts of two key components in Art2Prog: the image encoder and the program generator. Additionally, we evaluated the influence of the training set on model performance. Our models are tested on an evaluation set comprising up to 13 layers. Thus, we conducted training on two different sets, one with layers up to 5 and another up to 10, to determine whether training with longer programs enhances the performance of the synthesizer.

We first compared the evaluation scores of different image encoders, including sequentially stacked Conv2d blocks with ReLU activation (CNNs) and ResNet-18. Notably, we introduced a minor modification to the original ResNet structure by removing Batch Normalization. This alteration led to a substantial improvement in performance. As indicated in the  $A_{L=5}$  and  $B_{L=5}$  of Table 2, compared with CNNs, ResNet demonstrates higher reconstruction accuracy ( $IoU_{rgba}$ ,  $IoU$  and MSE) and the ability to precisely infer longer programs (with the highest rate of successfully inferred programs and maximum program length). When trained with longer programs (up to 10 layers), ResNet demonstrates a slight improvement in performance. Conversely, CNNs encounter difficulties in training under these conditions. By comparing the qualitative results in Fig. 4, it is evident that utilizing ResNet facilitates the construction of complex programs in



**Fig. 4.** Qualitative results of different architecture designs of Art2Prog. We pick the best models ( $A_L=5$ ,  $B_L=10$ , and  $C_L=10$ ) based on their average inference accuracy for comparison.

the majority of scenarios. However, in specific instances, such as those illustrated in the third column of this figure, CNNs demonstrate superior performance in accurately reproducing a given image.

Subsequently, we demonstrated the necessity of using GPT-2 as the program generator, as opposed to Pointer Network (PtrNet) [24]. When utilizing PtrNet as the generator, the program is not flattened but treated as separate lines of code as in [5]. We observed that GPT-2 consistently demonstrates superior accuracy, regardless of whether it is trained with longer program or not. Upon comparing the qualitative results depicted in the figure, it is observed that GPT-2 exhibits a marginally superior capability in preserving the sharp details within raster images. However, PtrNet achieves satisfactory performance in the majority of cases. Overall, models trained with longer programs exhibit better generalization capabilities in complex scenes.

### 5.3 Comparison with the state-of-the-art methods

We compared our proposed method against two other state-of-the-art (SOTA) image vectorization methods: LIVE [16] and REPL [5]. We conducted this evaluation using a consistent dataset comprising 1000 data points, each generated under our DSL and containing up to 13 object layers per program. The results of this comparison are presented in Table 3 and Fig. 5.

Regarding the colored image-to-SVG method, LIVE exhibits inaccuracies in color inference, which negatively impacts its average pixel accuracy (MSE). Moreover, LIVE demonstrates difficulties in handling overlapped areas and in reconstructing paths that self-intersect. As illustrated in Fig. 5(a), Art2Prog consistently maintains layer integrity in scenarios with overlapping layers and

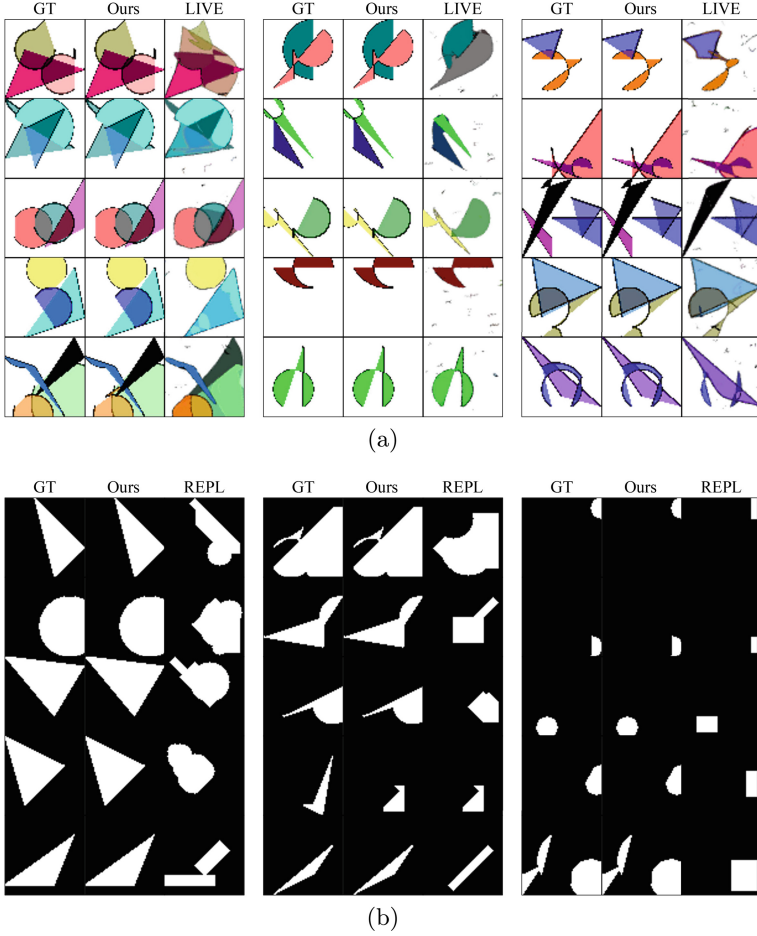


Fig. 5. Our qualitative comparison with SOTA methods LIVE and REPL.

Table 3. Inference comparison on the evaluation set. Here we only picked the transparency channel for  $IoU$  calculation. The timeout for each inference is 15 seconds in this table.

Model	Search Algorithm	$IoU$ ( $\uparrow$ )	MSE ( $\downarrow$ )
REPL [5]	GS	0.4514	-
REPL [5]	Beam	0.5338	-
REPL [5]	SMC	0.6630	-
LIVE [16]	-	0.8513	0.0508
Arg2Prog (ours)	GS	<b>0.9733</b>	<b>0.0491</b>

accurately reconstructs self-intersections. This figure showcases comparisons in various cases. The left column presents scenarios where LIVE overlooks details in overlapped layers of shapes, attributed to a misinterpretation of the segmentation process. The middle column reveals that LIVE performs poorly with areas of self-intersection and often fails to capture sharp shapes accurately. The column on the right demonstrates cases involving both self-intersection and layer overlap, areas where LIVE struggles to achieve accurate reconstruction. In contrast, our approach is capable of comprehensively representing these complexities as graphics programs.

Since the REPL inverse CAD model supports only binary images as input, we processed the input data accordingly. As illustrated in Fig. 5(b), it is evident that REPL struggles with the vectorization of complex and sharp shapes, often inaccurately interpreting shapes at the canvas’s edges. In the examples showcased in the left column of this figure, REPL frequently generates redundant shapes to represent simple structures, such as a single triangle or a circle. For sharp shapes, as depicted in the middle column, REPL similarly faces difficulties in achieving accurate reconstruction. The right column demonstrates scenarios where, when tasked with predicting the graphics program for a shape positioned in a corner, REPL often inaccurately predicts a rectangle instead of the correct shape, such as a circle. In contrast, Art2Prog demonstrates precise program inference regardless of the shape’s position and size.

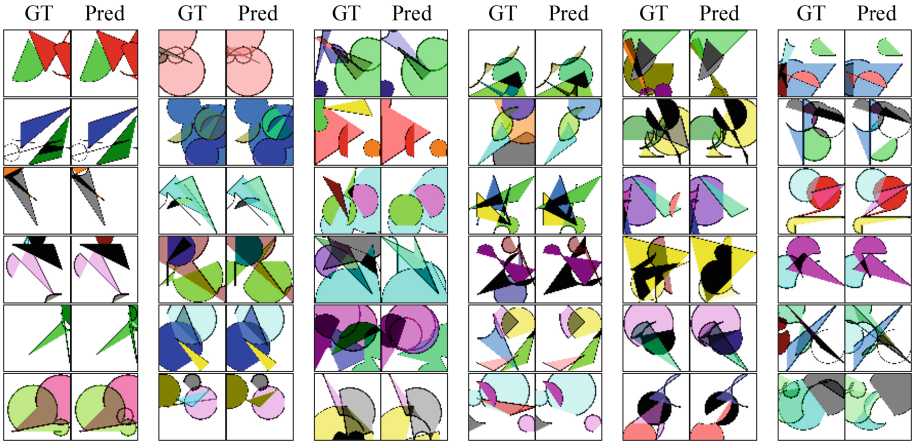
In summary, our model demonstrates superiority in color inference and the intricate reconstruction of compositions involving multiple shapes, producing outputs that are both clean and precise.

## 6 Discussion

In this section, we examine the instances where our method failed to produce accurate predictions, as illustrated in Fig. 6. These examples demonstrate failures in replicating the input raster image through generated graphic programs, particularly highlighting our system’s tendency to neglect minor details, as evident in the first column of examples in Fig. 6. To improve accuracy, enhancing the image encoding component to more effectively capture detailed image features may be beneficial.

Failures in accurately determining the appropriate color blending mode or the exact color in scenarios where one layer completely encompasses another are depicted in the second column. These challenges stem from the ambiguity in distinguishing between blended colors and the dominant color of the upper layer. Incorporating a dedicated, trainable module specifically designed for layer combination might mitigate this issue.

Moreover, in complex multi-layered images, our synthesis algorithm often misses segments, indicating difficulty in generating longer programs. This limitation points to potential advancements in program generation capabilities, possibly by exploring innovative architectural solutions or integrating more sophisticated large language models for future enhancement.



**Fig. 6.** Examples of failed cases, where the predicted graphics programs do not reproduce the target image comprehensively.

## 7 Conclusion

This paper presents a learning-based program synthesizer that aims to write a graphics program to represent the input image comprehensively. We propose a novel graphics program definition by separating the drawing steps towards a target RGBA image into several steps: (1) creating a new layer on the canvas, (2) outlining an enclosed shape with continuous lines, (3) filling the lined area in the current layer with colors and (4) combining layers with blending modes, which mimics the workflow of real-world digital artists. The quantitative and qualitative results in our experiments demonstrate that our approach achieves high-quality reconstruction results and effectively discerns the underlying drawing process. Though the drawing commands are highly simplified compared to existing drawing software, this paper can be considered as an initial step toward complex graphics program synthesis. Thus, future works can extend it to a broader range of more complex graphics.

## References

1. Akimoto, N., Zhu, H., Jin, Y., Aoki, Y.: Fast soft color segmentation. CVPR (June 2020)
2. Carlier, A., Danelljan, M., Alahi, A., Timofte, R.: Deepsvg: A hierarchical generative network for vector graphics animation. NeurIPS (2020)
3. Diebel, J.R.: Bayesian image vectorization: the probabilistic inversion of vector image rasterization. Ph.D. thesis, Stanford University (2008)
4. Dumancic, S., Guns, T., Cropper, A.: Knowledge refactoring for inductive program synthesis. AAAI (2021)

5. Ellis, K., Maxwell, M., Pu, Y., Sosa, F., Tenenbaum, J.B., Solar-Lezama, A.: Write, execute, assess: Program synthesis with a repl. *NeurIPS* **32** (2019)
6. Ellis, K., Ritchie, D., Solar-Lezama, A., Tenenbaum, J.: Learning to infer graphics programs from hand-drawn images. *NeurIPS* pp. 6062–6071 (2018)
7. Ellis, K., Wong, C., Nye, M., Sablé-Meyer, M., Morales, L., Hewitt, L., Cary, L., Solar-Lezama, A., Tenenbaum, J.B.: Dreamcoder: Bootstrapping inductive program synthesis with wake-sleep library learning. *PLDI* p. 835–850 (2021)
8. Feser, J., Dillig, I., Solar-Lezama, A.: Inductive program synthesis guided by observational program similarity. *Proceedings of the ACM on Programming Languages* **7**, 912–940 (2023), <https://api.semanticscholar.org/CorpusID:263220362>
9. Ganin, Y., Kulkarni, T., Babuschkin, I., Eslami, S., O, V.: Synthesizing programs for images using reinforced adversarial learning. *ICML (04 2018)*
10. Hilaire, X., Tombre, K.: Robust and accurate vectorization of line drawings. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**, 890–904 (2006)
11. Lecot, G., Levy, B.: Ardeco: Automatic Region DEtection and CONversion. p. 349–360 (2006)
12. Li, T., Lukác, M., Gharbi, M., Ragan-Kelley, J.: Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics* **39**, 1–15 (2020), <https://api.semanticscholar.org/CorpusID:221686970>
13. Liu, B., Song, K., Zhu, Y., Elgammal, A.: Sketch-to-art: Synthesizing stylized art images from sketches. *ACCV* p. 207–222 (2020)
14. Liu, S., Lin, T., He, D., Li, F., Deng, R., Li, X., Ding, E., Wang, H.: Paint transformer: Feed forward neural painting with stroke prediction. *ICCV* pp. 6578–6587 (2021), <https://api.semanticscholar.org/CorpusID:236956975>
15. Lopes, R.G., Ha, D.R., Eck, D., Shlens, J.: A learned representation for scalable vector graphics. *ICCV* pp. 7929–7938 (2019), <https://api.semanticscholar.org/CorpusID:102353397>
16. Ma, X., Zhou, Y., Xu, X., Sun, B., Filev, V., Orlov, N., Fu, Y., Shi, H.: Towards layer-wise image vectorization. *CVPR* (2022)
17. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
18. Reddy, P., Gharbi, M., Lukác, M., Mitra, N.J.: Im2vec: Synthesizing vector graphics without vector supervision. *CVPR* pp. 7338–7347 (2021), <https://api.semanticscholar.org/CorpusID:231802181>
19. Sharma, G., Goyal, R., Liu, D., Kalogerakis, E., Maji, S.: Csgnet: Neural shape parser for constructive solid geometry. *CVPR* (06 2018)
20. Shen, I., Chen, B.: Clipgen: A deep generative model for clipart vectorization and synthesis. *IEEE Transactions on Visualization and Computer Graphics* **28**, 4211–4224 (2021), <https://api.semanticscholar.org/CorpusID:235269254>
21. Sykora, D., Buriánek, J., Zara, J.: Sketching Cartoons by Example. *SBIM* pp. 27–34 (2005)
22. Sýkora, D., Buriánek, J., Žára, J.: Video codec for classical cartoon animations with hardware accelerated playback. *Advances in Visual Computing* pp. 43–50 (2005)
23. Tseng, H., Fisher, M., Lu, J., Li, Y., Kim, V., Yang, M.: Modeling artistic workflows for image generation and editing. *ECCV* (2020)
24. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. *NeurIPS* **28** (2015)
25. Xia, T., Liao, B., Yu, Y.: Patch-based image vectorization with automatic curvilinear feature alignment. *SIGGRAPH Asia* **10** (2009)
26. Xiang, X., Liu, D., Yang, X., Zhu, Y., Shen, X., Allebach, J.P.: Adversarial open domain adaptation for sketch-to-photo synthesis. *WACV* (2022)

27. Yuezhi, Y., Hao, P.: Discovering design concepts for cad sketches. NeurIPS (2022)
28. Zhang, L., Li, C., Simo-Serra, E., Ji, Y., Wong, T., Liu, C.: User-guided line art flat filling with split filling mechanism. CVPR (2021)
29. Zhao, A., G., Lewis, K.M., Durand, F., Gutttag, J.V., Dalca, A.V.: Painting many pasts: Synthesizing time lapse videos of paintings. CVPR (2020)
30. Zou, Z., Shi, T., Qiu, S., Yuan, Y., Shi, Z.: Stylized neural painting. CVPR pp. 15684–15693 (2020)



# PPCap: A Plug and Play Framework for Efficient Stylized Image Captioning

Xiangpeng Wei, Yi Li<sup>(✉)</sup>, Guisheng Liu, Yating Liu, and Yanqing Guo

Dalian University of Technology, Dalian 116024, Liaoning, China  
liyi@dlut.edu.cn

**Abstract.** Stylized image captioning aims at generating captions that accurately describe the image content while aligning with the desired style. In semi-supervised setting, existing methods typically first pre-train models on large-scale factual image-caption pairs and then fine-tune the pre-trained models on small-scale unpaired stylized corpus, requiring significant resources. In this paper, we propose PPCap, a novel Plug and Play framework for stylized image captioning, where only a stylized image captioning model needs to be trained on small-scale unpaired stylized corpus. Then it will form a generative style discriminator via Bayes rule by the contrast of the captions in different styles, guiding an off-the-shelf large-scale factual image captioning model to generate stylized image captions in a post-processing manner, which is flexible and efficient. Experimental results on SentiCap and FlickrStyle10k show that our framework achieves comparable performance to the state-of-the-art methods in the same setting while reducing training time by over 90%. Our code is available at <https://github.com/gWeiXP/PPCap>.

**Keywords:** Plug and Play framework · generative style discriminator · training efficiency

## 1 Introduction

Stylized image captioning aims at generating captions that accurately describe the image content while aligning with the desired style. When people describe an image, they often incorporate their styles. As shown in Fig. 1, people with different styles have different captions for the same image. Adding style elements to a caption not only enriches how images are presented but also helps users convey their emotions. For example, on social media platforms, pairing travel photos with poetic captions enhances their artistic appeal, while pairing daily life photos with humorous captions relaxes the audience, and so on.

There has been significant effort dedicated to stylized image captioning. Earlier methods mainly focus on using factual and stylized image-caption pairs to achieve supervised stylized image captioning [3, 17]. However, collecting image-caption pairs incurs significant human and time costs, especially for stylized data. To solve this problem, SAN [12] proposes a data augmentation framework to extend stylized image-caption pairs, while more researchers propose





**Factual:** a man and a dog sitting on a bench.  
**Pos:** a happy man sitting on a bench with his dog.  
**Neg:** a lonely man sitting on a bench with his dog.  
**Ro:** a man and his dog enjoy a beautiful day at the park.  
**Hu:** a man sitting on a bench looking for mermaids.

**Fig. 1.** An example of the captions generated by our framework in factual, positive (Pos), negative (Neg), romantic (Ro), and humorous (Hu) styles for an input image, with the style-related words underlined.

semi-supervised methods to reduce reliance on stylized image-caption pairs [4, 7, 8, 18, 29]. Most of them use large-scale factual image-caption pairs to pre-train the model, enabling it to accurately describe the image content, and then use small-scale unpaired stylized corpus to fine-tune the pre-trained model, enabling it to incorporate stylistic elements. However, the pre-training process dominates the total training time due to the data scale. And this process is redundant with the training of the factual image captioning model, resulting in resource wastage.

In this paper, we propose PPCap, a novel Plug and Play framework for stylized image captioning. Our motivation is to decouple the task of stylized image captioning into generating accurate captions for input images and incorporating style elements into the captions, where any off-the-shelf factual image captioning model can be used for the former, then we only need to design a post-processing module to be responsible for the latter, i.e., generating appropriate stylized words or phrases in appropriate positions of the factual captions. Through our framework, we no longer require the pre-training process, saving most of the training time. To achieve such a functionality, we design a generative style discriminator as the post-processing module, which can discern, word by word, whether all the candidates of each word align with the desired style. The discriminator actually is composed of a lightweight stylized image captioning model via Bayes rule, which can generate captions aligning with desired and undesired styles for input images. And to train the stylized image captioning model solely on unpaired stylized corpus, we use the CLIP model [21] to align images and captions to the CLIP embedding space, allowing us to use captions instead of corresponding images for training. In the subsequent sections, we will respectively denote the factual and stylized image captioning model as factual model and stylized model.

The main contributions are summarized as follows:

- We propose a novel Plug and Play framework for stylized image captioning where the task of stylized image captioning is decoupled into generating factual captions and incorporating style elements into the captions. Then we can utilize the existing knowledge of any off-the-shelf factual image captioning model to generate accurate captions, achieving good performance without the need for the pre-training process.

- We design a generative style discriminator composed of a lightweight stylized image captioning model via Bayes rule, which can incorporate style elements in appropriate positions of the captions, thereby guiding the factual image captioning model to generate stylized image captions.
- Experiments on SentiCap and FlickrStyle10k datasets verify the effectiveness of our proposed framework. Results demonstrate that our framework can achieve comparable performance to the state-of-the-art methods in semi-supervised setting while reducing training time by over 90%.

## 2 Related Work

### 2.1 PLM-Based Controllable Text Generation

The existing controlled text generation methods based on pre-trained language models (PLM) can be roughly divided into three categories according to the working mode of control signals: retraining, fine-tuning, and post-processing. Among them, the retraining methods [2, 10] need to consume a lot of computing resources and also face the problem of lacking labeled data. With the rapid increase of parameters in the PLM, even fine-tuning has become resource-intensive. The post-processing methods [5, 11, 27] can fix the parameters of the pre-trained language model and use a post-processing module to guide the decoding process of text generation, thereby ensuring the quality of the generated text while consuming less computing resources. A representative method of this type is GeDi [11]. It trains a small class-conditional language model as the generative discriminator to guide the generation from large PLM like GPT-2 [22] and GPT3 [1]. Inspired by it, we propose our framework, utilizing a small stylized image captioning model as the generative style discriminator to guide the generation from a large factual image captioning model. However, compared to controllable text generation, stylized image captioning represents a more challenging endeavor. Controlled text generation only requires the generated text to align with the desired control signals. In contrast, stylized image captioning requires generated captions to not only match the desired style but also accurately describe the content of the images. To tackle this issue, we conducted a derivation of the formulas, migrating this post-processing method from text generation to image captioning. Refer to Sect. 3.1 for specifics details.

### 2.2 Image Captioning

The task of image captioning is to generate a caption that accurately describes the content of the image for a given input image. And the research on image captioning has made remarkable progress in recent years [6, 13, 14, 23]. With the application of various technologies such as attention mechanisms, graph neural networks, reinforcement learning, transformer, vision-language pre-training techniques, retrieval augmentation to image captioning models, all kinds of models have been proposed for image captioning. These models have learned extensive knowledge in both vision and language, enabling them to generate accurate

captions for input images. In our framework, we aim to make full use of their learned visual and language knowledge in a post-processing method that re-ranks the generated text without altering their model parameters to conduct stylized image captioning task. In our experiments, we choose ClipCap [19] as the factual model, which is a lightweight model based on CLIP [21] and GPT-2. It can achieve CIDEr score of 108.4 on MSCOCO test set. And to demonstrate the plug-and-play capability of our framework, we also plug our style discriminator into PureT to observe the performance. It is a purely Transformer-based model and can achieve CIDEr scores of 120.2 when trained using cross-entropy (XE) loss and 138.2 when optimized using the self-critical sequence training (SCST) strategy on the MSCOCO test set.

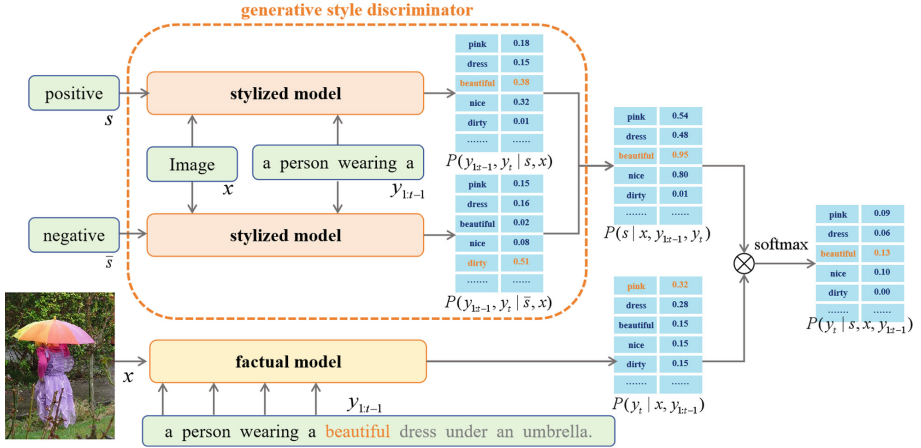
### 2.3 Stylized Image Captioning

Stylized image captioning works can be divided into two categories: methods using parallel stylized image-caption data (supervised method) and methods using non-parallel stylized corpus (semi-supervised method). Earlier works [3, 17] are all supervised methods, depending heavily on stylized sentences paired with images for training a stylized image captioning model. To address this issue, SAN [12] employs data augmentation to extend paired stylized datasets, while more methods adopt a semi-supervised approach, training models only using stylized sentences unpaired with images. StyleNet [7] decomposes the weight matrices in the LSTM network to model both factual sentences and stylized sentences. MSCap [8] uses a style gate to adaptively assign different weights to the image feature and the text feature. Semstyle [18] and MemCap [29] use sentiment terms and scene graphs, respectively, as an medium between vision and text to align images and captions. ADS-Cap [4] uses a contrastive learning module to align the image and text features and uses a conditional variational auto-encoder to memorize diverse stylistic patterns in latent space. But most of these works follow the traditional methodology of first pre-training models on large-scale factual image-caption pairs and then fine-tuning the pre-trained models on small monolingual textual corpus, or even directly training models with both types of data, which demands a considerable amount of computational resources and time. Our framework is also a semi-supervised method and only requires training a style discriminator on the stylized sentences unpaired with images to relieve this issue. Lately, TridentCap [25] simultaneously considers semantic alignment and mapping between image-fact-style trident data, fully mines the core relationship between them, and achieves a excellent performance. But it is also a supervised method and in addition to paired stylized images and captions, also needs corresponding factual captions as additional input to the model.

## 3 Method

### 3.1 Overview

Given an input image  $x$  and a style label  $s$ , our goal is to generate a sentence  $y_{1:T}$  that is semantically related to the given image  $x$  and consistent with the



**Fig. 2.** Overview of our framework. The lower part is the factual model, responsible for generating accurate captions for input images. The upper part is the generative style discriminator, responsible for discerning whether the candidates of each word align with the desired style. Inside the discriminator, the stylized model is fed with both desired and undesired styles, and the discrimination function is implemented via Bayes rule by the contrast of the outputs corresponding to these two styles.

linguistic style  $s$ , where  $T$  is the length of the sentence. Our framework consists of a factual image captioning model and a generative style discriminator. The whole framework is illustrated in Fig. 2, where any off-the-shelf factual image captioning model can serve as the factual model.

When predicting the  $t$ -th word  $y_t$ , the factual model is fed the image  $x$  and the generated words  $y_{1:t-1}$ , and then gives the probability distribution of the next word, namely  $P(y_t | x, y_{1:t-1})$ . In the meanwhile, the discriminator is fed the style label  $s$ , the image  $x$  and the generated words  $y_{1:t-1}$ , and then predicts the probability that all candidates of next word align with the label style  $s$ , namely  $P(s | x, y_{1:t-1}, y_t)$ . Then the probability distribution we desire  $P(y_t | s, x, y_{1:t-1})$  can be computed according to Formula 1. The derivation process is in appendix.

$$P(y_t | s, x, y_{1:t-1}) = \frac{P(s | x, y_{1:t-1}, y_t) P(y_t | x, y_{1:t-1})}{P(s | x, y_{1:t-1})}. \quad (1)$$

We aim for the factual model to play a primary role when generating factual words, with minimal interference from the style discriminator. Conversely, when generating stylized words, we expect the style discriminator to increase the probability of appropriate stylistic words. However, in our experiments, we find that because the factual model is trained only on factual data, it assigned lower scores to stylized words. When tasked with generating stylized words, the style discriminator cannot easily alter the output of the factual model based on Eq. 1. Therefore, we introduced a weight parameter  $w$  to bias generation more strongly towards the style. If the value of  $w$  is too large, the style discriminator

may interfere with the normal operation of the factual model when generating factual words. Therefore, choosing a suitable value for  $w$  is important to generating good stylized captions. In subsequent experiments, we investigate the impact of different values of  $w$  on the performance of the framework. In addition, when predicting the  $t$ -th word, the input image  $x$ , the style label  $s$  and the generated words  $y_{1:t-1}$  are known, therefore we ultimately compute the weighted posterior of  $P(y_t|x, y_{1:t-1})$  and  $P(s|x, y_{1:t-1}, y_t)$  to obtain  $P(y_t|s, x, y_{1:t-1})$  according to Formula 2.

$$P(y_t|s, x, y_{1:t-1}) \propto P(y_t|x, y_{1:t-1})P(s|x, y_{1:t-1}, y_t)^w. \quad (2)$$

### 3.2 Generative Style Discriminator

Compared to standard (non-generative) discriminator, which takes the entire sentence as input and outputs a corresponding score, generative discriminator, by receiving generated words and outputting scores for all possible candidate words, evidently saves computation<sup>1</sup>. In our framework, the function of the generative style discriminator is to give  $P(s|x, y_{1:t-1}, y_t)$  while the function of factual model is to give  $P(y_t|x, y_{1:t-1})$ . And any off-the-shelf factual image captioning model can serve as factual model. Then in this section, we show how the stylized model forms the generative style discriminator via Bayes rule and how the discriminator guides the factual model to generate stylized image captions.

Given the style label  $s$ , the image  $x$  and the generated words  $y_{1:t-1}$ , the stylized model can output  $P(y_t|s, x, y_{1:t-1})$ . And  $P(y_{1:t-1}|s, x)$  has been calculated in last time step, therefore it can further calculates  $P(y_{1:t-1}, y_t|s, x)$ . Then we use  $s$  and  $\bar{s}$  to represent the desired and undesired styles, respectively. Then according to Bayes rule and the law of total probability, the probability distribution  $P(s|x, y_{1:t-1}, y_t)$  can be transformed into:

$$P(s|x, y_{1:t-1}, y_t) = \frac{P(s, x, y_{1:t-1}, y_t)}{\sum_{s' \in \{s, \bar{s}\}} P(s', x, y_{1:t-1}, y_t)}. \quad (3)$$

Because the style  $s$  and the image  $x$  are independent of each other, that is, the occurrence or non-occurrence of one does not affect the other, we can consider that, for any given  $s' \in \{s, \bar{s}\}$ ,  $P(s', x) = P(s')P(x)$ , and then  $P(s', x, y_{1:t-1}, y_t)$  can be transformed into:

$$P(s', x, y_{1:t-1}, y_t) = P(s')P(x)P(y_{1:t-1}, y_t|s', x). \quad (4)$$

The distribution  $P(s|x, y_{1:t-1}, y_t)$  can be further transformed into formula 5, in which  $P(s)$  can be assumed to be a constant for uniform styles, learned or set manually as a hyper-parameter.

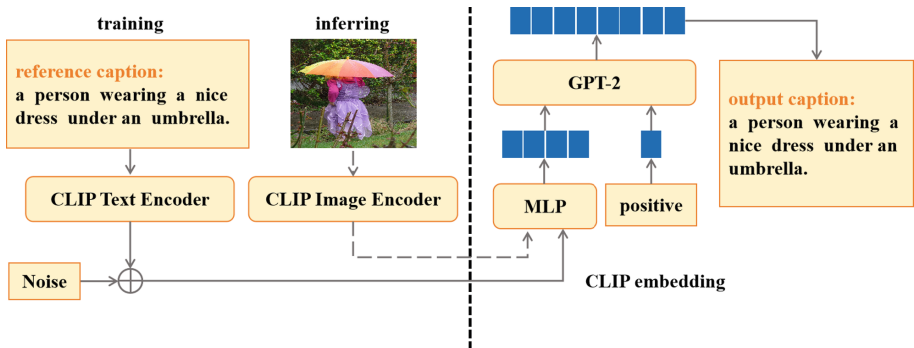
$$P(s|x, y_{1:t-1}, y_t) = \frac{P(s)P(y_{1:t-1}, y_t|s, x)}{\sum_{s' \in \{s, \bar{s}\}} P(s')P(y_{1:t-1}, y_t|s', x)}. \quad (5)$$

<sup>1</sup> The more detailed explanation is in the appendix.

This way, we can use the stylized model to form the generative style discriminator. As shown in Fig. 2, when generating stylized words, the factual model desires to generate words that still describe the image content. The stylized model with desired and undesired styles desires to generate words corresponding to the styles, and then according to formula 5, uses the contrast between the two output distributions to get  $P(s|x, y_{1:t-1}, y_t)$ , where the classification probability of desired words and undesired words will be increased and decreased, respectively. Finally we can compute the weighted posterior of  $P(y_t|x, y_{1:t-1})$  and  $P(s|x, y_{1:t-1}, y_t)$  to obtain  $P(y_t|s, x, y_{1:t-1})$  according to formula 2. In addition, when generating factual words, the difference between the outputs of the stylized model with desired and undesired styles is small and the discriminator dose not significantly impact the factual model. Therefore, our generative style discriminator can incorporate style elements in appropriate positions of the captions.

### 3.3 Stylized Model

The structure of stylized model is illustrated in Fig. 3, which takes the pre-trained language model GPT-2 as the main body. For the input style, it will use the embedding layer of GPT-2 to convert its tokens into its corresponding embedding vector. For the input image, it uses CLIP image encoder to extract its CLIP embedding vector and uses a simple multi-layer perceptron (MLP) to map it to the same space as the style embedding vector. Then the style embedding vector and image embedding vectors will be concatenated and used as the prefix input for GPT-2 to generate the caption with style.



**Fig. 3.** Structure of stylized model. The left part shows the encoding process, it will give the CLIP image embedding when inferring and the CLIP text embedding with noise when training. The right part shows the decoding process. The GPT-2 will decode the concatenation of style feature vector and CLIP feature vectors into the corresponding caption.

To train the stylized model only using unpaired stylized corpus, we use the CLIP embedding as the medium between vision and text to align images and

captions. The stylized model, during training and inference, will reconstruct texts from their respective CLIP text and image embedding. But we find CLIP embeddings of images and their corresponding captions are not interchangeable, which means there is a gap between the CLIP embeddings of images and captions. To solve the problem, we inject noise into the embedding during training, as suggested in [20].

When training the stylized model, the MLP and GPT-2 are fine-tuned together, and the overall training loss  $L$  consists of the standard generative language modeling loss  $L_g$  and a discriminative loss  $L_d$  as suggested in [11]:

$$L_g = \frac{1}{N} \sum_{i=1}^N -\frac{1}{T_i} \log P(y_{1:T}^i | s^i, x^i) \quad (6)$$

$$L_d = \frac{1}{N} \sum_{i=1}^N -\log P(s^i | x^i, y_{1:T}^i) \quad (7)$$

$$L = \lambda L_g + (1 - \lambda) L_d \quad (8)$$

where  $\lambda$  is a weight balancing the generative loss and discriminative loss,  $N$  is the number of samples, and  $T$  is the length of the captions.

## 4 Experiment

### 4.1 Dataset and Experimental Settings

We conduct experiments on two stylized image captioning datasets, including FlickrStyle10K [7] and SentiCap [17]. FlickrStyle10K contains 10K Flickr [28] images with stylized captions, where only the 7K training set are public and each image is labeled with 5, 1, and 1 captions for factual, humorous, and romantic styles, respectively. Following [8], we randomly select 6,000 and 1,000 of them as the training and test sets, respectively. SentiCap contains 2360 MSCOCO [16] images with 5013 positive captions and 4500 negative captions. The positive and negative subsets contain 998/673 and 997/503 images for training/testing respectively, and we split 100 samples from the training set for validation.

We evaluate our method in three aspects: the relevance with input images, the fluency, and accuracy of style. BLEU-1, BLEU-3, METEOR, and CIDEr [16] are used to evaluate the relevance between the generated captions and the input images. They are mostly calculated based on n-gram overlap between candidates and ground truth captions, and are the widely used automatic evaluation metrics in image captioning task. And we also utilize CLIPScore and RefCLIPScore [9] for evaluation in the ablation experiments. Compared to the reference-based n-gram overlap metric, they are more sensitive to detect potentially subtle inaccurate details in captions and more correlated with human judgments. Then we employ the average perplexity (ppl) to evaluate the fluency of generated captions. Following [4], we use a language modeling toolkit SRILM [24] to calculate the ppl. To evaluate the accuracy of style of generated captions, the style classification accuracy (cls) is adopted. The cls represents the proportion of generated

captions that align with the desired style. Following [4], we train a logistic regression classifier using stylized captions and factual captions for each of the four styles. The trained classifiers is used to classify the style of generated captions and the average accuracy of them can reach 96%.

We choose ClipCap [19] as our factual model. For the stylized model, we use the ViT-L/14 backbone for CLIP encoder and the GPT-2(small) for decoder. When training the stylized model, only the captions in stylized image captioning datasets is used. The number of training epochs and batch size is set to 20 and 64. The weight  $\lambda$  and noise variance is set to 0.8 and 0.016. Positive style and negative style each serve as the undesired style for the other on SentiCap dataset, and factual style serve as the undesired style for both humorous and romantic styles on FlickrStyle10k dataset. When inferring, the weight  $w$  is set to 300, 100, 30, 39 for positive, negative, romantic, humorous, respectively. In addition, for a fair experimental result, we retrain the factual model on the MSCOCO training set that has removed the data existing in the SentiCap test set when conducting experiments on SentiCap.

## 4.2 Comparison with State-of-the-Art

**Performance Comparisons.** We compare our framework with several state-of-the-art semi-supervised methods for stylized image captioning, including MSCap, Memcap and ADS-Cap. And in order to reflect the recent advances of stylized image captioning, we also add some supervised methods to Table 1, namely SF-LSTM, SAN and TridentCap. The supervised methods use paired image-caption data for training, where SAN also incorporates retrieval augmentation and data augmentation (DA), while TridentCap employs trident image-factual-style data for training and also requires a factual captioning model to provide a factual caption as input to the decoder during inference. Compared to them, for the factual captioning data, the semi-supervised methods also require paired image-caption data. But for the stylized captioning data, only the unpaired stylized corpus will be used. While our framework only requires training a stylized model on unpaired stylized corpus, it also needs a off-the-shelf factual image captioning model to ensure the fidelity of the image content.

Table 1 shows the performance comparison results including the relevance with input images, the fluency, and accuracy of style in four styles. We have observations that, when the cls exceeds 90%, meaning that the generated captions essentially align with the desired styles, our framework can achieve better results than earlier semi-supervised methods (MSCap and MemCap) in the relevance with input images (measured by Bleu-1, Bleu-3, CIDEr and METEOR), and achieve comparable performance to the state-of-the-art method (ADSCap), which means our framework can accurately describe the content of images. And we notice that, our method has a low ppl on SentiCap dataset, indicating good fluency in generated captions, it is high on FlickrStyle10K dataset, for reasons we give later. In addition, even only using unpaired stylized corpus, our framework can achieve comparable performance comparable to the supervised method SF-LSTM, which indicates our framework can be easily applied to a broader range of



**Table 1.** Performance comparisons on the test splits of positive (pos), negative (neg), romantic (roman), and humorous (humor) styles. For metric ppl, the lower value is better. For other metrics, the higher value is better. The top two scores for each metric are bolded and underlined, respectively.

Method		Style	Bleu-1	Bleu-3	METEOR	CIDEr	ppl(↓)	cls
Supervised	SF-LSTM [3] (Paired)	pos	50.5	19.1	16.6	60.0	–	–
		neg	50.3	20.1	16.2	59.7	–	–
		roman	27.8	8.2	11.2	37.5	–	–
		humor	27.4	8.5	11.0	39.5	–	–
	SAN [12] (DA)	pos	53.0	23.4	18.1	72.0	11.7	100.0
		neg	51.2	20.5	17.6	67.0	14.8	100.0
		roman	29.5	9.9	12.5	47.2	13.7	99.4
		humor	29.5	9.9	12.5	47.2	13.7	99.4
	TridentCap [25] (Trident)	pos	57.1	24.6	18.7	77.4	13.4	100.0
		neg	56.8	25.9	19.0	80.7	12.4	100.0
		roman	31.9	11.4	13.4	60.4	9.3	100.0
		humor	30.6	11.2	12.8	56.6	12.6	100.0
Semi-supervised	MSCap [8]	pos	46.9	16.2	16.8	55.3	19.6	92.5
		neg	45.5	15.4	16.2	51.6	19.2	93.4
		roman	17.0	2.0	5.4	10.1	20.4	88.7
		humor	16.3	1.9	5.3	15.2	22.7	91.3
	MemCap [29]	pos	51.4	17.0	16.6	52.8	18.1	96.1
		neg	49.2	18.1	15.7	59.4	18.9	<b>98.9</b>
		roman	19.7	4.0	7.7	19.7	19.7	91.7
		humor	19.8	<u>4.0</u>	7.2	18.5	17.0	97.1
	ADSCap [4]	pos	<u>52.5</u>	<u>18.9</u>	<u>18.5</u>	<u>64.8</u>	<b>13.1</b>	<b>99.7</b>
		neg	<b>52.3</b>	<b>21.0</b>	<b>18.0</b>	<b>65.1</b>	12.4	<u>98.2</u>
		roman	<b>25.6</b>	<b>6.7</b>	<b>10.9</b>	<b>33.1</b>	<b>10.6</b>	<b>95.9</b>
		humor	<b>23.7</b>	<b>6.3</b>	<b>10.3</b>	<b>31.6</b>	<b>12.8</b>	<b>97.3</b>
	Ours	pos	<b>53.3</b>	<b>20.3</b>	<b>18.6</b>	<b>68.1</b>	<b>13.1</b>	<u>97.0</u>
		neg	<u>51.5</u>	<u>19.0</u>	<u>16.9</u>	<u>62.7</u>	<u>14.7</u>	97.2
		roman	<u>22.3</u>	5.3	<u>10.2</u>	<u>32.6</u>	35.8	95.9
		humor	<u>21.3</u>	3.9	<u>9.2</u>	<u>27.5</u>	43.9	90.3

application scenarios without a heavy reliance on paired training data. We also note that compared to the updated supervised methods (SAN and TridentCap), our framework exhibits poorer performance and requires improvement.

**Table 2.** The time required for pre-training/retraining and fine-tuning, and the proportion of pre-training time in the entire training process.

method	pre-training/retraining	fine-tuning	proportion
ADSCap [4]	7h 26 m 49 s	37 m 20 s	92.3%
Ours	3h 40 m 33 s (optional)	19 m 55 s	91.7%

**Table 3.** The performance of plugging our framework into different factual models

method	SentiCap					FlickrStyle10k				
	C	ppl	cls	CLIPS	CLIPS <sup>Ref</sup>	C	ppl	cls	CLIPS	CLIPS <sup>Ref</sup>
CLIPCap( $w=0$ )	85.7	18.4	0.2	66.0	69.9	52.5	14.9	9.2	68.2	66.0
CLIPCap	65.4	13.9	97.1	59.6	64.9	30.1	39.9	93.1	63.7	62.6
PureT-XE( $w=0$ )	99.8	21.4	0.1	65.6	70.3	50.5	15.9	1.7	62.5	62.1
PureT-XE	72.0	16.9	97.5	58.7	64.4	23.7	43.7	86.3	57.3	57.4
PureT-SCST( $w=0$ )	112.2	15.7	0.3	65.9	71.1	53.8	14.2	1.0	62.9	62.8
PureT-SCST	90.3	16.5	96.7	60.5	66.7	28.1	43.8	85.8	58.2	58.3

**Efficiency Comparison.** We measure the time required for pre-training and fine-tuning of ADSCap and our framework, as shown in Table 2, where all experiments are conducted under the same settings to ensure consistency<sup>2</sup>. The cause of reducing training time in our framework is that, previous methods like ADSCap need to use the large-scale factual image-caption pairs to pretrain their model, which accounts for more than 90% of the total training time but our framework does not need the process. It should be emphasized that the pre-training time of our framework in Table 2 actually refers to the training time of the factual model. For a fair experimental result, we retrain the factual model on the MSCOCO training set that has removed the data existing in the SentiCap test set, which is not required in practical applications. We record this data to illustrate that, it is due to the data scale that training using factual image-caption pairs will always occupy the majority of the total training time, regardless of the method used. And in our framework, any off-the-shelf factual image captioning model can be directly used as the factual model to generate factual captions for input images. Then we only need to use the small-scale unpaired stylized corpus to train the stylized model, and then use it to construct the generative style discriminator to guide the factual model generate stylized image captions, eliminating the need for the pre-training process and thus reducing training time by over 90%. Furthermore, the composition of our stylized model with pre-trained CLIP and GPT also enhances the efficiency of model training.

<sup>2</sup> As only ADSCap has published its code, we only measured its training time.

**Table 4.** The performance of factual model, style model, factual+style framework and PPCap framework on SentiCap dataset

method	positive					negative				
	C	ppl	cls	CLIPS	CLIPS <sup>Ref</sup>	C	ppl	cls	CLIPS	CLIPS <sup>Ref</sup>
factual model	86.0	18.0	0	66.1	70.2	85.3	18.6	0	65.9	69.5
style model	28.1	13.7	100.0	45.2	50.7	25.9	12.4	97.6	43.5	48.6
factual+style	41.1	14.0	82.7	51.1	56.1	40.9	13.6	81.1	51.1	55.3
PPCap	68.1	13.1	97.0	60.2	65.9	62.7	14.7	97.2	58.9	63.6

### 4.3 Ablation Study

**Performance with Different Factual Models.** To demonstrate the plug-and-play capability of our framework, we also plug our generative style discriminator into other factual models to observe the performance. The only potential issue that may arise is that, the vocabularies used by the discriminator and the factual model are different. To align the vocabularies of the discriminator and the factual model, We set the probability of words not used by the discriminator to 0 and discard the words not used by the factual model. We select PureT [26], a purely Transformer-based model, as the factual model. As mentioned in Sect. 2.2, it performs better than CLIPCap on MSCOCO test set. For each evaluation metric, we compute the average scores for all styles on each dataset. Table 3 presents the results. The results on SentiCap align with our expectations. Specifically, we utilize the existing knowledge of pre-trained factual image captioning model to generate accurate captions. Therefore, models with better performance can achieve higher scores within this framework. We also noticed that, compared to CLIPCap, plugging into PureT does not achieve better CIDEr scores on FlickrStyle10k. We set weight  $w$  to 0, so the factual model is not influenced by the discriminator. Then we find that although PureT achieves better CIDEr scores on MSCOCO, it does not perform better on FlickrStyle10k. We think this is because FlickrStyle10k’s styles are more complex with more style-related words in captions and the factual model itself does not generate words related to style. And the CLIPScore indicate that the generated captions actually align semantically with the images content. Overall, PPCap can be applied to any auto-regressive factual models to achieve stylized image captioning, as long as it can give the probability distributions output at each time step.

**The Role of the PPCap Framework.** In order to show the role of the PPCap framework, we compare the performance of factual model, style model, factual+style framework and PPCap framework on SentiCap dataset, as shown in Table 4 and Fig. 4<sup>3</sup>. It is evident that the factual model can accurately describe the content of the image and the stylized model can achieve a high classification

<sup>3</sup> More examples are in the appendix.



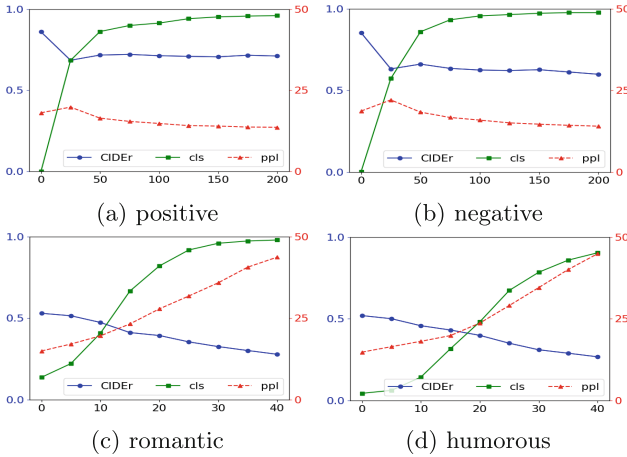
**FM:** a kitchen with a stove, refrigerator and sink.  
**SM:** a **nice** room with **black leather sofas wood tables**.  
**FS:** a kitchen with a **beautiful** display of **knives**.  
**PPCap:** a **nice** room with a sink, oven, and refrigerator.

**FM:** a hospital room with a bed and two beds.  
**SM:** a **nice** room with **black leather sofas wood tables**.  
**FS:** a **nice** room with a sink, toilet, shower, and mirror.  
**PPCap:** a **nice** room with a bed with a big screen tv.

**Fig. 4.** Two examples, each consisting of an image and its corresponding positive captions generated by the factual model (FM), stylized model (SM), factual+style framework (FS) and PPCap. The words related to the desired style and unrelated to the image content are highlighted in red and blue, respectively. (Color figure online)

score (cls). However, as shown in Fig. 4, the stylized model can only generate approximate captions for images, with some content unrelated to the image, namely hallucination. It may even produce identical captions for two similar but different images. We consider this is because training solely on unpaired stylistic corpus makes the model focus more on linguistic styles and thus causes inconsistency with images. On one hand, the scale of stylistic corpus is too small. On the other hand, the CLIP model does not perfectly align images and texts and substituting text for images in training can not yield satisfactory results, as demonstrated in [15]. Then, we hope to design a framework that can achieve good scores in both content and style. But simple weighted combination (factual+style) not only fails to effectively integrate style into captions but also retains hallucination. In contrast, PPCap can achieve high scores in both content and style simultaneously. It decouples the task of stylized image captioning into generating factual captions and incorporating style elements into the captions. The function of the generative style discriminator is only to incorporate style elements in appropriate positions of the factual captions. The hallucination mainly comes from the factual model and the pre-trained factual model is capable of accurately describing the content of images. Therefore, PPCap can improve stylistic accuracy without introducing incorrect content and the hallucination problem in our framework is not severe. We also use the new evaluation metrics CLIPScore and RefCLIPScore, which utilize the pre-trained CLIP model to directly compute the similarity between images and captions, thus making them more sensitive to detecting potentially subtle inaccuracies in captions. As shown in Table 4, after incorporating stylized elements, PPCap can achieve scores similar to those of the factual model.

**The Impact of Different Values of  $w$ .** In addition, we conduct experiments on the impact of the generative style discriminator with different weights  $w$  on the factual model. The results are shown in Fig. 5. For each kind of styles,



**Fig. 5.** The impact of the discriminator with different weights  $w$  on the factual model, namely the changes in CIDEr, cls, and ppl with the variation of  $w$ .

when the weight  $w$  is set to 0, the generated caption becomes a factual caption, where cls approaches 0 and CIDEr shows a relatively high value. Then, as the value of the weight  $w$  increases, style factors are gradually incorporated into the generated captions, where cls increases and CIDEr decreases. And when the cls exceeds 90%, meaning that the generated captions essentially align with the desired styles, CIDEr still maintains a relatively high value, which implies that the generated captions still can accurately describe the content of the images. This demonstrates the effectiveness of our framework, wherein the factual model is employed to ensure the fidelity of the image content, guided by the discriminator to incorporate style into factual captions, resulting in stylized image captions.

#### 4.4 Limitations and Discussion

While efficient, the performance of PPCap shows a slight gap compared to the SOTA method in the same setting. Especially on FlickrStyle10K, the ppl is high and increases as  $w$  increases in Fig. 5. Our explanation is as follows: the discriminator guides the factual model by leveraging the contrast between different styles. On SentiCap, there is a noticeable difference between positive captions and negative captions. But on FlickrStyle10k, the styles are implied in the whole sentence and the difference between stylized captions and factual captions is not as pronounced, which results in the discriminator disrupting the fluency of generated captions while guiding the factual model. In future research, we aim to train a stylized model that considers multiple styles as undesired styles, highlighting the distinctions between various styles to further enhance the performance of the framework. Furthermore, generating long and detailed stylized captions is more practically meaningful, and the emergence of large vision language models (LVLMs) has provided opportunities for this. However, discriminators trained

only on short texts cannot effectively guide LVLMS. In the future, we plan to explore constructing long stylized corpus to train the discriminator for improving combination with LVLMS.

## 5 Conclusion

In this paper, we propose a novel Plug and Play framework PPCap for efficient stylized image captioning, where only a stylized image captioning model needs to be trained on the small-scale unpaired stylized corpus. Then It can function as a generative style discriminator by Bayes rule and guide an off-the-shelf factual image captioning model to generate accurate stylized captions. Experimental results on two widely used stylized image captioning datasets demonstrate that our framework achieves outstanding performance while reducing training time by over 90%. In our future work, we aim to enhance the ability of PPCap to capture differences between different styles to further improve its performance.

**Acknowledgements.** This work is supported in part by the National Natural Science Foundation of China (No. 62106037, No. 62076052), and in part by the Major Program of the National Social Science Foundation of China (No.19ZDA127).

## References

1. Brown, T., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)
2. Chan, A., Ong, Y.S., Pung, B., Zhang, A., Fu, J.: CoCon: a self-supervised approach for controlled text generation. arXiv preprint [arXiv:2006.03535](https://arxiv.org/abs/2006.03535) (2020)
3. Chen, T., et al.: “factual” or “emotional”: stylized image captioning with adaptive learning and attention. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 519–535 (2018)
4. Cheng, K., Ma, Z., Zong, S., Zhang, J., Dai, X., Chen, J.: ADS-Cap: a framework for accurate and diverse stylized captioning with unpaired stylistic corpora. In: CCF International Conference on Natural Language Processing and Chinese Computing, pp. 736–748. Springer (2022)
5. Dathathri, S., et al.: Plug and play language models: a simple approach to controlled text generation. arXiv preprint [arXiv:1912.02164](https://arxiv.org/abs/1912.02164) (2019)
6. Fei, Z., Fan, M., Zhu, L., Huang, J., Wei, X., Wei, X.: Uncertainty-aware image captioning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 614–622 (2023)
7. Gan, C., Gan, Z., He, X., Gao, J., Deng, L.: StyleNet: generating attractive visual captions with styles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3137–3146 (2017)
8. Guo, L., Liu, J., Yao, P., Li, J., Lu, H.: MSCap: multi-style image captioning with unpaired stylized text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4204–4213 (2019)
9. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: ClipScore: a reference-free evaluation metric for image captioning. arXiv preprint [arXiv:2104.08718](https://arxiv.org/abs/2104.08718) (2021)

10. Keskar, N.S., McCann, B., Varshney, L.R., Xiong, C., Socher, R.: Ctrl: a conditional transformer language model for controllable generation. arXiv preprint [arXiv:1909.05858](https://arxiv.org/abs/1909.05858) (2019)
11. Krause, B., et al.: Gedi: generative discriminator guided sequence generation. arXiv preprint [arXiv:2009.06367](https://arxiv.org/abs/2009.06367) (2020)
12. Li, G., Zhai, Y., Lin, Z., Zhang, Y.: Similar scenes arouse similar emotions: parallel data augmentation for stylized image captioning. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 5363–5372 (2021)
13. Li, J., Vo, D.M., Sugimoto, A., Nakayama, H.: Evcap: retrieval-augmented image captioning with external visual-name memory for open-world comprehension. arXiv preprint [arXiv:2311.15879](https://arxiv.org/abs/2311.15879) (2023)
14. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning, pp. 19730–19742. PMLR (2023)
15. Liang, V.W., Zhang, Y., Kwon, Y., Yeung, S., Zou, J.Y.: Mind the gap: understanding the modality gap in multi-modal contrastive representation learning. In: Advances in Neural Information Processing Systems, vol. 35, pp. 17612–17625 (2022)
16. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
17. Mathews, A., Xie, L., He, X.: Senticap: generating image descriptions with sentiments. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
18. Mathews, A., Xie, L., He, X.: Semstyle: learning to generate stylised image captions using unaligned text. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8591–8600 (2018)
19. Mokady, R., Hertz, A., Bermano, A.H.: Clipcap: clip prefix for image captioning. arXiv preprint [arXiv:2111.09734](https://arxiv.org/abs/2111.09734) (2021)
20. Nukrai, D., Mokady, R., Globerson, A.: Text-only training for image captioning using noise-injected clip. arXiv preprint [arXiv:2211.00575](https://arxiv.org/abs/2211.00575) (2022)
21. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
22. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
23. Ramos, R., Martins, B., Elliott, D.: LMCap: few-shot multilingual image captioning by retrieval augmented language model prompting. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 1635–1651. Association for Computational Linguistics, Toronto, Canada (2023). <https://aclanthology.org/2023.findings-acl.104>
24. Stolcke, A.: Srlm-an extensible language modeling toolkit. In: Seventh International Conference on Spoken Language Processing (2002)
25. Wang, L., Qiu, H., Qiu, B., Meng, F., Wu, Q., Li, H.: TridentCap: image-fact-style trident semantic framework for stylized image captioning. IEEE Trans. Circuits Syst. Video Technol. **34**(5), 3563–3575 (2024). <https://doi.org/10.1109/TCSVT.2023.3315133>
26. Wang, Y., Xu, J., Sun, Y.: End-to-end transformer based model for image captioning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 2585–2594 (2022)

27. Yang, K., Klein, D.: Fudge: controlled text generation with future discriminators. arXiv preprint [arXiv:2104.05218](https://arxiv.org/abs/2104.05218) (2021)
28. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. *Trans. Assoc. Comput. Linguist.* **2**, 67–78 (2014)
29. Zhao, W., Wu, X., Zhang, X.: MemCap: memorizing style knowledge for image captioning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12984–12992 (2020)





# Harlequin: Color-Driven Generation of Synthetic Data for Referring Expression Comprehension

Luca Parolari<sup>1</sup>, Elena Izzo<sup>1</sup>, and Lamberto Ballan<sup>1</sup>

University of Padova, Padua, Italy

{luca.parolari,elena.izzo}@phd.unipd.it, lamberto.ballan@unipd.it

**Abstract.** Referring Expression Comprehension (REC) aims to identify a particular object in a scene by a natural language expression, and is an important topic in visual language understanding. State-of-the-art methods for this task are based on deep learning, which generally requires expensive and manually labeled annotations. Some works tackle the problem with limited-supervision learning or relying on Large Vision and Language Models. However, the development of techniques to synthesize labeled data is overlooked. In this paper, we propose a novel framework that generates artificial data for the REC task, taking into account both textual and visual modalities. At first, our pipeline processes existing data to create variations in the annotations. Then, it generates an image using altered annotations as guidance. The result of this pipeline is a new dataset, called *Harlequin*, made by more than 1M queries. This approach eliminates manual data collection and annotation, enabling scalability and facilitating arbitrary complexity. We pre-train three REC models on Harlequin, then fine-tuned and evaluated on human-annotated datasets. Our experiments show that the pre-training on artificial data is beneficial for performance.

**Keywords:** Synthetic Data Generation · Referring Expression Comprehension · Visual Grounding.

## 1 Introduction

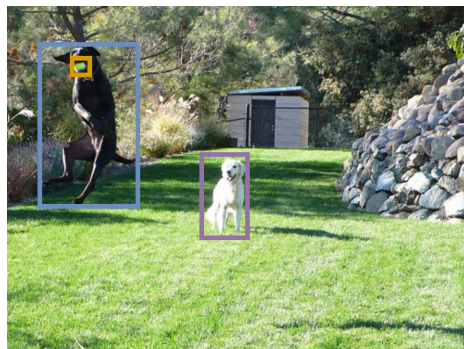
The expressiveness and variety of the human language are the basis of communication between people. Their ability to interact and understand each other attracts researchers to design models able to communicate with them. In this context, the task of Referring Expression Comprehension (REC) [42], also known as Visual Grounding [7, 20] or Phrase Localization [29, 34], aims to identify a specific object in a scene described by a phrase, called referring expression or sometimes query. The research progress in this task has been made possible thanks to the active development of datasets. Since 2015, Flickr30k Entities [27], ReferIt [18], RefCOCO, and two variants RefCOCO+ and RefCOCOG, [26, 43] were released.

These datasets are human-labeled and consist of triplets composed of an image, a referring expression, and a bounding box. Fig. 1 shows an example. However, the gathering and annotation of such data is time-consuming and resource-intensive, representing a critical bottleneck for the collection of sufficiently large training sets and new benchmarks.

Current works face this issue exploring limited supervision learning techniques such as weakly-supervised [28], semi-supervised [15], and unsupervised [34] or rely on large Vision and Language models pre-trained on a massive amount of multimodal data [17]. However, the development of techniques and pipelines to create new, reasoning-oriented datasets is overlooked, limited by fine-grained annotations required by the Referring Expression Comprehension task. Some works explore the generation of the queries by either working on their properties or structure [5, 14, 33]. However a method to generate both queries and images has not been investigated yet.

In this paper, we propose a pipeline for generating synthetic data for the Referring Expression Comprehension task, taking into account both textual and visual modalities. Recent developments in text-to-image generation with diffusion models allowed fine-grained control over the output by either embedding guidance signals like bounding box, keypoints, or semantic maps with language [21] or even expressing them by means of text [37]. Inspired by these advancements, we argue that (i) the process of manual collection and annotation of data for this task can finally be avoided, and (ii) new benchmarks with arbitrary size and complexity can be created. The proposed pipeline and extensive experiments we run address those hypotheses.

Broadly speaking, our pipeline is composed of two modules. The first is the Annotation Generation Engine. It is responsible for generating new referring expressions (REs) with consistent bounding box annotations. We use REs from Flickr30k Entities as seeds and generate their variations to keep consistency with the arrangement of objects in the image. REs are altered by varying their attributes, specifically the color attribute. The second is the Image Generation Engine. Guided by the annotation obtained in the previous step, it generates a new image. The synthesized image should represent the given caption and depict objects at specific locations that look like the given description. Objects are described through referring expressions, which may have varied attributes.



A white dog looks at another dog catching a ball in the air

**Fig. 1.** Annotations required by the Referring Expression Comprehension task. In this example, the image has one caption with three referring expressions. Each referring expression is accompanied by the location of the referred object (bounding box).

Following this strategy, we synthetically generate Harlequin, a new dataset consisting of train, validation, and test sets. Harlequin is the first dataset totally synthetic generated for the Referring Expression Comprehension task. The experiments show that its use in pre-training stage boost the results on real data, reducing labeling effort and errors in annotations.

Our contributions can be summarized as follows:

(i) We propose a novel pipeline for generating synthetic data for the Referring Expression Comprehension task, increasing richness and variability and reducing to zero the human effort required for collecting annotations; (ii) We introduce Harlequin, a new dataset for the Referring Expression Comprehension task, which is entirely synthetically generated; (iii) We prove the effectiveness of our synthetic dataset if used in a pre-training stage to transfer knowledge on real datasets; (iv) We release both the dataset and the code.<sup>1</sup>

## 2 Related Work

*Referring Expression Comprehension* Among different approaches studied in literature [38, 42], recently the transformer-based approach emerged, demonstrating superior performance. TransVG [7] makes use of transformer for both intra- and inter-modality correspondence. VLTVG [36] employs a language-guided context encoder to extract discriminative features of the referred object. QRNet [40] introduces query-aware dynamic attention to extract query-refined visual features with a hierarchical structure. VG-LAW [32] adds adaptive weights to the visual backbone to make it an expression-specific feature extractor. LGR-NET [25] emphasizes the guidance of the referring expression for cross-modal reasoning. InterREC [35] increases object-level relational-level interpretability through an image semantic graph and a reasoning order tree.

*Synthetic Data Generation* In the last decade different areas of research started to investigate the use and generation of synthetic data to lower the cost of data and automation collection. In [12], the authors argued the interchangeability between real and synthetic datasets and demonstrated the improvements of performance pre-training the models on virtual data encouraging the generation of synthetic data in various domains such as autonomous driving [12], gardening [19], deepfake detection [1], object detection and 3D reconstruction [31]. In many cases, datasets were created by means of simulators which guarantee complete control over synthetic environments such as Unity [16], Blender [3] and CARLA [9]. Newer trends instead employ generative models to increase the automation in data generation and labeling process [1].

*Text-to-image Generation* Diffusion-based models demonstrated astonishing abilities in generating complex and realistic images. Recently, the existing pre-trained text-to-image diffusion models allowed fine-grained control in image generation, specifying requirements at the level of bounding boxes, masks, and

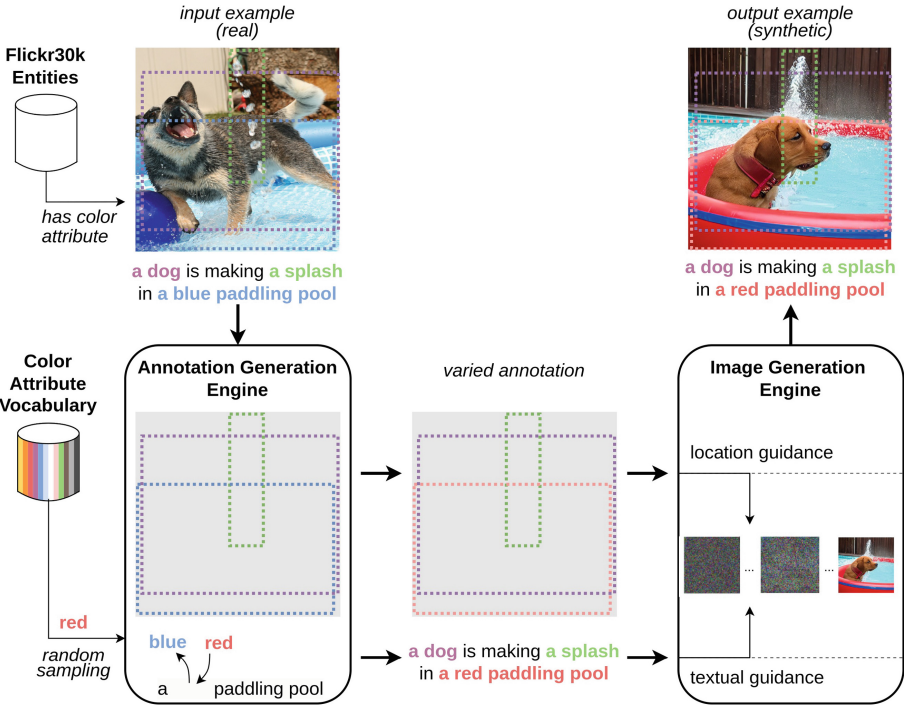
<sup>1</sup> <https://github.com/lparolari/harlequin>

edge or depth maps. For example, GLIGEN [21] uses a pre-trained T2I diffusion model and, freezing its weights, injects the grounding information into new trainable layers via a gated mechanism, focusing primarily on bounding boxes as the grounding condition. Similarly, ReCo [39] extends stable diffusion, adding position tokens to enable open-ended regional texts for high-level region control. Finally, ControlNet [45] introduces conditional control connecting the trainable copy and the large pre-trained text-to-image diffusion models via “zero convolution” layers to eliminate harmful noise during training.

### 3 Human-annotated Datasets for Referring Expression Comprehension

The most popular datasets in the field are Flickr30k Entities, ReferIt, and especially RefCOCO family. All these datasets were built on top of existing sets of captioned images, and they were human-annotated to align a referring expression with the bounding box of the mentioned entity in the image. In particular, Flickr30k Entities dataset was built to augment Flickr30k [41] image captions with 244k coreference chains yielding almost 276k bounding boxes in 32k images. ReferIt, which contains images from the TC-12 expansion of the ImageCLEF IAPR dataset [10], has 131k expressions in 20k photographs of natural scenes. RefCOCO, RefCOCO+, and RefCOCOg were built on top of MSCOCO dataset [22]. RefCOCO and RefCOCO+ count 142k referring expressions in 20k images, instead RefCOCOg counts 85k expressions in 27k images. They aimed to collect images with multiple instances of the same object class to increase the complexity. Besides, RefCOCOg focused on rich and natural descriptions, while RefCOCO and RefCOCO+ on appearance-based descriptions.

Although crowd-sourcing protocols allowed the collection of a noticeable amount of annotations, we believe that such a time-consuming and resource-intensive task severely limits the gathering of new datasets where generalization, adaptability, and reasoning properties can be learned and evaluated. In this paper, we investigate a pipeline for generating synthetic data for the Referring Expression Comprehension task, having control of both visual and textual content. As a starting point towards this direction, we decide to add some constraints in the generation of the data in order to properly validate the pipeline, the synthetic dataset and its applicability to real data. In particular, inspired by [12], we generate a dataset applying variations on the existing Flickr30k Entities one. Among others, we chose Flickr30k Entities for seed samples because every image is annotated with a sentence, yielding many referring expressions. From a generative point of view, this setting alleviates the amount of guessing and constrains the possible space of images that can be generated to a subset, where objects are precisely described and spatially located.



**Fig. 2.** Our pipeline. It processes existing samples from Flickr30k Entities data. We select the ones characterized by at least one *color* attribute in their referring expressions. The Annotation Generation Engine processes the sample’s caption, referring expressions and locations where the color attribute is replaced with a randomly chosen color. The caption is updated accordingly. Then, the Image Generation Engine creates the new image using new annotations provided by the Annotation Generation Engine as guidance for the generation.

## 4 The Proposed Pipeline

The proposed approach, depicted in Fig. 2, relies on two components to generate synthetic data for Referring Expression Comprehension. The former, termed **Annotation Generation Engine**, is in charge of creating annotations to guide image generation. The latter, named **Image Generation Engine**, is responsible for synthesizing images enforcing the guidance provided by the Annotation Generation Engine. Specifically, given an input annotation  $\mathbf{a}$  composed of an image caption  $\mathbf{c}$  and set of referring expressions along with referred object locations  $\{(q_i, l_i)\}_{i=1}^N$ , the Annotation Generation Engine produces new annotations by varying attributes in the  $p$ -th referring expression, with  $p \in [1, N]$ . Then, the Image Generation Engine uses the annotation provided by the Annotation Generation Engine to generate a synthetic image  $I$  exploiting GLIGEN [21], a generative model based on Stable Diffusion [30].

Since people frequently use colors to describe and disambiguate objects [18], we select *color* as the attribute to alter in the annotations. The color attribute has also proven to have a strong impact in several computer vision tasks, ranging from visual recognition problems (like object detection and image captioning) [24] to visual tracking [6]. Therefore, for each referring expression we generate several variations where the color attribute is replaced with a new color. This enhances richness and variability of the dataset, because with one single seed many other examples can be generated representing objects with different colors, and possibly new orientations, perspectives and views of the same scene. Moreover, altering the *color* attribute offers some advantages: (i) a simple variation of the textual content has a strong impact on the generated images, allowing the models to learn to disambiguate between similar scenes; (ii) this alteration does not affect the position of the object in the image, retaining the original layout of objects in the image; (iii) it is (relatively) easy to understand and manipulate by a generative model.

#### 4.1 The Annotation Generation Engine

The Annotation Generation Engine (AGE) is a function defined over the set of annotations  $\mathcal{A}$ . It is specifically designed for Referring Expression Comprehension task and produces compatible annotations by altering queries in existing samples:  $\phi : \mathcal{A} \rightarrow \mathcal{A}$ . The AGE component takes an annotation  $\mathbf{a}$  in input. The annotation consists of a caption  $\mathbf{c}$  and a non-empty set of entities  $E$ . Each entity is described by the textual form of a referring expression and the location of the referred object:

$$\text{Annotation: } \mathbf{a} = (\mathbf{c}, E) \quad (1)$$

$$\text{Caption: } \mathbf{c} = [c_1, \dots, c_L] \quad (2)$$

$$\text{Entities: } E = \{(\mathbf{q}_i, \mathbf{l}_i)\}_{i=1}^N \quad (3)$$

where  $\mathbf{c}$  is a caption of  $L$  tokens,  $N$  is the number of referring expressions,  $\mathbf{q}_i = [c_j, \dots, c_k]$  with  $1 \leq j \leq k \leq L$  is the textual representation of the referring expression from a subset of contiguous tokens in  $\mathbf{c}$ ,  $\mathbf{l}_i = [\alpha_{\min}, \beta_{\min}, \alpha_{\max}, \beta_{\max}]$  is with top-left and bottom-right coordinates of the referred object. The AGE returns a new annotation where the  $p$ -th referring expression is varied by replacing a color attribute,  $p \in [1, N]$ . The location is not altered. Tokens in the caption are updated accordingly to the new referring expression, while other referred objects are not varied and serve as context. Mathematically, the output of  $\phi(\mathbf{a})$  is  $\hat{\mathbf{a}} = (\hat{\mathbf{c}}, \hat{E})$  where

$$\hat{\mathbf{c}} = [c_1, \dots, c_{j-1}, \overbrace{\hat{c}_j, \dots, \hat{c}_k}^{\hat{q}_p}, \dots, c_L] \quad (4)$$

$$\hat{E} = \{\hat{\mathbf{q}}_p, \mathbf{l}_p\} \cup \{(\mathbf{q}_i, \mathbf{l}_i)\}_{i=1, i \neq p}^N \quad (5)$$

with  $\hat{\mathbf{q}}_p = [\hat{c}_j, \dots, \hat{c}_k]$  the new referring expression where the color attribute is changed. Specifically, we replace in  $\mathbf{q}_p$  the color with a new randomly sampled

one. Sampling is done on a vocabulary  $C$  of 12 color attributes based on [44]: black, gray, white, red, orange, yellow, green, cyan, blue, purple, pink, and brown. The variation function  $\phi$  is applied 6 times ( $|C|/2$ ) per referring expression with color attribute. We chose 6 as a trade-off between the number of annotations generated and the variability introduced through multiple sampling.

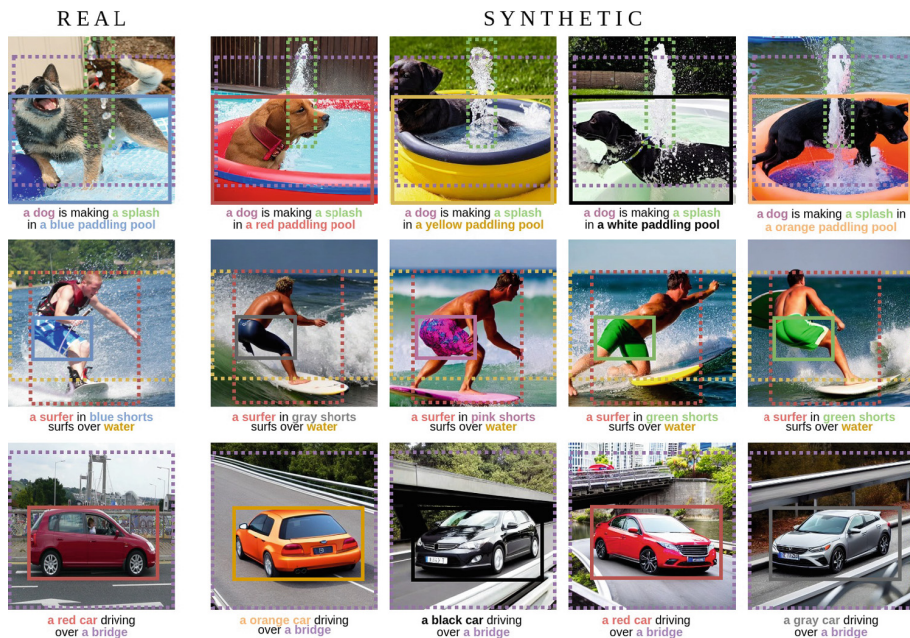
The current definition of  $\phi$  keeps fixed all objects’ locations and  $N - 1$  referring expressions. This is done to preserve the spatial arrangement of the objects, i.e. the layout and the image context. Objects’ locations are particularly relevant as they express complex semantic meaning. For example, the size of bounding boxes may express perspective: a person in the foreground should be bigger with respect to one in the background. Moreover, they could also identify relations: in the scene represented by “a person reading a book”, the bounding box of the book should be small but also close to the bounding box of the person. In order to keep this rich semantic, in this work we prefer to focus on variation of text, which is more intuitive to generate and evaluate.

## 4.2 The Image Generation Engine

The Image Generation Engine (IGE) is responsible for generating synthetic images. This component receives an annotation  $\hat{\mathbf{a}}$  obtained from the Annotation Generation Engine. It returns an image  $I \in \mathcal{I}$  from the domain of images encoding semantic information expressed in  $\hat{\mathbf{a}}$ . More in detail, we define the IGE as a function  $\psi : \mathcal{A} \rightarrow \mathcal{I}$ :  $\psi(\hat{\mathbf{a}}) = I$ .

We implement the Image Generation Engine component with Grounded-Language-to-Image Generation (GLIGEN) [21]. GLIGEN is a generative model based on Stable Diffusion [30], which is capable of generating detailed and high-quality images. Although the pipeline does not bind the IGE component with a specific generative model, we chose GLIGEN for different reasons. Unlike mainstream generative models, GLIGEN allows fine-grained control over the output image. This is a fundamental aspect because we are interested in providing samples for REC task. Specifically, we are interested in generating images that are coherent to the annotations, i.e. locations of the referred objects. For this reason, a critical feature of the chosen generator is the ability to guide the synthesizing process through “objects description”, beyond the image caption. That is, an image is generated by describing its content through a caption as in Stable Diffusion, but a set of pairs (referring expression, object location), namely entities, is also provided. These entities instruct GLIGEN with the objects’ location and information on their appearance features. The more accurate the positioning of objects and fidelity to descriptions, the better the supervision signal for the Referring Expression Comprehension task.

Although the main focus of GLIGEN is the conditioning on entities, i.e., description and location of objects, it can also work with other modalities: images, keypoints, hed map, canny map, semantic map, normal map. Every modality can be used to control the generation of the output image. In this work, we focus on the standard modality, which is compatible with the format of annotation  $\hat{\mathbf{a}}$  produced by the Annotation Generation Engine.



**Fig. 3.** Examples produced by our pipeline. On the left, we show reference images along with their annotations from Flickr30k Entities. On the right, we report some generated variations. Colors are altered and guide, along with objects’ locations, the image synthesis.

## 5 Harlequin Dataset

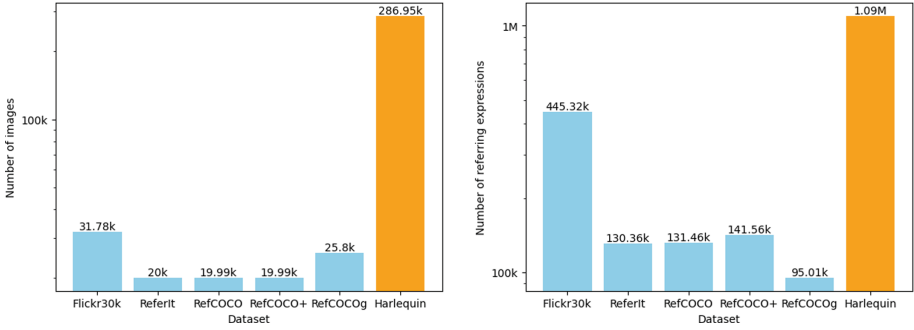
We introduce the first totally synthetic generated dataset for the Referring Expression Comprehension task, termed Harlequin,<sup>2</sup> collected via our pipeline. We report some examples in Fig. 3. The dataset is originated from Flickr30k Entities: we select samples characterized by referring expressions containing the *color* attribute to variate them. Since the Image Generation Engine is eager in terms of resources, we first generate all the new annotations with Annotation Generation Engine using the selected samples as seeds. Secondly, we run the image synthesis adopting a frozen instance of GLIGEN in “generation” mode with “text + box” modality and batch size 1.<sup>3</sup>

Harlequin comprises a total of 286,948 synthetic images and 1,093,181 annotations targeting color attributes and following the coco format. It has  $2.60 \pm 1.14$  words per referring expression on average, in line with Flickr30k Entities statistics. The median value is 2, while the longest referring expression is 14 words. Harlequin follows Flickr30k Entities’ data splits. It provides 988,342 annotations

<sup>2</sup> Harlequin, or Arlecchino in Italian, is a character from the Italian commedia dell’arte known for his colorful patched costume.

<sup>3</sup> <https://github.com/gligen/GLIGEN>.





**Fig. 4.** Dataset statistics. We report the number of images and referring expressions per dataset on the left and right, respectively. Harlequin is highlighted in orange.

over 259,930 images for the training set, 52,554 annotations over 13,584 images for the validation set, and 52,284 annotations over 13,434 images for the test set. Fig. 4 visualizes the amount of data in Harlequin compared to existing, manually annotated, and collected datasets. Harlequin doubles the amount of referring expression in Flickr30k, the largest dataset available in the literature, and provides a noticeably larger amount of images.

Harlequin presents some interesting properties. For instance, the generated images display the same objects under various orientations and on different backgrounds, increasing the variability and complexity of Harlequin with respect to Flickr30k Entities, while retaining its supervision signal (Fig. 3, third row). Moreover, we observe that our generation strategy fixes some errors in the human-annotated labels. As a matter of fact, we noticed that Flickr30k Entities contains some samples annotated with the wrong locations of the bounding boxes. The pipeline addresses this issue, generating new images coherent with the given annotations where the referred object is correctly inside the provided bounding box. Finally, we bring up that the used variation function  $\phi$  inevitably leads to the generation of unrealistic-colored objects (e.g. “the blue dog”). Independently of that, the results show that Referring Expression Comprehension models learn a robust representation from Harlequin. This is coherent with the fact that humans are usually capable of identifying an object regardless of its color and use this information to disambiguate similar objects.

## 6 Experiments

We present experimental results obtained in Referring Expression Comprehension by pre-training two models on Harlequin, our synthetic dataset, and fine-tuning them on realistic datasets. We show that the pre-training improves performance. We discuss the role that variations in original annotation play in the improvement of results, and finally, we analytically evaluate the contribution of the color variations through an ablation study.

## 6.1 Implementation details

We pre-train TransVG [7], VLTVG [36], and LGR-NET [25] on our synthetic dataset and then fine-tune them on RefCOCO family datasets initializing the weights of the model with those obtained after the pre-training. We followed the implementation details of TransVG, VLTVG, and LGR-NET. For all the experiments on TransVG and VLTVG, *pre-training* on Harlequin, *fine-tuning* on RefCOCO family, and from-scratch *baselines*, we initialized the weights of the visual transformers with those of DETR [4] based on the ResNet-50 [13] available on projects’ page. Instead, the linguistic branch is initialized with the weights of the BERT model [8]. We set the batch size to 32 and use AdamW as the optimizer. For the *pre-training* on Harlequin, we train both TransVG and VLTVG for 60 epochs, the value suggested by authors for Flickr30k, dropping the learning rate by a factor of 10 after 40 epochs. Instead, the *fine-tuning* experiments and supervised *baselines* are trained for 90 epochs with a learning rate dropped by a factor of 10 after 60 epochs. When using the TransVG model, we set the weight decay to  $10^{-4}$ , the initial learning rate of the vision-language module and prediction head to  $10^{-4}$ , and of the visual branch and linguistic branch to  $10^{-5}$ . When using the VLTVG model, the initial learning rate of the feature extraction branches is  $10^{-5}$  and  $10^{-4}$  for all the other components. Moreover, we freeze the weights of the visual and textual branches in the first 10 epochs. As concerns LGR-NET model, we pre-trained the model on Harlequin and the fine-tuning and supervised baselines experiments are trained for 15 epochs. We used Swin Transformer Small [23] as the backbone, BERT as the textual extractor, and followed the implementation details provided by the authors. During evaluation, we set batch size to 32. We carried out all the experiments on a single NVIDIA RTX A5000. We used the code provided online.<sup>4,5,6</sup>

## 6.2 Evaluation Models and Metrics

The models we chose for the evaluation of Harlequin on the Referring Expression Comprehension task are TransVG [7], VLTVG [36], and LGR-NET [25], as mentioned above. TransVG proposes an alternative prediction paradigm to directly regress the target coordinates. It makes use of transformer for both intra- and inter-modality correspondence. A regression token is added to the multi-modal transformer and is optimized through a regression head that directly outputs the object’s location. VLTVG employs a visual-linguistic verification mechanism alongside a language-guided context encoder to extract discriminative features of the referred object. The visual-linguistic verification module enhances visual features, emphasizing regions related to the referring expression, whereas the language-guided context encoder collects meaningful visual contexts. Ultimately, a multi-stage cross-modal decoder is utilized to iteratively analyze the encoded visual and textual features, refining the object representation for precise

<sup>4</sup> <https://github.com/djiajunustc/TransVG>

<sup>5</sup> <https://github.com/yangli18/VLTVG>

<sup>6</sup> <https://github.com/lmc8133/LGR-NET>.

**Table 1.** Results. We show the performance of three methods, TransVG, VLTVG and LGR-NET, on the Referring Expression Comprehension task with pre-training on Harlequin (*Synth*→*Real*) and without (*Real*). The pre-training shows superior or comparable performance on three benchmarks: RefCOCO, RefCOCO+ and RefCOCOg. We report the standard accuracy percentage.

Method	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test
<i>TransVG</i> [7]:								
Real	63.33	69.05	55.62	64.69	69.02	<b>55.76</b>	64.04	63.22
Synth→Real	<b>65.77</b>	<b>70.66</b>	<b>56.80</b>	<b>66.66</b>	<b>72.01</b>	55.66	<b>65.13</b>	<b>64.33</b>
(Improv.)	+2.44	+1.61	+1.18	+1.97	+2.99	-0.10	+1.09	+1.11
<i>VLTVG</i> [36]:								
Real	<b>69.66</b>	74.33	<b>61.35</b>	70.83	76.02	<b>61.71</b>	<b>70.57</b>	<b>70.03</b>
Synth→Real	69.60	<b>75.76</b>	61.14	<b>71.46</b>	<b>77.16</b>	61.30	70.04	69.57
(Improv.)	-0.06	+1.43	-0.21	+0.63	+1.12	-0.41	-0.53	-0.46
<i>LGR-NET</i> [25]:								
Real	82.71	85.77	79.31	71.11	75.45	63.35	70.75	71.11
Synth→Real	<b>84.38</b>	<b>87.13</b>	<b>80.67</b>	<b>71.40</b>	<b>75.60</b>	<b>64.70</b>	<b>74.61</b>	<b>75.22</b>
(Improv.)	+1.67	+1.36	+1.36	+0.29	+0.15	+1.35	+3.86	+4.11

target localization. LGR-NET emphasizes the guidance of textual features for cross-modal reasoning extending the standard textual features generating three embeddings: coordinate, word, and sentence. The textual features are, then, employed for alternated cross-modal reasoning exploiting a loss enhances the cross-modal alignment while localizing the referred object.

The evaluation metric is the standard accuracy. Given a referring expression, it considers a prediction to be correct if and only if the intersection over union between the predicted and the ground truth bounding box is at least 0.5.

### 6.3 Results

Tab. 1 shows the performance of TransVG [7], VLTVG [36], and LGR-NET [25] in Referring Expression Comprehension task. Specifically, we report the results obtained in two settings. In the first, we train the model from scratch on realistic datasets: RefCOCO, RefCOCO+, and RefCOCOg. In the second, we pre-train the model on Harlequin, our synthetic dataset, and then fine-tune it on realistic datasets. In both cases, we report the evaluation on the three RefCOCO datasets.

TransVG shows homogeneous improvement among all datasets. It improves by 2.44%, 1.61%, and 1.18% in RefCOCO splits and shows superior performance also in RefCOCO+ and RefCOCOg. For VLTVG, despite the model starts from a higher performance with respect to TransVG, it shows a remarkable 1.43% and 1.12% improvement on RefCOCO and RefCOCO+'s testA. As concerns

LGR-NET, we improve the supervised baselines on all the datasets reaching up to +3.86% and +4.11% on the RefCOCOg splits. We recall that the reported improvement emerges in a cross-dataset setting. As a matter of fact there is no overlap between the pre-training data, synthetically generated from Flickr30k Entities, and the fine-tuning datasets.

**Table 2.** Ablation study. We evaluate the performance of TransVG and VLTVG on a subset of test sets where referring expressions contain at least one color attribute. Column *% Anns* reports the percentage of annotations with at least a color attribute with respect to original test sets. Columns *Real* and *Synth→Real* show the performance without or with pre-training on Harlequin. We report standard accuracy in percentage.

Evaluation			TransVG [7]		VLTVG [36]	
Dataset		% Anns	Real	Synth→Real	Real	Synth→Real
RefCOCO	val (color)	23.2	64.35	<b>69.50 (+5.15)</b>	<b>77.15</b>	77.11 (-0.04)
	testA (color)	37.5	68.73	<b>72.91 (+4.18)</b>	79.46	<b>81.48 (+2.02)</b>
	testB (color)	17.8	56.46	<b>60.11 (+3.65)</b>	68.95	<b>70.28 (+1.33)</b>
RefCOCO+	val (color)	34.6	68.91	<b>70.07 (+1.16)</b>	75.79	<b>77.13 (+1.34)</b>
	testA (color)	37.5	71.63	<b>74.20 (+2.59)</b>	79.13	<b>80.95 (+1.82)</b>
	testB (color)	26.8	55.73	<b>57.33 (+1.60)</b>	<b>65.65</b>	63.82 (-1.83)
RefCOCOg	val (color)	41.4	62.37	<b>65.04 (+2.67)</b>	<b>73.23</b>	73.09 (-0.14)
	test (color)	41.7	62.12	<b>64.94 (+2.82)</b>	73.05	<b>73.70 (+0.65)</b>

The results demonstrate that pre-training on synthetically generated data is feasible in the Referring Expression Comprehension task. Annotations required by this task challenge generative models, where their artistic traits need to deal with fine-grained constraints on objects’ locations and descriptions. Nevertheless, our pipeline proves that the generation and collection of heavily annotated data with zero human effort is possible. This is an important milestone and opens a wide range of future directions where data can be crafted to overcome the increasing need for annotations. We argue that the artificial nature of data is overcome when the control over semantic properties is appropriately exploited. Merely generating a dataset may not imply good performance, especially if the model is tested on realistic benchmarks. The generated dataset must encode some knowledge that the model can learn in order to compete with real-world datasets.

#### 6.4 Impact of the Color Attribute

In this section, we evaluate the impact of our pre-training on realistic samples with the *color* attribute. We follow the same training scheme. However, here the test sets are limited to samples containing a referring expression with a color. As shown in Tab. 2, TransVG [7] demonstrates a boost in performance among

RefCOCO family datasets, with remarkable +5.15%, +4.18% and +3.65% on RefCOCO. The pre-training shows superior or comparable performance also for VLTVG [36], with the exception of testB for RefCOCO+. We recall that no changes to the model’s architecture have been made to encode extra knowledge about colors. The improvement is solely guided by learning patterns from data.

However, these results were expected. As a matter of fact, Harlequin is mainly composed of referring expressions that contain a color attribute. Consequently, models pre-trained on our dataset primarily acquire generalization capabilities in identifying and distinguishing objects with different colors.

## 7 Conclusion

In this work, we design a new pipeline that aims to generate synthetic data for the Referring Expression Comprehension task. It involves two components: the Annotation Generation Engine for creating new expressive annotations and the Image Generation Engine to generate synthetic images conditioned by the annotations. Our strategy can generate datasets with arbitrary dimensions and complexity without human effort and reduce some errors in labeling. We adopt the method to generate Harlequin, the first dataset collected for the Referring Expression Comprehension task. Harlequin is built on top of Flickr30k Entities’ annotations and is generated varying color attributes in the original referring expressions. We validate our approach by pre-training state-of-the-art models on Harlequin and demonstrate that the acquired generalization capabilities improve the performance after the fine-tuning on real data.

In future work, we plan to investigate the potential and flexibility of our pipeline to progressively get rid of each input until the entire sample is generated from scratch. In particular, we believe that the proposed variation function could be extended to work with other attributes besides the color or could be learned. Some of them, such as *size* and *location*, also require the manipulation of bounding boxes’ coordinates besides queries. There has been effort to face this new challenge. For example, LayoutGPT [11] generates a reasonable arrangement of objects given a textual description and returns their coordinates. This tool, combined with our pipeline, could alleviate the problem of having fixed layout of objects. Finally, we believe that the generation of the referring expressions could be automatized through prompting strategies, which have been proven effective for task adaptation in Large Language Models [2].

**Acknowledgments.** We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support. This work is also supported by the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by NextGenerationEU.

## References

1. Amoroso, R., Morelli, D., Cornia, M., Baraldi, L., Bimbo, A.D., Cucchiara, R.: Parents and children: Distinguishing multimodal deepfakes from natural images. CoRR **abs/2304.00500** (2023)

2. Arora, S., Narayan, A., Chen, M.F., Orr, L.J., Guha, N., Bhatia, K., Chami, I., Ré, C.: Ask me anything: A simple strategy for prompting language models. In: Proc. of the International Conference on Learning Representations (ICLR) (2023)
3. Blender Online Community: Blender - a 3D modelling and rendering package (2024), <http://www.blender.org>
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proc. of the European Conference on Computer Vision (ECCV) (2020)
5. Chen, Z., Wang, P., Ma, L., Wong, K.K., Wu, Q.: Cops-ref: A new dataset and task on compositional referring expression comprehension. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
6. Danelljan, M., Shahbaz Khan, F., Felsberg, M., Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
7. Deng, J., Yang, Z., Chen, T., Zhou, W., Li, H.: Transvg: End-to-end visual grounding with transformers. In: Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
8. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (2019)
9. Dosovitskiy, A., Ros, G., Codevilla, F., López, A.M., Koltun, V.: CARLA: an open urban driving simulator. In: Proc. of the Conference on Robot Learning (CoRL) (2017)
10. Escalante, H.J., Hernández, C.A., González, J.A., López-López, A., Montes-y-Gómez, M., Morales, E.F., Sucar, L.E., Pineda, L.V., Grubinger, M.: The segmented and annotated IAPR TC-12 benchmark. *Comput. Vis. Image Underst.* **114**(4), 419–428 (2010)
11. Feng, W., Zhu, W., Fu, T., Jampani, V., Akula, A.R., He, X., Basu, S., Wang, X.E., Wang, W.Y.: Layoutgpt: Compositional visual planning and generation with large language models. *CoRR* **abs/2305.15393** (2023)
12. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. *CoRR* **abs/1605.06457** (2016)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
14. Jiang, H., Lin, Y., Han, D., Song, S., Huang, G.: Pseudo-q: Generating pseudo language queries for visual grounding. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
15. Jin, J., Ye, J., Lin, X., He, L.: Pseudo-query generation for semi-supervised visual grounding with knowledge distillation. In: Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2023)
16. Juliani, A., Berges, V., Vckay, E., Gao, Y., Henry, H., Mattar, M., Lange, D.: Unity: A general platform for intelligent agents. *CoRR* **abs/1809.02627** (2018)
17. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: MDETR - modulated detection for end-to-end multi-modal understanding. In: Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
18. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.L.: Referitgame: Referring to objects in photographs of natural scenes. In: Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014)




19. Le, H., Mensink, T., Das, P., Karaoglu, S., Gevers, T.: EDEN: multimodal synthetic dataset of enclosed garden scenes. In: Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2021)
20. Li, M., Sigal, L.: Referring transformer: A one-step approach to multi-task visual grounding. In: Proc. of Advances in Neural Information Processing Systems (NeurIPS) (2021)
21. Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., Lee, Y.J.: GLIGEN: open-set grounded text-to-image generation. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
22. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Proc. of the European Conference on Computer Vision (ECCV) (2014)
23. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
24. Lu, J., Rao, J., Chen, K., Guo, X., Zhang, Y., Sun, B., Yang, C., Yang, J.: Evaluation and enhancement of semantic grounding in large vision-language models. In: Proc. of the AAAI Workshop on Responsible Language Models (ReLM) (2024)
25. Lu, M., Li, R., Feng, F., Ma, Z., Wang, X.: LGR-NET: Language Guided Reasoning Network for Referring Expression Comprehension. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)* **in press** (2024)
26. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
27. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV) (2015)
28. Rigoni, D., Parolari, L., Serafini, L., Sperduti, A., Ballan, L.: Weakly-supervised visual-textual grounding with semantic prior refinement. In: Proc. of the British Machine Vision Conference (BMVC) (2023)
29. Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., Schiele, B.: Grounding of textual phrases in images by reconstruction. In: Proc. of the European Conference on Computer Vision (ECCV) (2016)
30. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
31. Roovere, P.D., Moonen, S., Michiels, N., Wyffels, F.: Dataset of industrial metal objects. *CoRR* **abs/2208.04052** (2022)
32. Su, W., Miao, P., Dou, H., Wang, G., Qiao, L., Li, Z., Li, X.: Language adaptive weight generation for multi-task visual grounding. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
33. Tanaka, M., Itamochi, T., Narioka, K., Sato, I., Ushiku, Y., Harada, T.: Generating easy-to-understand referring expressions for target identifications. In: Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
34. Wang, J., Specia, L.: Phrase localization without paired training examples. In: Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

35. Wang, W., Pagnucco, M., Xu, C., Song, Y.: InterREC: An Interpretable Method for Referring Expression Comprehension. *IEEE Transactions on Multimedia (TMM)* **25**, 9330–9342 (2023)
36. Yang, L., Xu, Y., Yuan, C., Liu, W., Li, B., Hu, W.: Improving visual grounding with visual-linguistic verification and iterative reasoning. In: *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
37. Yang, Z., Gan, Z., Wang, J., Hu, X., Ahmed, F., Liu, Z., Lu, Y., Wang, L.: Unitab: Unifying text and box outputs for grounded vision-language modeling. In: *Proc. of the European Conference on Computer Vision (ECCV)* (2022)
38. Yang, Z., Gong, B., Wang, L., Huang, W., Yu, D., Luo, J.: A fast and accurate one-stage approach to visual grounding. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019)
39. Yang, Z., Wang, J., Gan, Z., Li, L., Lin, K., Wu, C., Duan, N., Liu, Z., Liu, C., Zeng, M., Wang, L.: Reco: Region-controlled text-to-image generation. In: *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023)
40. Ye, J., Tian, J., Yan, M., Yang, X., Wang, X., Zhang, J., He, L., Lin, X.: Shifting more attention to visual backbone: Query-modulated refinement networks for end-to-end visual grounding. In: *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
41. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (ACL)* **2**, 67–78 (2014)
42. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: Mattnet: Modular attention network for referring expression comprehension. In: *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
43. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: *Proc. of the European Conference on Computer Vision (ECCV)* (2016)
44. Zhan, Y., Xiong, Z., Yuan, Y.: RSVG: exploring data and models for visual grounding on remote sensing data. *IEEE Trans. Geosci. Remote Sens.* **61**, 1–13 (2023)
45. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2023)





# Size-Modulated Deformable Attention in Spatio-Temporal Video Grounding Pipelines

Hans Tiwari<sup>1</sup>✉ , Selen Pehlivan<sup>2</sup> , and Jorma Laaksonen<sup>1</sup> 

<sup>1</sup> Aalto University School of Science, Espoo, Finland

[hans.tiwari@aalto.fi](mailto:hans.tiwari@aalto.fi), [jorma.laaksonen@aalto.fi](mailto:jorma.laaksonen@aalto.fi)

<sup>2</sup> VTT Technical Research Centre of Finland, Oulu, Finland

[selen.pehliwantort@vtt.fi](mailto:selen.pehliwantort@vtt.fi)

**Abstract.** The integration of attention mechanisms into computer vision tasks, inspired by the success of Transformers in natural language processing, has revolutionized various applications such as object detection and visual grounding. In this paper, we focus on spatio-temporal video grounding (STVG), a computer vision task that aims to jointly extract spatial and temporal regions from videos based on textual descriptions. Leveraging recent advancements in attention-based Transformer architectures, particularly in object detectors, and building upon a recent baseline model, we integrate two enhancements in attention modules: Width-Height Modulation and Deformable Attention units. These enhancements aim to improve the accuracy and efficiency of STVG techniques in two datasets, HC-STVG and VidSTG, by addressing challenges related to feature inconsistencies and prediction reliability across video frames. As a result, our study contributes to advancing the baseline models in spatio-temporal video grounding, bridging the gap between computer vision and natural language processing domains.

**Keywords:** Video Grounding · Spatio-Temporal Video Grounding · Transformers · Attention Unit

## 1 Introduction

The recent success of Transformers in natural language processing has led to the integration of attention mechanisms into computer vision tasks, such as image classification, object detection, and action recognition [8]. Particularly, Transformers have shown competitive performance in object detection with the

---

Supported by the Research Council of Finland in project #345791 *Understanding speech and scene with ears and eyes (USSEE)*. We acknowledge CSC – IT Center for Science, Finland for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through its Extreme Scale Access program.

DETR architecture [1], eliminating the need for hand-crafted components such as anchor generation and non-maximum suppression. More recently, MDETR [10], a multimodal architecture, has further extended this framework to detect objects in images based on free-form text, i.e., phrase grounding.

In video analysis, the grounding problem is mainly explored as the temporal video grounding task, which seeks to identify segment boundaries within videos for described actions. In our study, we focus on the spatio-temporal video grounding tasks (STVG), which aims to jointly extract spatial and temporal regions by employing a series of bounding boxes spanning identified frames, leveraging both spatial and temporal localization losses. This can be considered a crucial multimodal task bridging computer vision and natural language processing, with applications in video indexing, retrieval, and analysis [25].

To tackle the modeling of multimodal representations and spatio-temporal relationships, attention-based architectures [9, 23] are emerging as robust solutions for STVG, leveraging the latest object detection progress [1]. Our objective is to explore integrating recent advances, particularly attention modules, around the baseline Spatio-Temporal Consistency Aware Transformer (STCAT) [9], for the STVG task. Our integration particularly delves into the evaluation and enhancement of attention units within the spatial decoder (see Fig. 1).

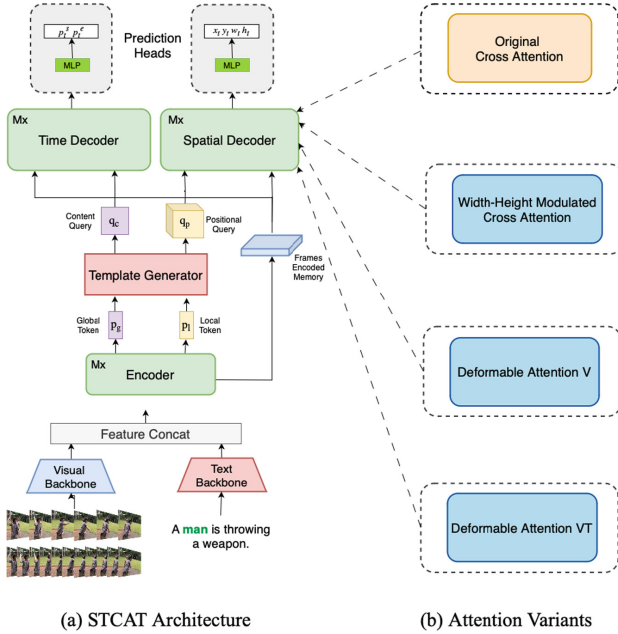
Building upon recent developments in DETR-based object detectors, we initially incorporate the Width-Height Modulation from DAB-DETR [13] and the deformable attention unit from Deformable-DETR [28] into the attention units used in the spatial decoder component of the architecture. Subsequently, these two mechanisms are combined into a single attention unit. Our proposed improvements in the attention unit also include rethinking of the implicit and explicit integration of multimodal representations, visual-text encodings, within the attention block. The experimental results conducted on the HC-STVG [20] and VidSTG [27] benchmark datasets demonstrate that our proposed enhancements yield improved performance. Furthermore, we show in this work that improved attention units can lead to better grounding results for small-scale objects. The source code of our model is available on GitHub<sup>1</sup>.

## 2 Background

Video grounding presents a challenging task that lies at the intersection of computer vision and natural language processing. It involves aligning video regions with corresponding textual descriptions. The complexity of this task is compounded by inherent ambiguity and variability found in natural language descriptions, as well as the multimodal nature of both video and text data. Within the domains of temporal video grounding (TVG) and spatial-temporal video grounding (STVG), research has been conducted to address this challenge.

---

<sup>1</sup> <https://github.com/Hans7331/stvg-work>



**Fig. 1.** The baseline STCAT [9] drawn at a coarse scale and our proposed attention variants integrated into the baseline architecture.

**Temporal Video Grounding (TVG)** Several approaches have been proposed for temporal video grounding, with advancements often revolving around improved feature representations, attention mechanisms, and training strategies [2, 3, 24].

Cross-modal based approaches aim at fostering a rich interaction between the video and text modalities. The Memory Augmented Network (MAN) [25] is an exemplar, utilizing memory networks to capture the cross-modal dynamics. Rank-based approaches generate numerous candidate segments and rank them according to their predicted relevance to the textual description [5]. Recent models have started incorporating more intricate mechanisms to bridge the gap between the vision and text modalities.

Building upon the success of Transformers [22], recent methods have begun utilizing self-attention mechanisms to selectively focus on the most relevant video frames to a given textual description. The development of unified embedding spaces, where video frames and textual tokens coexist, has been another trend. These shared spaces facilitate better understanding and alignment between the two modalities [26]. With the vast diversity in video content and textual descriptions, few-shot learning techniques are being employed to adapt temporal video grounding models to new tasks with limited labeled data [17], incorporating Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based architectures.

**Spatio-Temporal Video Grounding (STVG)** While retaining the temporal grounding aspect, STVG identifies spatial regions in a video segment where the described action or object is present. It outputs a series of bounding boxes, creating a tube-like spatio-temporal structure for a sentence query in an untrimmed video sequence. Early methods like STGRN [27] and STGVT [20] rely on a two-stage process. First, a pretrained tube detector generates candidate proposals. Then, the best tube proposal is chosen from these candidates, ranked according to their similarity to the sentence query. Recently, STVG models have emerged as one-stage approaches. These include STVGBert [19], which introduces an improved visual-linguistic transformer that does not rely on any pretrained detectors. However, the method faces a feature alignment issue due to the absence of complete video content. Another one-stage model is TubeDETR [23], based on a Transformer model inspired by the DETR [1] and MDETR [10] architectures. The Transformer includes a video and text encoder for spatial multimodal interactions and a space-time decoder to perform spatio-temporal localization. While STVGBert is primarily built around its core module and TubeDETR employs a video-text encoder followed by a space-time decoder, CSDVL [12] proposes a framework where a static vision-language stream and a dynamic vision-language stream collaboratively reason for localization.

STCAT [9], used as a baseline in our experiments, is more intricate. It incorporates a cross-modal encoder, a template generator, and a query-based decoder in a one-stage approach. The architecture of STCAT emphasizes the importance of feature alignment consistency in the STVG task.

More recently, CoSTA [11] employs a space-time entanglement framework to address space-time interaction. Another concurrent work, CG-STVG [7], relies on an encoder-decoder architecture with a context-guided decoder that integrates mined context from video at each decoding stage. The integration of large language models is becoming increasingly significant in grounding tasks. One recent study, PG-Video-LLaVA [16], introduces a grounding module designed to localize objects in videos based on user instructions. Although the experiments were conducted on videos from the VidSTG and HC-STVG datasets, the assessments specifically focused on the spatial grounding task.

### 3 Our Model

This section summarizes the basics of the chosen baseline architecture and introduces two proposed attention unit variants. The proposed variations consists of an enhanced attention mechanism that incorporates Width-Height Modulation and a Deformable Attention unit, which replaces the standard attention block. Width-Height Modulation enhances the spatial attention maps by integrating scale information directly into them, allowing for robust feature extraction from objects with varying widths and heights. On the other hand, Deformable Attention unit addresses the challenges of applying Transformer attention on image feature maps, focusing only on selected key sampling points around a reference point.

### 3.1 Baseline STVG Model

Numerous methods have been proposed to solve the spatio-temporal video grounding task by treating it as a parallel frame-grounding problem. This approach, however, has led to several challenges, particularly related to feature and prediction inconsistencies. These inconsistencies, especially when occurring together, can hinder the accuracy and reliability of the grounding process, making it imperative to seek alternative strategies.

To address the challenges, the Spatio-Temporal Consistency Aware Transformer (STCAT) [9] introduces an innovative end-to-end one-stage framework. Its essence lies in its ability to ensure consistent grounding across video frames. It achieves this by employing a novel multimodal template as a global objective. This template serves to constrict the grounding region, ensuring that predictions are consistently associated across different video frames. A visual representation of this architecture, depicted at a coarse scale, is shown in Fig. 1.

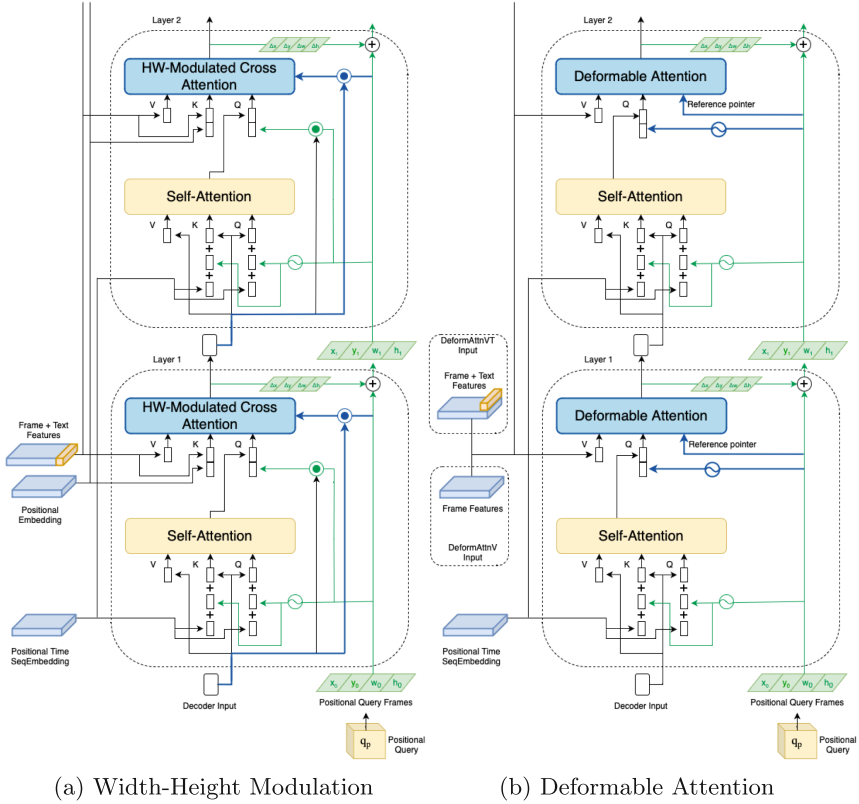
The architecture employs both visual and textual backbones to meticulously extract visual and textual features. To extract visual features, the model utilizes the pretrained weights of ResNet-101 [10]. On the other hand, for textual features, pretrained RoBERTa [14] is leveraged. These extracted features are then concatenated and channeled into the encoder block. Within this block, global and local tokens are processed, which carry overarching video-level contexts and specific frame-level contexts, respectively. These tokens are subsequently directed to the template generator block. Here, the content query is utilized as input for the time decoder, while the positional query serves as input for the spatial decoder. Notably, these two decoders share similarities and their intricacies will be elaborated upon in subsequent sections. Finally, the prediction heads come into play, determining the start and end times through the time decoder’s prediction head. Concurrently, the spatial decoder’s prediction head ascertains the coordinates for the bounding boxes associated with each frame, resulting in comprehensive prediction output.

In the context of STVG, the attention unit’s functionality and its variants are of significant interest. As the STCAT model’s code was available when we needed a baseline for STVG tasks, we based our video grounding model on it and proposed various modifications in its spatial decoder’s attention unit, as will be described next.

### 3.2 Width-Height Modulation

The concept of Width-Height Modulation emerges as an enhancement in the STCAT’s decoder’s attention unit as shown in Fig.2a. Traditional positional attention maps, often visualized as Gaussian-like priors, have been conventionally assumed to be isotropic with a fixed size for all objects [13]. This assumption inadvertently neglects the scale information, specifically the width and height of objects.

To address this limitation and enhance the positional prior, a novel approach, originally proposed in DAB-DETR [13], is being leveraged here: the integration of



**Fig. 2.** Two proposed attention unit variants integrated into the spatial decoder of the baseline model.

scale information directly into the attention maps. In the conventional positional attention map, the query-to-key similarity is computed as:

$$\text{Attn}((x, y), (x_{ref}, y_{ref})) = \frac{\text{PE}(x) \cdot \text{PE}(x_{ref}) + \text{PE}(y) \cdot \text{PE}(y_{ref})}{\sqrt{D}}, \quad (1)$$

where  $\text{PE}(\cdot)$  stands for the sinusoidal position encoding, the  $\frac{1}{\sqrt{D}}$  factor serves as a rescaling term, as suggested by [22].  $(x, y)$  are the coordinates of the current position being attended to, while  $(x_{ref}, y_{ref})$  are those of a reference position used to compute the attention score. To better accommodate objects of varying scales, the positional attention maps can be modulated by dividing the reference anchor width and height by those of the query. This modulation can be represented as:

$$\text{Modulate}_x = \text{PE}(x) \cdot \text{PE}(x_{ref}) \cdot \frac{w_{q,ref}}{w_q}, \quad (2)$$

$$\text{Modulate}_y = \text{PE}(y) \cdot \text{PE}(y_{ref}) \cdot \frac{h_{q,ref}}{h_q}, \quad (3)$$

$$\text{ModulateAttn}((x, y), (x_{ref}, y_{ref})) = \frac{\text{Modulate}_x + \text{Modulate}_y}{\sqrt{D}}, \quad (4)$$

where  $w_q$  and  $h_q$  denote the width and height of the anchor  $A_q$ , and  $w_{q,ref}$  and  $h_{q,ref}$  represent the reference width and height. These two are computed from the positional query  $C_q$  as:

$$w_{q,ref}, h_{q,ref} = \sigma(\text{MLP}(C_q)). \quad (5)$$

This modulation in the positional attention facilitates robust extraction of features from objects with diverse spatial sizes. The video and text inputs here are processed in a manner similar to the baseline STCAT, with modifications made exclusively in the spatial decoder component.

### 3.3 Deformable Attention Unit

The Deformable Attention Transformer [28] introduces a novel approach to address the challenges of applying Transformer attention on image feature maps. Unlike traditional Transformer attention mechanisms, which consider all possible spatial locations, this innovative method aims to optimize computational efficiency and enhance performance. The deformable attention module focuses only on a select set of key sampling points around a reference point. This approach, regardless of the spatial size of the feature maps, can mitigate issues related to convergence and spatial resolution of features.

Given an input feature map  $x \in \mathbb{R}^{C \times H \times W}$ , let  $q$  index a query element with content feature  $z_q$  and a 2-D reference point  $p_q$ , the deformable attention feature is formulated as:

$$\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^M W_m \sum_{k=1}^K A_{mqk} \cdot W_m^0 x(p_q + \Delta p_{mqk}), \quad (6)$$

where  $m$  and  $k$  index the attention head and the sampled keys, respectively. The attention weight  $A_{mqk}$  lies in the range  $[0,1]$ , normalized by  $\sum_{k=1}^K A_{mqk} = 1$ , and the sampling offset  $\Delta p_{mqk}$  is obtained through linear projection over the query feature  $z_q$ .

The Template Generator, as introduced in the STCAT framework by [9], produces two key outputs: the positional query  $q_p$  and the content query  $q_c$ . In the case of deformable attention, the decoder input is the content query  $q_c$ . Conversely, in the original STCAT framework, this decoder input is a zero-initialized embedding, as depicted in Fig. 2b. The reference anchor, which remains consistent with the original STCAT framework, is the positional query  $q_p$ . This reference anchor plays a crucial role in guiding and constraining the grounding region for each frame in the video.

In this work, we introduce two variations of deformable attention blocks: DeformAttnV and DeformAttnVT. The DeformAttnV variant is inspired by the original design presented in Deformable DETR [28]. Since the deformable attention module is designed to handle visual data exclusively, the input primarily comprises of visual, with no textual element. However, the textual context is inherently encoded via the template generator. This indirect inclusion of textual context in the spatial pipeline ensures that the essence of the original STCAT architecture is preserved.

Conversely, in the DeformAttnVT variant, the architecture explores the potential of leveraging multi-scaling within deformable attention. Recognizing the challenge of memory constraints and the necessity of maintaining performance, a pragmatic approach is taken. Here, the textual feature is treated as a scale of the visual feature, with a width equivalent to the length of the textual feature and a height of one. This approach aims to incorporate textual context into the spatial decoder pipeline.

We observed improvements with DeformAttnV, which may be attributed to the robustness of the Template Generator and/or the effectiveness of the spatial decoder, both of which contribute to enhancing the performance metrics. In the case of DeformAttnVT, while this approach offers a creative solution to integrate textual context, its efficacy may be limited due to the inherent complexities of distinguishing between the two types of features within the deformable attention framework.

To further enhance performance, we integrated Width-Height Modulation (W&H) with Deformable Attention, creating two variants: DeformAttnV+W&H and DeformAttnVT+W&H. In both, we combined the robust spatial and textual decoding of Deformable Attention with the scale-awareness of W&H. This leverages the selective key sampling of Deformable Attention while ensuring positional priors are modulated for object scales. W&H adjusts the cross-attention input by scaling the positional query frame embedding according to the ratios of reference height and width to the object’s dimensions. Similarly, this modulation is integrated into the deformable attention unit by adjusting the width and height dimensions of the positional embeddings, thereby enhancing the model’s spatial awareness and focus on relevant features. Furthermore, notable improvements were seen in the case of DeformAttnVT+W&H, highlighting the effectiveness of combining these two approaches to enhance performance metrics.

### 3.4 Training Objectives

In our approach, we utilize a set of pivotal loss functions, similar to STCAT [9], to optimize the performance of our model. The computation of  $L_{bbox}$  involves several components. Firstly, the L1 Loss ( $L_{L1}$ ) is utilized, which measures absolute differences between true and predicted values. It is primarily employed for spatial localization. Furthermore, the gIoU Loss ( $L_{giou}$ ) plays a critical role, especially beneficial for object detection tasks. Both  $L_{L1}$  and  $L_{giou}$  are utilized in the computation of  $L_{bbox}$  for spatial localization. The Temporal Loss component ( $L_{temp}$ ) is calculated based on the KL Divergence Loss ( $L_s$  and  $L_e$ ), which



are employed to measure the divergence between target and predicted probability distributions for starting and ending positions in temporal localization tasks. For binary classification purposes, we employ the Binary Cross Entropy Loss ( $L_{seg}$ ) to measure the error in predicting frame membership to the ground-truth segment. Furthermore, we incorporate the Guided Attention Loss ( $L_{att}$ ), which penalizes the model for focusing on irrelevant information by computing the negative logarithm of attention weights. This loss ensures that the model prioritizes relevant parts of the input, such as the action segment.

The losses are combined to compute the composite loss  $L$  defined as:

$$L = \lambda_{bbox}L_{bbox} + \lambda_{temp}L_{temp} + \lambda_{seg}L_{seg} + \lambda_{att}L_{att}, \quad (7)$$

where  $\lambda_{bbox}$ ,  $\lambda_{temp}$ ,  $\lambda_{seg}$ , and  $\lambda_{att}$  are coefficients that control the contribution of the corresponding loss components  $L_{bbox}$ ,  $L_{temp}$ ,  $L_{seg}$ , and  $L_{att}$  to the composite loss  $L$ . The mentioned losses are also used in the TubeDETR [23] model.

## 4 Experiments

In this section, we discuss the datasets employed and outline their characteristics and significance in our research. Subsequently, we introduce the evaluation metrics chosen to assess the model’s performance rigorously. Following this, we delve into the implementation details, providing information on hardware setups, optimization strategies, and training procedures. Finally, we present both quantitative and qualitative results obtained from our experiments and compare them with baseline STCAT performances, providing insights into the effectiveness and robustness of our proposed approach.

### 4.1 Datasets

**HC-STVG Dataset** Tang et al. [20] provide the HC-STVG dataset focusing solely on humans with 16,500 description-video pairs from various movie scenes. The training split of the dataset contains 10131, the validation split 2000, and the test split 4413 videos, respectively. The dataset ensures that test and training samples are not derived from the same raw video. Each video clip is accompanied by a descriptive statement and trajectories of the corresponding person, represented as a series of bounding boxes. Notably, all clips include multiple individuals, enhancing the challenge of video comprehension. Throughout our study, the HC-STVG dataset has been primarily used for testing and comparisons due to its comprehensive public availability and lightweight nature.

**VidSTG Dataset** Zhang et al. [27] introduce VidSTG, a large-scale STVG dataset by augmenting the sentence annotations on VidOR [18, 21]. VidOR is recognized as the most extensive object relation dataset, comprising 10,000 videos with detailed annotations for objects and their interrelations. Specifically, it categorizes 80 object types with dense bounding boxes and 50 relation predicate

categories among objects, including 8 spatial and 42 action relations. Each relation in VidOR is represented as a triplet  $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ , associated with temporal boundaries and spatio-temporal tubes of the human subject and object. Please note that the test set of the VidSTG dataset is not publicly available. Consequently, our research solely focuses on validation results, which are then compared with the reported validation results of TubeDETR [23].

## 4.2 Evaluation Metrics

A set of evaluation metrics commonly used in the previous STVG studies are adopted to rigorously assess the performance [4, 6, 9, 23, 27].

$m\_tIoU$ , mean temporal intersection over union, evaluates the temporal localization performance by the average temporal Intersection-over-Union ( $tIoU$ ). The  $tIoU$  is defined as the ratio of the intersection to the union of the predicted and ground truth clips,  $tIoU = \frac{|Ti|}{|Tu|}$ , where  $Ti$  and  $Tu$  are the intersection and union between the temporal locations of the predicted tube and ground-truth tube, respectively.

$m\_vIoU$ , mean visual intersection over union, provides an average of the  $vIoU$  scores across all testing videos, where  $vIoU = \frac{1}{|Su|} \sum_{t \in Si} IoU(\hat{bt}, bt)$  and  $\hat{bt}$  and  $bt$  are the detected and ground-truth bounding boxes at frame  $t$ , respectively.  $IoU$  is the Intersection-over-Union between these bounding boxes.  $Si$  and  $Su$  represent the union between the predicted and ground-truth tubes.

$vIoU@0.3$  and  $vIoU@0.5$  are metrics measuring the proportion of samples for which the  $vIoU$  score exceeds a certain threshold (0.3 and 0.5, respectively). Specifically,  $vIoU@R$  represents the ratio of samples with  $vIoU > R$  in the testing subset.

$m\_gt\_vIoU$ , mean ground truth  $vIoU$ , is computed analogously to  $vIoU$ , with a distinct difference in its focus. While the standard  $vIoU$  calculates the  $IoU$  between predicted boxes and ground truth boxes of the frames in the predicted temporal segment of the action,  $m\_gt\_vIoU$  evaluates the  $IoU$  between the same, but in the ground-truth temporal segment of the action for a more nuanced evaluation of the model’s spatial accuracy. The extended metrics,  $m\_gt\_vIoU@0.3$  and  $m\_gt\_vIoU@0.5$ , are derived similarly, setting  $IoU$  thresholds at 0.3 and 0.5, respectively. The primary significance of employing this metric is to segregate the spatial performance evaluations from other metrics, ensuring a more isolated and focused assessment of the bounding box predictions.

## 4.3 Implementation Details

The experiments were conducted utilizing a setup of 32 AMD GPUs. The optimizer used is AdamW [15]. The learning rate is dynamically adjusted during training based on the schedule. The base learning rate is  $10^{-4}$ , the text learning rate is  $5 \cdot 10^{-5}$ , the visual backbone learning rate is  $2 \cdot 10^{-5}$ , and the temporal learning rate is  $10^{-4}$ . Initially, a warm-up phase is used where the learning rate gradually increases. After the warm-up, the learning rate is decreased at specified

epochs (drop steps). The loss weight hyper-parameter values (see Section 3.4) used for experiments were  $\lambda_{bbox} = 5$ ,  $\lambda_{temp} = 2$ ,  $\lambda_{seg} = 2$ , and  $\lambda_{att} = 1$ .

For the HC-STVG dataset, the model input is set to a resolution of  $448 \times 448$ . During training, there is a 50% probability that the input will be flipped controlled with a variable. The sampling rate is set at 3.2 frames per second to ensure a uniform number of frames from each 20-second video, simplifying processing and analysis compared to using fps, which varies with video length. The training process spans 90 epochs with a batch size of one video. The training data is shuffled before each epoch. The learning rate decreases at the 50th and 90th epochs. Pre-validation is not used in this setup, instead directly the test set is used, because here the split does not contain any validation dataset.

For the VidSTG dataset, the model resolution and probability thresholds remain the same as those of HC-STVG. Additionally, there is a 50% probability for temporal cropping also controlled with a variable. The number of training samples is set to 64. The training process is set to run for a maximum of 7 epochs, with a batch size of one video.

**Table 1.** Performance comparison on the HC-STVG test set among models featuring different attention unit variants. The performance values are reported at epoch 90. STCAT\* shows our replicated results for comparison.

Methods	$m\_tIoU$	$m\_vIoU$	$vIoU@0.3$	$vIoU@0.5$	$m\_gt\_vIoU$	$gt\_vIoU@0.3$	$gt\_vIoU@0.5$	Params/M	FLOPs/T
STVGBert [19]	20.42	29.37	11.31	—	—	—	—	—	—
TubeDETR [23]	43.70	32.40	49.80	23.50	—	—	—	—	—
STCAT [9]	49.44	35.09	57.67	30.09	—	—	—	—	—
CG-STVG [7]	52.80	38.40	61.50	36.30	—	—	—	—	—
CoSTA [11]	52.85	38.97	63.10	38.19	—	—	—	—	—
STCAT*	47.74	34.16	56.21	29.22	68.43	90.17	81.03	159.7	2.10
W&H Modulation	48.86	<u>34.95</u>	<u>56.81</u>	28.88	68.80	90.78	81.12	159.8	2.98
DeformAttnV	<u>49.06</u>	34.88	<b>57.93</b>	<u>29.91</u>	<b>69.62</b>	<b>91.55</b>	<b>83.19</b>	165.0	1.40
DeformAttnVT	47.38	33.88	54.83	28.71	69.24	90.95	81.55	165.0	1.79
DeformAttnV+W&H	47.30	33.70	54.83	28.10	69.27	90.95	<u>82.67</u>	165.0	1.40
DeformAttnVT+W&H	<b>49.26</b>	<b>35.09</b>	<u>56.81</u>	<b>32.41</b>	<u>69.52</u>	<b>91.55</b>	<u>82.67</u>	165.0	1.79

#### 4.4 Results and Comparison to the Baseline

Table 1 reports the outcomes of recent studies, including our replication of [9], followed by recent works, TubeDETR [23], CG-STVG [7] and CoSTA [11], and our proposed improvements on the HC-STVG dataset. It is important to note that CG-STVG and CoSTA are very recent advancements in the field, and at the time of this study, their codes were not accessible. Consequently, STCAT has been employed as the baseline for comparative analysis throughout our study. Initially, our experiments were conducted with the default parameters at epoch 90 following [9]. However, when the experiment conducted in STCAT was replicated, the results obtained in STCAT\* did not match the original values. Compared to STCAT\*, significant improvement is observed with both the DeformAttnV and W&H Modulation models. We can see that the best results were

obtained with the model where the original spatial decoder had been replaced with the deformable attention unit, as seen in the DeformAttnV result. Subsequently, by integrating deformable attention and W&H modulation, we observed improved performance particularly with DeformAttnVT+W&H model across almost all metrics, with a significant increase of roughly three points in the metric  $vIoU@0.5$  from 29.22 in STCAT\* to 32.41 in DeformAttnVT+W&H. DeformAttnVT+W&H outperforms all other methods including the DeformAttnV+W&H in  $vIoU@0.5$ . We also report in Table 1 the number of parameters and FLOPs of our improved models. While STCAT\* and our models have similar number of parameters, our DeformableAttnVT and DeformableAttnVT+W&H have fewer FLOPs compared to STCAT\*. Please note that while deformable attention is primarily designed to support multi-scale processing, memory constraints necessitated the use of a single scale.

**Table 2.** Performance comparison on the VidSTG validation set among models featuring different attention unit variants. The performance values are reported at epoch 7. STCAT\* shows our replicated results.

Methods	$m\_tIoU$	$m\_vIoU$	$vIoU@0.3$	$vIoU@0.5$	$m\_gt\_vIoU$	$gt\_vIoU@0.3$	$gt\_vIoU@0.5$
TubeDETR [23]	46.90	26.20	36.10	24.10	—	—	—
STCAT*	49.72	<u>28.57</u>	<b>39.59</b>	<u>27.22</u>	<u>52.88</u>	<u>70.29</u>	<b>59.94</b>
W&H Modulation	<b>49.95</b>	<b>28.70</b>	<u>39.53</u>	<b>27.40</b>	<b>53.12</b>	<b>71.19</b>	<u>59.86</u>
DeformAttnV	49.58	27.03	37.82	25.02	50.29	69.25	56.95
DeformAttnVT	<b>49.95</b>	27.42	38.69	25.73	50.68	69.34	57.10
DeformAttnV+W&H	49.78	27.06	37.68	25.27	50.39	69.23	57.03
DeformAttnVT+W&H	49.54	27.08	37.54	25.02	50.71	69.12	57.18

Table 2 presents the evaluation results obtained on the VidSTG dataset. Our performance evaluation utilizes the same split as the TubeDETR architecture [23], allowing for direct comparison with their study. While all proposed variations outperform TubeDETR [23], W&H Modulation gives better results than STCAT\* model. Notably, the value of  $gt\_vIoU@0.3$  increases by almost a point from 70.29 in STCAT\* to 71.19 with W&H Modulation.

#### 4.5 Analysis on Object Scales

Evaluation criteria are pivotal in understanding both the temporal and spatial accuracy of the model in the context of STVG task. As crucial as the above metrics are, to gain a deeper understanding of the model’s performance across different scales of bounding boxes within the dataset, we introduce several variations of these metrics. These metrics extend the current spatial evaluation metrics, such as  $m\_vIoU$  and its thresholded versions, to specific subsets of the test set. Given the variability in width and height for each sample in the dataset,

these subsets are delineated based on the threshold of the metric as follows:

$$\text{BBoxRatio} = \frac{1}{N} \sum_{t=1}^N \frac{\text{GTBBoxArea}_t}{\text{FrameTotalArea}_t}, \quad (8)$$

where  $t$  indexes the ground truth frames in the video from 1 to  $N$  covering the action occurrence and GTBBox represents the ground truth bounding boxes in the corresponding video.

Subsequently, the videos are categorized into *Small*, *Medium*, and *Large* subsets, each containing a varying number of samples based on the thresholded BBoxRatio in eq. (8). The resulting distribution of HC-STVG dataset according to BBoxRatio is as follows: 709 *Small* boxes (BBox Ratio  $\leq 0.25$ ), 401 *Medium* boxes, and 50 *Large* boxes ( $0.5 < \text{BBox Ratio}$ ). As can be seen in the results of Table 3, the DeformAttnV model consistently outperforms STCAT\* across all scales, while the DeformAttnVT+W&H model results in the best performance specifically for the small-scale boxes.

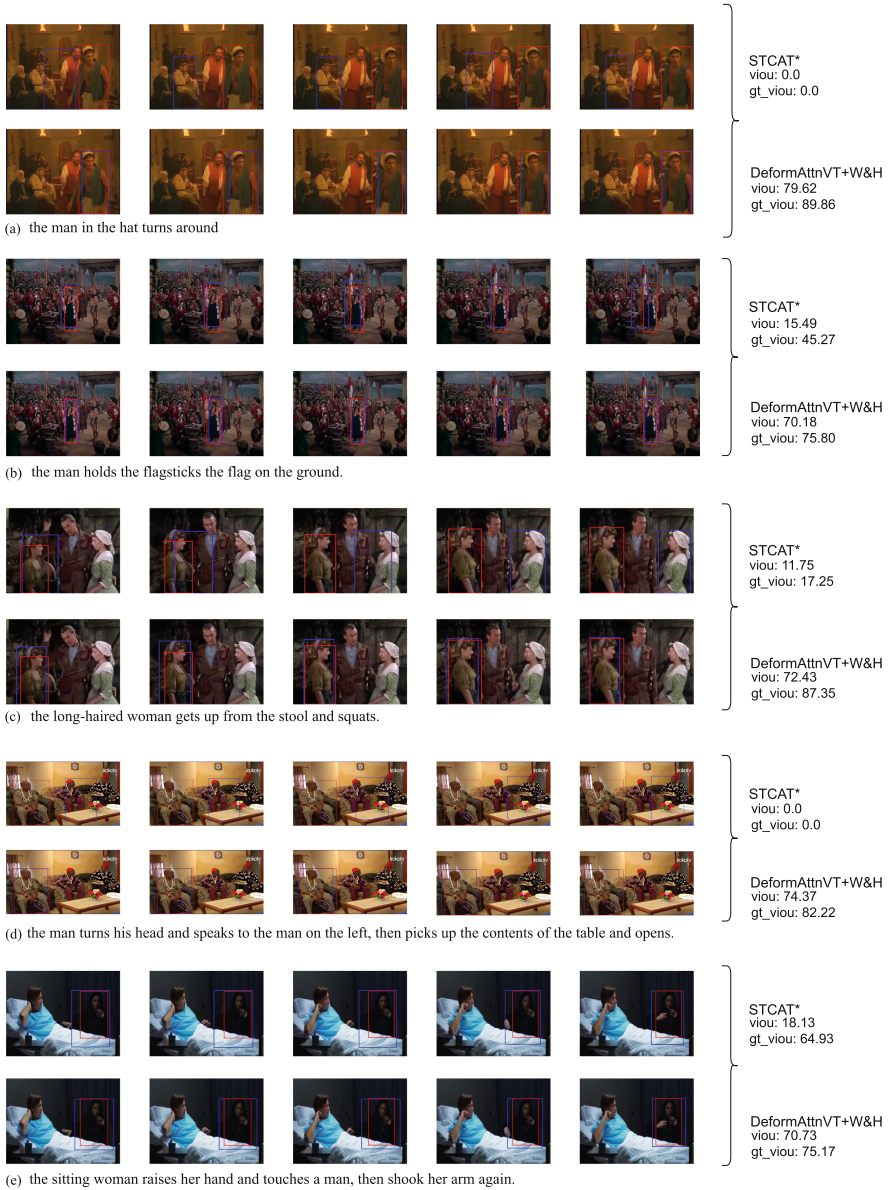
**Table 3.** Performance comparison on the HC-STVG test set across various bounding box scales. STCAT\* shows our replicated results.

Methods	Small		Medium		Large	
	$vIoU@0.3$	$vIoU@0.5$	$vIoU@0.3$	$vIoU@0.5$	$vIoU@0.3$	$vIoU@0.5$
STCAT*	53.17	26.94	61.10	32.17	<u>60.00</u>	<u>38.00</u>
W&H Modulation	<u>53.74</u>	26.09	<b>62.84</b>	33.42	52.00	32.00
DeformAttnV	<u>53.74</u>	<u>28.07</u>	<u>61.85</u>	<u>36.91</u>	<b>62.00</b>	<b>44.00</b>
DeformAttnVT	53.17	24.82	59.35	34.66	42.00	36.00
DeformAttnV+W&H	52.61	23.98	58.60	34.91	56.00	32.00
DeformAttnVT+W&H	<b>54.02</b>	<b>28.21</b>	61.60	<b>39.15</b>	58.00	<u>38.00</u>

## 4.6 Qualitative Analysis

We conducted qualitative analysis on a subset of samples extracted from the HC-STVG dataset. In Fig. 3, parallel results with STCAT\* and DeformAttnVT+W&H are shown together with ground truth bounding boxes. The examples provided in (a)–(c) pertain to instances involving *Small*-sized objects, while (d)–(e) relate to *Medium*-sized objects (see Section 4.5).

Examining example (a), the phrase *the man in the hat turns around* describes a scene featuring a man wearing a hat. Particularly, there are two men with hats present in the video. Our model adeptly identifies and tracks the correct individual wearing the hat, as indicated by the ground truth box highlighted in red. In contrast, STCAT\* appears to misidentify a different individual with a hat. This discrepancy is similarly evident in examples (c) and (d), wherein



**Fig. 3.** Qualitative Analysis comparison of STCAT\* and DeformAttnVT+W&H on the HC-STVG dataset. Examples (a)–(c) relate to small and (d)–(e) to medium size boxes. Red frames show the ground truth and blue ones the detected objects. The increased values of metric  $m_{vIoU}$  and  $m_{gt_vIoU}$  evince the quantitative improvements obtained in Table 3.

our model consistently outperforms STCAT\* in tracking the correct individual across frames. In example (b), the trajectory of STCAT\* bounding boxes exhibits significant drifting, whereas our model demonstrates a steady performance in intersection over union. Lastly, in example (e), both models effectively ground the detailed text, with our model demonstrating superior performance in bounding box size estimation, closely aligning with the ground truth.

## 5 Conclusions

In this paper we presented improvements to spatio-temporal video grounding, in particular to the STCAT model that served as our baseline, and evaluated them on the HC-STVG and VidSTG datasets. First, a comparative analysis between the original attention unit and our proposed version consisting of Width-Height Modulation and two variants of Deformable Attention revealed subtle differences in respect to seven performance measures.

With the HC-STVG dataset, the combination of Width-Height Modulation and Deformable Attention with both vision and text was best-performing. With the VidSTG dataset, the Width-Height Modulation alone performed the best. In another experiment, we showed improvement for especially small and medium sized objects. This is very important particularly for small objects that are generally the most challenging ones to ground correctly. Our qualitative validations further verified that the quantitative improvements obtained with our proposed DeformAttnVT+W&H model were also visible in more accurate and stable spatial location of the objects.

Looking ahead, our proposed variants for the spatial attention could be applied to other state-of-the-art baseline STVG models and extended to the temporal domain for improved spatio-temporal grounding. Yet another direction of extension could be to apply grounding also in the audio domain.

## References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229. Springer (2020)
2. Chen, J., Chen, X., Ma, L., Jie, Z., Chua, T.S.: Temporally grounding natural sentence in video. In: ENLP. pp. 162–171 (2018)
3. Chen, J., Ma, L., Chen, X., Jie, Z., Luo, J.: Localizing natural language in videos. In: AAAI. vol. 33, pp. 8175–8182 (2019)
4. Chen, Z., Ma, L., Luo, W., Wong, K.Y.K.: Weakly-supervised spatio-temporally grounding natural sentence in video. arXiv preprint [arXiv:1906.02549](https://arxiv.org/abs/1906.02549) (2019)
5. Gao, J., et al.: Fast, accurate, and lightweight temporal action proposal generation. In: ECCV (2020)
6. Ging, S., Zolfaghari, M., Pirsiavash, H., Brox, T.: COOT: Cooperative hierarchical transformer for video-text representation learning. NIPS **33**, 22605–22618 (2020)
7. Gu, X., Fan, H., Huang, Y., Luo, T., Zhang, L.: Context-guided spatio-temporal video grounding. In: CVPR. pp. 18330–18339 (2024)

8. Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., Tao, D.: A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(1), 87–110 (2023). <https://doi.org/10.1109/TPAMI.2022.3152247>
9. Jin, Y., Yuan, Z., Mu, Y., et al.: Embracing consistency: A one-stage approach for spatio-temporal video grounding. *NIPS* **35**, 29192–29204 (2022)
10. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: MDETR – modulated detection for end-to-end multi-modal understanding. In: *ICCV*. pp. 1780–1790 (2021)
11. Liang, Y., Liang, X., Tang, Y., Yang, Z., Li, Z., Wang, J., Ding, W., Huang, S.L.: Costa: End-to-end comprehensive space-time entanglement for spatio-temporal video grounding. In: *AAAI*. vol. 38, pp. 3324–3332 (2024)
12. Lin, Z., Tan, C., Hu, J.F., Jin, Z., Ye, T., Zheng, W.S.: Collaborative static and dynamic vision-language streams for spatio-temporal video grounding. In: *CVPR*. pp. 23100–23109 (2023)
13. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: DAB-DETR: Dynamic anchor boxes are better queries for DETR. *arXiv preprint arXiv:2201.12329* (2022)
14. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692* (2019)
15. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: *ICLR* (2019)
16. Munasinghe, S., Thushara, R., Maaz, M., Rasheed, H.A., Khan, S., Shah, M., Khan, F.: Pg-video-llava: Pixel grounding large video-language models. *arXiv preprint arXiv:2311.13435* (2023)
17. Nag, S., Zhu, X., Xiang, T.: Few-shot temporal action localization with query adaptive transformer. *arXiv preprint arXiv:2110.10552* (2021)
18. Shang, X., Di, D., Xiao, J., Cao, Y., Yang, X., Chua, T.S.: Annotating objects and relations in user-generated videos. In: *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. pp. 279–287. ACM (2019)
19. Su, R., Yu, Q., Xu, D.: STVGBert: A visual-linguistic transformer based framework for spatio-temporal video grounding. In: *ICCV*. pp. 1513–1522 (2021). 10.1109/ICCV48922.2021.00156
20. Tang, Z., Liao, Y., Liu, S., Li, G., Jin, X., Jiang, H., Yu, Q., Xu, D.: Human-centric spatio-temporal video grounding with visual transformers. *IEEE Trans. Circuits Syst. Video Technol.* **32**(12), 8238–8249 (2021)
21. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. *Commun. ACM* **59**(2), 64–73 (2016)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *NIPS* (2017)
23. Yang, A., Miech, A., Sivic, J., Laptev, I., Schmid, C.: TubeDETR: Spatio-temporal video grounding with transformers. In: *CVPR*. pp. 16442–16453 (2022)
24. Zhang, S., Peng, H., Fu, J., Luo, J.: Learning 2D temporal adjacent networks for moment localization with natural language. In: *AAAI*. vol. 34, pp. 12870–12877 (2020)
25. Zhang, Z., et al.: MAN: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In: *CVPR*. pp. 1247–1255 (2019)
26. Zhang, Z., et al.: Cross-modal interaction networks for query-based moment retrieval in videos. In: *ACM SIGIR*. pp. 655–664 (2020)



27. Zhang, Z., Zhao, Z., Zhao, Y., Wang, Q., Liu, H., Gao, L.: Where does it exist: Spatio-temporal video grounding for multi-form sentences. In: CVPR (2020)
28. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for End-to-End object detection. arXiv preprint [arXiv:2010.04159](https://arxiv.org/abs/2010.04159) (2020)



# Dual Branch Non-Autoregressive Image Captioning

Yuanqiu Liu, Hong Yu, Hui Li, Xin Han, and Han Liu<sup>(✉)</sup>

Dalian University of Technology, Dalian 116024, China  
liu.han.dut@gmail.com

**Abstract.** Image captioning is a typical task in multimodal learning. Many existing image captioning models rely on autoregressive paradigms, causing notable delays in inference and impacting practical applications. While non-autoregressive methods effectively address the issue of inference delay, there still exists a performance gap when compared to autoregressive models. In this paper, we introduce a dual branch non-autoregressive image captioning model that significantly enhances performance. Firstly, we leverage both region and grid features to fully exploit the fine-grained aspects of the image. To prevent an increase in inference delay, we designed a dual branch network to handle these two features separately. Secondly, we design a word retrieval module to augment the semantic richness of the inputs to the non-autoregressive decoder. Meanwhile, our approach incorporates multiple teacher models in the knowledge distillation process, which aims to preserve the diversity of our model by avoiding reliance on a single autoregressive teacher model. Experiments on the MSCOCO dataset show that our dual branch non-autoregressive image captioning model achieves new state-of-the-art performances, boosting a 128.8% CIDEr score on the ‘Karpathy’ offline test split and delivering a 17× inference speedup.

**Keywords:** Image captioning · Non-autoregressive · Dual branch.

## 1 Introduction

Image captioning [30] is a challenging multimodal task aimed at generating a caption that reflects the image content and conforms to human language habits for an image. Image captioning models usually adopt the encoder-decoder architecture [1], where the encoder is responsible for image processing and the decoder generates caption statements. Early image captioning models [10, 13] used the pre-trained CNN and object detection models to extract image features as input and used Recurrent Neural Networks (RNNs) or Transformer [26] to generate words in the captions one by one.

By using the Transformer-based decoder, the autoregressive image captioning models [6, 28] not only improve the model performance but also greatly speed up the training speed because it can be trained in parallel. During the model inference process, they generate each word conditioned on the sequence of previously

generated words [25] and result in high inference latency which is unacceptable in some real-time applications, such as blind assistants. Non-autoregressive models [5, 8] can significantly improve the inference speed of the model, as they can generate words in parallel during both training and inference processes. However, there is still a performance gap between the non-autoregressive image captioning model and the autoregressive model.

Research on non-autoregressive image captioning [5, 8] usually focuses on decoders, using iterative refinement and sequence-level knowledge distillation methods to improve model generation quality. In the encoder stage, they mainly use region features extracted from the object detection model such as Faster R-CNN [9] as the basis for generating captions. However, due to the inability of covering areas outside the target, region features do not contain contextual information [23]. Therefore, in the terms of accuracy, region features can help the model perceive the objects in the image more effectively. However, as for the completeness and fluency of sentences, grid features play a more important role since they are extracted from the entire image and contain more contextual information. More and more autoregressive models integrate multiple image features [29] to achieve better performance, while non-autoregressive models overlook this issue. As shown in Figure 1, another typical issue is missing input for the non-autoregressive decoder, and most existing models use [MASK] sequences as inputs [8], which contain no semantic information and have the same initial values for each position. It also makes it difficult to predict the correct words at the corresponding positions and exacerbates the decoding inconsistency problem.



**Fig. 1.** The autoregressive image captioning model generates words one by one, while the non-autoregressive model generates words in parallel, which can greatly improve the inference speed of the model.

On the other hand, existing non-autoregressive image captioning models use the sequence-level knowledge distillation [15] to solve the problem of decoding inconsistency. They usually employ an autoregressive image captioning model as a teacher model to guide the training of non-autoregressive models [11], thereby helping them re-establish semantic dependencies. However, semantic information such as word combinations contained in a single autoregressive model is insufficient, and the performance of non-autoregressive models is vulnerable to be limited by the performance of the teacher model.

In this paper, we adopt region and grid features to deeply mine the information contained in images and design a dual branch structure that prevents the model from greater computational overhead and inference delay. Meanwhile, a

novel word retrieval module is designed for our model. It retrieves highly correlated words from the vocabulary as input for the non-autoregressive decoder, which helps the model generate the correct words in the correct positions. To enable non-autoregressive models to fully learn semantic knowledge and diverse expression methods, we propose a sequence-level knowledge distillation method with multiple teacher models, using the results generated by multiple autoregressive models as supervisory information, which alleviates the problem of word repetition caused by decoding inconsistency.

We validate the proposed dual branch non-autoregressive image captioning model on offline ‘Karpathy’ test split [14] and online test server of the MSCOCO dataset [20]. The experimental results demonstrate that the proposed model achieves new state-of-the-art performance in generation quality with a 128.8% CIDEr score. Meanwhile, our model achieves the fastest inference speed with a decoding speed improvement of 17 times<sup>1</sup>. The main contributions of this paper are summarized as follows:

- We leverage both region and grid features to fully exploit the fine-grained aspects of the image and propose a dual branch model that integrates two types of features without reducing inference speed.
- We design a word retrieval module to address the issue of missing inputs and augment the semantic richness of the inputs to the non-autoregressive decoder.
- We utilize multiple autoregressive teacher models in the knowledge distillation, thereby reducing the dependency of our model on a single autoregressive teacher model and imparting richer semantic information.
- We conduct comprehensive experiments on the MSCOCO dataset, attaining new state-of-the-art performance on both the ‘Karpathy’ offline test split and the online test server. Concurrently, our model demonstrates a significant enhancement in the inference speed.

## 2 Related Works

### 2.1 Visual Representations for Image Captioning

The early image captioning models [3] use a convolutional neural network (CNN) to extract the global features of the image to guide the caption generation. Due to the lack of fine-grained information of the image in this way, the later models [22] segment the image into grids to extract grid features, and calculate the weight of each grid with attention operations in the text generation process. So that the generated caption is more fine-grained. With the development of the object detection model, Anderson et al. [2] use Faster R-CNN [9] to extract the region features of the image, which greatly improve the generation quality of the image captioning model. In recent years, with the emergence of Transformer architecture [26] and the further development of Transformer-based object detection models such as Swin Transformer [21], the extraction of image features is

<sup>1</sup> The source code is available at <https://github.com/Liu-Yuanqiu/DBNAIC.git>.

more diversified. Although image features contain more fine-grained information, each still has its focus. Generally speaking, region features are considered to contain more object information while grid features contain contextual information because regions do not cover image regions outside of the object, while grid features cover the entire image.

## 2.2 Non-Autoregressive Image Captioning

Image captioning model [6, 12] is widely used in real life, such as blind assistant. In practical application scenarios, the lower latency is one of the most important requirements, so non-autoregressive image captioning has received widespread attention. Gao et al. [8] first explores a non-autoregressive decoding method, which uses [MASK] tokens as the input of the decoder and generates subtitles in multiple stages in parallel. It executes random masks at each stage to eliminate duplicate words. Non-autoregressive models also use methods such as iterative refinement [8] and knowledge distillation [15] to narrow the performance gap between non-autoregressive and autoregressive models. However, the quality of the non-autoregressive image captioning model is still inferior to the autoregressive models. On the one hand, non-autoregressive models lose the semantic dependence of forward words. On the other hand, issues such as insufficient visual understanding and missing decoder inputs still exist in non-autoregressive models, resulting in performance gaps. In this paper, we propose a dual branch non-autoregressive image captioning model to tackle the problem of inadequate visual understanding and design a word retrieval module to generate decoder input. More importantly, we improve the sequence-level knowledge distillation algorithm to further boost the quality of model generation.

## 3 The Proposed Method

Our overall model architecture is shown in Figure 2, which also adopts the widely used encoder-decoder architecture. The encoder part is responsible for processing image features (Sec. 3.1), the non-autoregressive decoder part is responsible for caption generation (Sec. 3.3), and both the encoder and decoder are stacked by multi-layer Transformers. Between the encoder and decoder, we use a word retrieval module (Sec. 3.2) to generate inputs for the non-autoregressive decoder. In addition, we divide the entire model into two branches, and process the grid features and region features respectively. Due to the identical structure of the two branches, we will only introduce one of them in the following paper. Finally, we merge the results of the two branches.

### 3.1 The Encoder Module

In our model, multi-head self-attention is widely used in encoders and decoders, which can be formulated as follows:

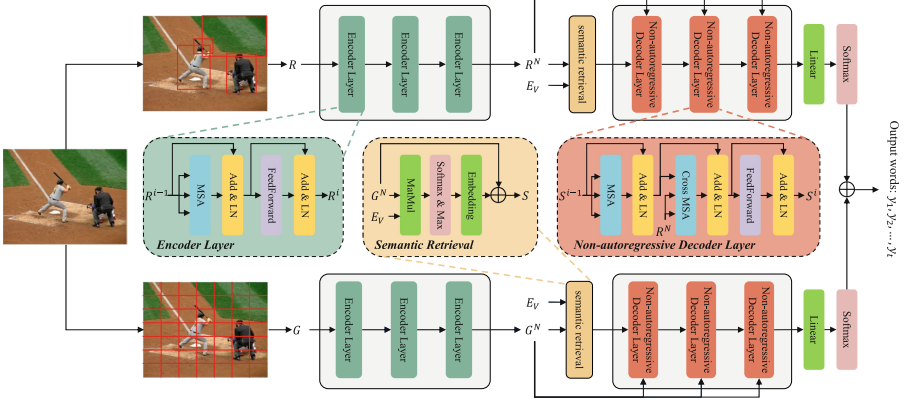
$$\text{MSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h), \quad (1)$$

$$head_i = \text{Attention}(\mathbf{Q}^i, \mathbf{K}^i, \mathbf{V}^i), i = 1, 2, \dots, h, \quad (2)$$

where  $h$  is the number of heads,  $\mathbf{Q}^i, \mathbf{K}^i, \mathbf{V}^i$  are the  $i$ -th slice of  $\mathbf{Q}, \mathbf{K}$  and  $\mathbf{V}$  respectively. The  $\text{Attention}(\cdot)$  operation uses  $\text{Softmax}(\cdot)$  to calculate the similarity score:

$$\text{Attention}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \text{Softmax}\left(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d_k}}\right)\mathbf{v}, \quad (3)$$

where  $d_k$  is the dimension of  $k$ .



**Fig. 2.** Overview of our proposed dual branch non-autoregressive image captioning model. The upper branch uses region features  $\mathbf{R}$  to generate captions, while the lower branch uses grid features  $\mathbf{G}$ . In the middle area, we present the structure of each layer of the encoder and the non-autoregressive decoder, as well as a semantic retrieval module for generating decoder input. The  $\mathbf{E}_V$  represents word embedding, and the output  $\mathbf{S}$  of the semantic retrieval module serves as the initial input of the decoder.

We use two encoders with the same architecture to extract region features and grid features respectively. Grid features  $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_m\}$  and region features  $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n\}$  are derived from the paper GRIT [23], where  $\mathbf{g}_i \in \mathbb{R}^{D_G}$ ,  $\mathbf{r}_i \in \mathbb{R}^{D_R}$ , and  $m$  is the number of grid features,  $n$  is the number of region features. Taking the processing of region features as an example, the encoder includes  $N$  layers of Transformer, where each layer of Transformer includes a multi-head self-attention module and a feedforward module. The calculation formula for the  $l$ -th layer Transformer can be written as follows:

$$\hat{\mathbf{G}}^l = \text{LayerNorm}(\mathbf{G}^{l-1} + \text{MSA}(\mathbf{W}_{E,Q}^{g,l} \mathbf{G}^{l-1}, \mathbf{W}_{E,K}^{g,l} \mathbf{G}^{l-1}, \mathbf{W}_{E,V}^{g,l} \mathbf{G}^{l-1})), \quad (4)$$

$$\mathbf{G}^l = \text{LayerNorm}(\hat{\mathbf{G}}^l + \text{FeedForward}(\hat{\mathbf{G}}^l)), \quad (5)$$

where  $\mathbf{G}^{l-1}$  denotes the output of block  $l-1$ , and  $\mathbf{G}$  is used as the input of layer 0.  $\mathbf{W}_{E,Q}^{g,l}, \mathbf{W}_{E,K}^{g,l}, \mathbf{W}_{E,V}^{g,l} \in \mathbb{R}^{D \times D}$  are learnt parameter matrices.  $\text{FeedForward}(\cdot)$

consists of two linear layers with a ReLU activation function in between, as formulated below:

$$\text{FeedForward}(\mathbf{x}) = \mathbf{W}_2^{g,l} \text{ReLU}(\mathbf{W}_1^{g,l} \mathbf{x}), \quad (6)$$

where  $\mathbf{W}_1^{g,l} \in \mathbb{R}^{(4D) \times D}$  and  $\mathbf{W}_2^{g,l} \in \mathbb{R}^{D \times (4D)}$  are learnt parameter matrices of two linear layers respectively.

Through N-layer self-attention and feedforward network calculations, we ultimately obtain the encoder output  $G^N$  of the grid features. Similarly, the encoder output  $R^N$  for region features can also be obtained.

### 3.2 The Word Retrieval Module

We propose a word retrieval module to address the issue of missing inputs in non-autoregressive decoders. The word retrieval module calculates the similarity between visual features (region features or grid features) and word embeddings in the vocabulary as a basis to obtain the retrieved words. For an example, region features are projected into the semantic space firstly:

$$\hat{\mathbf{R}}_S = \mathbf{W}_S^R \mathbf{R}^N, \quad (7)$$

where  $\mathbf{W}_S^R \in \mathbb{R}^{D \times D}$  is the projection matrix to be learned. Then, based on the vocabulary  $V = \{v_1, v_2, \dots, v_{D_v}\}$  and its corresponding embedding matrix  $\mathbf{E}_V = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{D_v}\}$ , where  $\mathbf{E}_V \in \mathbb{R}^{D_v \times D}$  and  $D_v$  is the number of words in the vocabulary, we calculate the similarity  $\mathbf{P}_R = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$  between image features and word vectors, where  $n$  is the number of the region features:

$$\mathbf{P}_R = \text{Softmax}(\hat{\mathbf{R}}_S \mathbf{E}_V^T). \quad (8)$$

We select the word with the highest similarity as the retrieval result, and then generate a set of word embeddings  $\mathbf{W}_R$ :

$$\mathbf{W}_R = \{\mathbf{e}_{\text{argmax}(\mathbf{p}_i)}\}, \quad \text{where } i \in \{1, 2, \dots, n\}, \quad (9)$$

$$\text{argmax}(\mathbf{p}_i) = \arg \max_j \mathbf{p}_{i,j}, \quad \text{where } j \in \{1, 2, \dots, D_v\}. \quad (10)$$

Finally, similar to residual networks, we add semantic vectors  $\hat{\mathbf{R}}_S$  and word embedding  $\mathbf{W}_R$  to obtain the output of the word retrieval module  $\bar{\mathbf{R}}_S$ :

$$\bar{\mathbf{R}}_S = \hat{\mathbf{R}}_S + \mathbf{W}_R. \quad (11)$$

Through word retrieval module, we get  $n$  (the number of region features) words that may appear in the caption from the entire vocabulary as input to the decoder. Similar to the calculation method of  $\bar{\mathbf{R}}_S$ , we can also obtain  $\bar{\mathbf{G}}_S$  as the input for the grid feature decoder.

However, it is still a problem that the number of grid features  $m$  and region features  $n$  do not match  $t$ , the number of features input by the decoder

(equivalent to the length of the caption). We reduce the number of visual features to the number required by the decoder through a simple method. Given  $\mathbf{R}_S = \{\bar{\mathbf{r}}_{s,1}, \bar{\mathbf{r}}_{s,2}, \dots, \bar{\mathbf{r}}_{s,n}\}$ , the  $j$ -th element of the decoder input  $\mathbf{R}_S = \{\mathbf{r}_{s,1}, \mathbf{r}_{s,2}, \dots, \mathbf{r}_{s,t}\}$  is computed as:

$$\mathbf{r}_{s,j} = \sum_i w_{ij} \cdot \bar{\mathbf{r}}_{s,i}, \quad (12)$$

$$w_{ij} = \exp\left(-\left(j - i \cdot (t/n)\right)^2 / \tau\right), \quad (13)$$

where  $\tau$  is a hyper-parameter controlling the sharpness of the function. With the same method, we obtain the decoder input  $\mathbf{G}_S$  for the grid feature branch.

### 3.3 The Non-Autoregressive Decoder Module

Like the encoder, the decoder also has two identical branches, which use grid features and region features to generate words. Firstly, we perform self-attention calculation on the input and use the obtained results with visual features to calculate cross-attention, thereby generating a probability distribution for the caption statement. Then, we directly add the probability distributions obtained from the two branches as the final generation result.

After the calculation of the word retrieval module, we finally obtain the input  $\mathbf{R}_S$  and  $\mathbf{G}_S$  for the non-autoregressive decoder. As mentioned earlier, the decoder includes self-attention blocks and cross-attention blocks. The self-attention blocks are used to model the relationship between inputs and can be accurately expressed as follows:

$$\tilde{\mathbf{R}}_S^l = \text{LayerNorm}(\mathbf{R}_S^{l-1} + \text{MSA}(\mathbf{W}_{S,Q}^{r,l} \mathbf{R}_S^{l-1}, \mathbf{W}_{S,K}^{r,l} \mathbf{R}_S^{l-1}, \mathbf{W}_{S,V}^{r,l} \mathbf{R}_S^{l-1})), \quad (14)$$

where  $\mathbf{R}_S^{l-1}$  denotes the output of block  $l-1$ , and  $\mathbf{R}_S$  is used as the input of block 0.  $\mathbf{W}_{S,Q}^{r,l}, \mathbf{W}_{S,K}^{r,l}, \mathbf{W}_{S,V}^{r,l} \in \mathbb{R}^{D \times D}$  are learnt parameter matrices.

Cross-attention blocks are used to model the relationship between semantic and visual features, the output of the  $l$ -th block is computed as follows:

$$\check{\mathbf{R}}_S^l = \text{LayerNorm}(\tilde{\mathbf{R}}_S^l + \text{MSA}(\mathbf{W}_{C,Q}^{r,l} \tilde{\mathbf{R}}_S^l, \mathbf{W}_{C,K}^{r,l} \mathbf{R}_S^N, \mathbf{W}_{C,V}^{r,l} \mathbf{R}_S^N)), \quad (15)$$

$$\mathbf{R}_S^l = \text{LayerNorm}(\check{\mathbf{R}}_S^l + \text{FeedForward}(\check{\mathbf{R}}_S^l)), \quad (16)$$

where  $\mathbf{W}_{C,Q}^{r,l}, \mathbf{W}_{C,K}^{r,l}, \mathbf{W}_{C,V}^{r,l} \in \mathbb{R}^{D \times D}$  are learnt parameter matrices and the  $\text{FeedForward}(\cdot)$  is defined in Eq. 6. After decoding at layer  $N$ , we obtain the output  $\mathbf{R}_S^N$  of the region feature branch. The output  $\mathbf{G}_S^N$  of the grid feature branch is obtained using the same method as the region feature branch.

According to the output of the decoder, the conditional distribution over the vocabulary  $V$  is given by:

$$p(Y|R) = \text{Softmax}(\mathbf{W}_F^R \mathbf{R}_S^N), \quad (17)$$



$$p(Y|G) = \text{Softmax}(\mathbf{W}_F^G \mathbf{G}_S^N), \quad (18)$$

where  $p(Y|R)$  and  $p(Y|G)$  represent the probability distributions obtained from the region feature branch and the grid feature branch respectively.  $\mathbf{W}_F^R, \mathbf{W}_F^G \in \mathbb{R}^{D_v \times D}$  are learnt parameters and  $D_v$  is the number of words in the vocabulary,  $Y = \{y_1, y_2, \dots, y_t\}$  is the generated caption, and  $t$  is the length of a caption.

### 3.4 Objective Functions

In this paper, we also use the cross-entropy loss function to optimize the model as the classical non-autoregressive image captioning models do. The loss of the whole model is obtained by adding the losses of the two branches:

$$L_{XE}(\theta) = - \sum_{t=1}^l \log(p(y_t^*|R)) - \sum_{t=1}^l \log(p(y_t^*|G)). \quad (19)$$

where  $\theta$  denotes the parameters of the whole model, and  $y_t^*$  denotes the target ground truth word.

On the other hand, sequence-level knowledge distillation algorithms are widely used in non-autoregressive decoding to improve the generation quality. It owes to that knowledge distillation algorithms can bring much more information to student models than normal training methods. In this paper, we also employ the knowledge distillation algorithm to convert the supervised signal of the model from target ground truth  $Y^* = \{y_1^*, y_2^*, \dots, y_t^*\}$  to the caption  $\hat{Y}^*$  generated by the autoregressive model. We use multiple autoregressive models instead of a single one as the teacher model. The captions generated by multiple teacher models are constructed as training sets, greatly enhancing the diversity of training samples.

The final loss function is formulated as follows:

$$L_{XE}(\theta) = - \sum_{i=1}^t \log(p(\hat{y}_i^*|R)) - \sum_{j=1}^t \log(p(\hat{y}_j^*|G)). \quad (20)$$

## 4 Experiments

### 4.1 Datasets

We conducted experiments on the popular MSCOCO dataset [20] in the image captioning task, which includes 123287 images, and each was annotated with 5 reference captions. In this paper, we follow the widely used ‘Karpath’ split [14], where 113287 images for training, 5000 images for validation, and 5000 images for testing. During the training phase, we extract words that appear more than 5 times from the training set to form a vocabulary.

## 4.2 Baselines

The comparative models include AIC [11], PNAIC [7], SATIC [32], SAIC [31], MNIC [8], FNIC [4], and CMAL [11]. These models are classified into three categories: autoregressive models, partially non-autoregressive models and non-autoregressive models. The scores and latency of the models come from their papers and the speedups are recalculated based on the data reported in their paper using AIC (bw=3) as the benchmark.

## 4.3 Evaluation Metrics

We adopt the widely used metrics to evaluate the quality of the generated captions and compare with other methods, including BLEU-1/4 [24], METEOR [17], ROUGE-L [19], and CIDEr [27].

**Table 1.** Generation quality, latency, and speedup on MSCOCO “Karpathy” split. “-” denotes that the results are not reported. Latency is the time to decode a single image without mini batching, averaged over the whole test split, and tested on a GeForce GTX 1080 Ti GPU.

Models	BLEU-1↑	BLEU-4↑	METEOR↑	ROUGE↑	CIDEr↑	Latency↓	SpeedUp↑
<b>Autoregressive models</b>							
AIC(bw=1) [11]	79.8	38.4	29.0	58.7	126.6	134ms	1.66×
AIC(bw=3) [11]	80.3	38.9	29.1	58.9	128.8	222ms	1.00×
<b>Partially Non-autoregressive models</b>							
PNAIC [7]	79.9	37.5	28.2	58.0	125.2	32ms	6.94×
SATIC [32]	80.6	37.6	28.3	58.1	126.2	35ms	6.34×
SAIC [31]	80.3	38.4	29.0	58.2	127.1	54ms	4.11×
<b>Non-autoregressive models</b>							
MNIC [8]	75.4	30.9	27.5	55.6	108.1	61ms	3.64×
FNIC [4]	-	36.2	27.1	55.3	115.7	-	8.15×
CMAL [11]	80.3	37.3	28.1	58.0	124.0	16ms	13.88×
Ours	<b>81.7</b>	<b>39.5</b>	<b>28.8</b>	<b>59.4</b>	<b>128.8</b>	<b>13ms</b>	<b>17.08×</b>

## 4.4 Experimental Settings

The embedding size  $D$  of the region and grid features is set to 512, the number of transformer heads is 8, and the number of layers  $N$  for the encoder and decoder of each branch is 3. In the training process, we employ GRIT [23], PureT [28] and COSNet [18] as teacher models, and train the models under the cross-entropy loss  $L_{XE}$  for 90 epochs. The hyperparameter  $\tau$  in the semantic retrieval module is set to 0.5, and the length of the generated caption  $t$  is 20, as the vast majority of captions do not exceed 20 in length. Adam [16] optimizer is used during the training phase.

**Table 2.** Results on online MSCOCO testing server.

Models	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
<b>Partially Non-autoregressive models</b>														
PNAIC [7]	80.1	94.4	64.0	88.1	49.2	78.5	36.9	68.2	27.8	36.4	57.6	72.2	121.6	122.0
SAIC [31]	80.0	94.5	64.1	88.2	49.2	78.8	37.2	67.8	28.0	36.8	57.7	72.4	121.4	123.7
<b>Non-autoregressive models</b>														
CMAL [11]	79.8	94.3	63.8	87.2	48.8	77.2	36.8	66.1	27.9	36.4	57.6	72.0	119.3	121.2
Ours	<b>80.9</b>	<b>95.3</b>	<b>65.1</b>	<b>88.5</b>	<b>50.2</b>	<b>79.0</b>	<b>38.1</b>	<b>68.4</b>	<b>28.3</b>	<b>37.1</b>	<b>58.8</b>	<b>73.4</b>	<b>121.5</b>	<b>122.9</b>

## 4.5 Offline Evaluation

Table 1 reports the performance of the compared models and our proposed dual non-autoregressive image captioning model on the offline test set of the MSCOCO dataset [20]. Our model achieves state-of-the-art performance in all evaluation metrics. Not only does it surpass all existing non-autoregressive models in all evaluation metrics, but it also surpasses all partially non-autoregressive models in all metrics except METEOR, which is only 0.2% lower than the best partially non-autoregressive model. Compared with the strong baseline CMAL [11], our model achieves more than 1.4% on BLUE-1, BLUE-4, ROUGE, and CIDEr metrics. Especially for the CIDEr score, our model achieves 128.8%, which is a new-state-of-art performance in image captioning with non-autoregressive models. Even compared with the best partially non-autoregressive model SAIC [31], our model still achieves a performance improvement of 1.0% on multiple evaluation metrics, while the other non-autoregressive image captioning models never exceeded the partially non-autoregressive model in performance. On the other hand, our model, as a non-autoregressive model, matches or even surpasses the performance of autoregressive models using reinforcement learning. In terms of BLUE-4 and ROUGE metrics, the performance of our model is about 0.5% higher than AIC (bw=3), while 1.4% higher in BLUE-1. Our method performs similar or better compared with the autoregressive models on four of the five metrics.

As a non-autoregressive model, inference delay is also an aspect that must be paid attention to. Our model only requires 13 ms in inference speed, achieving optimal performance. Compared to other partially non-autoregressive and non-autoregressive models, we achieved the maximum speed improvement, the inference speed of our model is 17 times faster than the autoregressive model.

## 4.6 Online Evaluation

Table 2 reports the performance on the official online testing server of MSCOCO dataset [20]. In online testing, five reference descriptions (c5) and forty reference descriptions (c40) are used for evaluation. Compared with the existing non-autoregressive image captioning model, our model achieves the best performance

on all evaluation metrics. Especially for CIDEr scores, we achieved 121.5% (c5) and 122.9% (c40), exceeding the optimal performance model CMAL [11] by 2.2% and 1.7%, respectively. Compared with the partially non-autoregressive model, our model also achieves the best score, except for a slightly lower on CIDEr (c40) than SAIC [31].

In summary, the significant improvement on different evaluation metrics of our proposed dual branch non-autoregressive image captioning model demonstrates its advantages, and the greatest acceleration in inference speed of model greatly accelerates the non-autoregressive image captioning.

**Table 3.** Comparison of performance between single branch model and dual branch model.

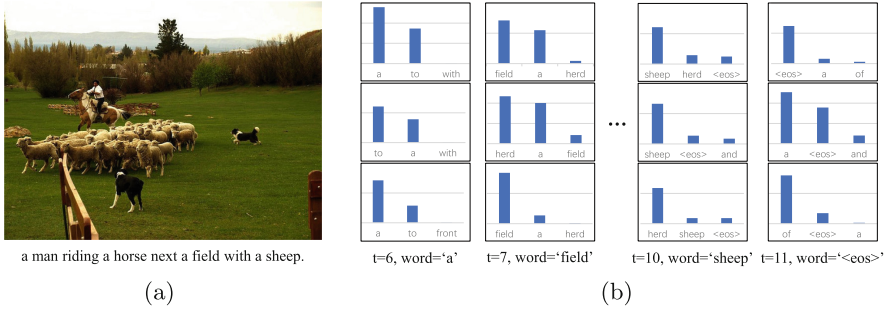
Models	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr
Ours w/o G	80.7	38.1	28.5	58.6	125.3
Ours w/o R	80.5	38.1	28.4	58.6	124.7
Ours	<b>81.7</b>	<b>39.5</b>	<b>28.8</b>	<b>59.4</b>	<b>128.8</b>

#### 4.7 Ablation Study

To demonstrate the effectiveness of the method proposed in this paper, we designed multiple ablation experiments for validation.

**Influence of Region Branch and Grid Branch.** Our model has a two-branch structure for processing regional features and grid features to cover different aspects of image information, and the results generated by these two branches are merged at the end. As shown in Table (reftab:branch), Ours w/o G and Ours w/o R represent the scores of the region branch and the grid branch, respectively. Using branch alone results in a performance degradation, specifically a 3.5% drop in the CIDEr score for the regional branch and a 4.1% drop for the grid branch. It shows that the two branches focus on different types of image information and that combining the results of both can further improve the quality of the non-autoregressive model. Moreover, since the two branches are independent of each other and can be computed in parallel, there is no significant delay in the inference speed of the model.

We also visualize the contribution of region and grid branches, as shown in Figure 3. In the figure, we show the largest probability distributions generated by the two branch and our model. Most of the time, the results generated by the region branch (the second column in Figure 3(b)), the grid branch (the third column in Figure 3(b)), and the dual branches (the first column in Figure 3(b)) are the same, but the region branch and the grid branch diverge sometimes. It can be found that when there is a deviation in the generated results of a



**Fig. 3.** The contribution of the region branch and grid branch to the accuracy of generated captions. 3a displays the image and the generated caption, while 3(b) displays the generation probability value of our model and two branches, with only the top three display. The first column in 3(b) represents the generation probability values of our model. The second column represents the generation probability values of the region branch, and the third column represents the generation probability value of the grid branch.

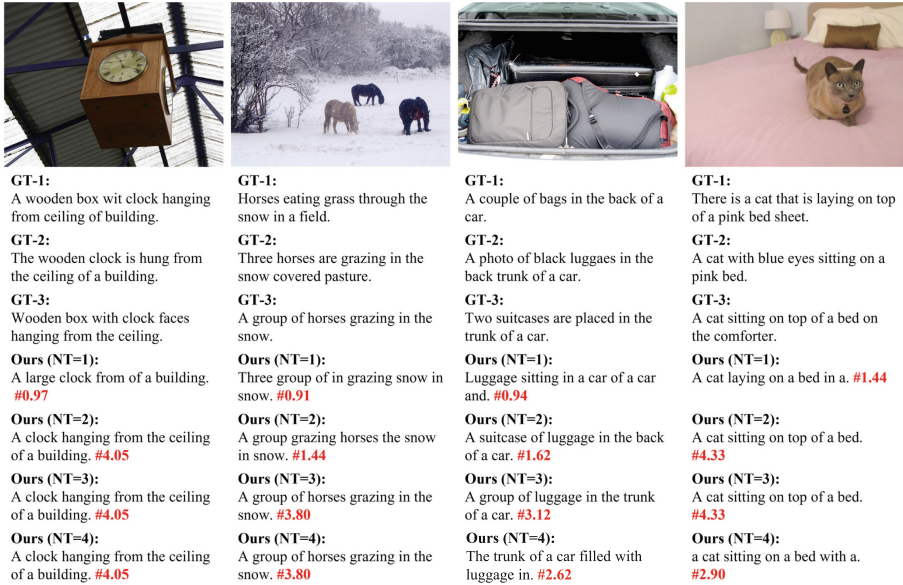
certain branch, the combination of the generated results of two branches can often generate the most suitable word, which proves the effectiveness of our model with dual branches.

**Table 4.** Performance comparison of using different numbers of teacher models. NT represents the number of teacher models.

NT	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr
1	<b>82.3</b>	38.2	28.3	58.3	126.2
2	80.8	38.1	28.2	58.6	125.0
3	81.7	<b>39.5</b>	<b>28.8</b>	<b>59.4</b>	<b>128.8</b>
4	82.2	39.4	28.4	59.0	128.1

**Influence of the Number of Teacher Models.** To explore the role of multi teacher models in sequence-level knowledge distillation, we conduct experiments with different numbers of teacher models and compare them with the complete model. As shown in Table 4, our model can achieve good results on BLUE-1 when supervised using a teacher model, however, the scores on relevant metrics concerned with the quality of sentence-level generation (e.g., BLUE-4 and CIDEr) are low. This is mainly due to the fact that high accuracy at the word level corresponds to multiple reference captions. As shown in Figure 4, although each position generates words that correspond to a real caption, the lack of sufficient word combination information leads to duplicate words and semantic

incoherence in the generated captions, resulting in lower accuracy for BLUE-4 and CIDEr.



**Fig. 4.** Examples of captions generated by our dual branch non-autoregressive image captioning model (Ours) with ground truths (GT-1/2/3), and Ours (NT= $i$ ) represents our model using  $i$  teacher models. The red number represents the corresponding CIDEr score for the caption.

Meanwhile, it also performs poorly with two teacher models which is largely due to the divergence of the two models. When using three teacher models, we achieve the optimal score on multi evaluation metrics. It's mainly because at least two models will reach consensus in most cases during the training process, which is similar to the case of human voting in reality. We also train our model with four teacher models and obtain slightly lower results compared with three teacher models. It can be concluded that the model has learned enough semantic knowledge from the three teacher models.

#### 4.8 Visualization Analysis

We display some examples of generated image captions in Figure 4. It can be seen that some challenging issues, such as semantic inconsistency and repetition are solved in the generated captions. Our proposed model not only recognizes objects in the images effectively but also ensures that the generated sentences are complete and coherent, effectively conveying the content depicted in the image. Some of these sentences are even basically the same as the sentences annotated by humans.

## 5 Conclusion

In this paper, we propose a novel dual branch non-autoregressive image captioning model. Without affecting the reasoning speed of the model, region features and grid features are fused to improve the quality of model generation. Meanwhile, we design a word retrieval module to enrich the semantic information contained in the input. At the same time, by using sequence-level knowledge distillation algorithms with multiple teacher models, our model can learn richer word combination information and generate more accurate and coherent captions. Experiments on the MSCOCO dataset have shown that our proposed model achieves state-of-the-art performance with a 128.8% CIDEr score and the inference speed is increased by 17 times faster than the autoregressive model.

**Acknowledgement.** This work was supported by National Natural Science Foundation of China (No. 62106035), Liaoning Binhai Laboratory Project (No. LBLF-2023-01), and Chunhui Project Foundation of the Education Department of China (No. HZKY20220419). We also would like to thank Dalian Ascend AI Computing Center and Dalian Ascend AI Ecosystem Innovation Center for providing inclusive computing power and technical support.

## References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6077–6086 (2018)
2. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6077–6086 (2018)
3. Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J.C., et al.: From captions to visual concepts and back. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1473–1482 (2015)
4. Fei, Z.: Fast image caption generation with position alignment. CoRR [abs/1912.06365](https://arxiv.org/abs/1912.06365) (2019)
5. Fei, Z.: Iterative back modification for faster image captioning. In: ACM International Conference on Multimedia (ACM MM). p. 3182-3190 (2020)
6. Fei, Z.: Memory-augmented image captioning. In: AAAI Conference on Artificial Intelligence (AAAI). pp. 1317–1324 (2021)
7. Fei, Z.: Partially non-autoregressive image captioning. In: AAAI Conference on Artificial Intelligence (AAAI). pp. 1309–1316 (2021)
8. Gao, J., Meng, X., Wang, S., Li, X., Wang, S., Ma, S., Gao, W.: Masked non-autoregressive image captioning. CoRR [abs/1906.00717](https://arxiv.org/abs/1906.00717) (2019)
9. Girshick, R.: Fast r-cnn. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1440–1448 (2015)
10. Gu, J., Wang, G., Cai, J., Chen, T.: An empirical study of language cnn for image captioning. In: IEEE international conference on computer vision (ICCV). pp. 1222–1231 (2017)




11. Guo, L., Liu, J., Zhu, X., He, X., Jiang, J., Lu, H.: Non-autoregressive image captioning with counterfactuals-critical multi-agent learning. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 767–773 (2020)
12. Huang, L., Wang, W., Chen, J., Wei, X.Y.: Attention on attention for image captioning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4634–4643 (2019)
13. Ji, J., Luo, Y., Sun, X., Chen, F., Luo, G., Wu, Y., Gao, Y., Ji, R.: Improving image captioning by leveraging intra-and inter-layer global representation in transformer network. In: AAAI Conference on Artificial Intelligence (AAAI). pp. 1655–1663 (2021)
14. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3128–3137 (2015)
15. Kim, Y., Rush, A.M.: Sequence-level knowledge distillation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1317–1327 (2016)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
17. Lavie, A., Agarwal, A.: Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In: Workshop on Statistical Machine Translation. pp. 228–231 (2007)
18. Li, Y., Pan, Y., Yao, T., Mei, T.: Comprehending and ordering semantics for image captioning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17990–17999 (2022)
19. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81 (2004)
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV). pp. 740–755. Springer (2014)
21. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10012–10022 (2021)
22. Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 375–383 (2017)
23. Nguyen, V.Q., Suganuma, M., Okatani, T.: Grit: Faster and better image captioning transformer using dual visual features. In: European Conference on Computer Vision (ECCV). pp. 167–184 (2022)
24. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: Annual Meeting of the Association for Computational Linguistics (ACL). pp. 311–318 (2002)
25. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7008–7024 (2017)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS. pp. 5998–6008 (2017)
27. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4566–4575 (2015)
28. Wang, Y., Xu, J., Sun, Y.: End-to-end transformer based model for image captioning. In: AAAI Conference on Artificial Intelligence (AAAI). pp. 2585–2594 (2022)



29. Xian, T., Li, Z., Zhang, C., Ma, H.: Dual global enhanced transformer for image captioning. In: *Neural Networks*. vol. 148, pp. 129–141 (2022)
30. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: *International Conference on Machine Learning (ICML)*. pp. 2048–2057 (2015)
31. Yan, X., Fei, Z., Li, Z., Wang, S., Huang, Q., Tian, Q.: Semi-autoregressive image captioning. In: *ACM International Conference on Multimedia (ACM MM)*. pp. 2708–2716 (2021)
32. Zhou, Y., Zhang, Y., Hu, Z., Wang, M.: Semi-autoregressive transformer for image captioning. In: *IEEE International Conference on Computer Vision Workshops (ICCVW)*. pp. 3132–3136 (2021)



# Distill the Knowledge of Multimodal Large Language Model into Text-to-Image Vehicle Re-identification

Jianshu Zeng<sup>1</sup>  and Chi Zhang<sup>2</sup>  

<sup>1</sup> University of Chinese Academy of Sciences, Beijing 101408, China  
zengjianshu21@mailsucas.ac.cn

<sup>2</sup> Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China  
chi.zhang@ia.ac.cn

**Abstract.** Text-to-Image Vehicle Re-identification(TIVReid) aims to retrieve the target vehicle image according to a given description. For this task, efficient feature alignment of image and text modalities is crucial yet constrained by the lack of large-scale, high-quality datasets. Recently, the multimodal Large Language Model(MLLM) has shown remarkable performance in image-text understanding, which motivates this paper to explore the application of MLLM in TIVReid. We propose an effective method to distill the knowledge of MLLM into the TIVReid model with the following innovations: Firstly, we propose a prompt design approach that introduces the attribute-guided pre-prompt and optimized few-shot policy to guide MLLM to generate high-quality descriptions. Secondly, we devise a two-stage aligning strategy to better utilize the generated data. We relax the alignment on the non-target domain(generated data) in stage-1 and then enhance it on the target domain in stage-2. Finally, sufficient experiments have demonstrated the effectiveness of our method and that the generated data are comparable to or even superior to human-annotated data. Our method achieves significant improvement by 6.7%, 7.6%, and 4.9% in Rank-1, Rank-5, and mAP respectively, compared to the SOTA model on the T2I-VeRi dataset. Code and dataset will be open-sourced at <https://github.com/Fly-ShuAI/TIVR2>.

**Keywords:** Text-to-Image Vehicle Re-identification · Text-based Vehicle Retrieval · Multimodal large language model

## 1 Introduction

Text-to-Image Vehicle Re-identification(TIVReid) aims to retrieve the target vehicle image from a large set based on a given description[10]. Existing vehicle

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78456-9\\_22](https://doi.org/10.1007/978-3-031-78456-9_22).

re-identification tasks primarily focus on pure visual modality[20, 32, 33]. However, in many real-world application scenarios, we can only obtain text descriptions of the target vehicle. Therefore, the TIVReid task is highly worthwhile to study.

Currently, research on this task is in its early stages. Only Ding et al.[10] proposed a dataset for TIVReid, called T2I-VeRi. However, due to the limitations of manual annotation, the text descriptions of vehicles are relatively brief and some are insufficient for retrieving the target vehicle. The lack of high-quality large-scale data makes it difficult to effectively align features across the two modalities, resulting in lower performance of existing models.

Recently, Multimodal Large Language Model(MLLM) (such as GPT4-V[24]) has shown excellent performance in multiple multimodal tasks[12]. MLLM possesses a wealth of knowledge and powerful multimodal understanding capabilities but we can only interact with it through conversation, which precludes their direct use for the large-scale training and inference.

Motivated by the main problem in TIVReid and the power of MLLM, this paper aims to explore the potential applications of MLLM in TIVReid. We explore an effective method to distill the knowledge of MLLM into the TIVReid model: First, through attribute-guided pre-prompt and optimized few-shot policy, we leveraged GPT4-V to construct a new large-scale, high-quality Text-to-Image Vehicle Re-identification Dataset, called TIVRD2. Following this, we devise a two-stage aligning strategy called **Global feature First, Local feature added After**(GFLA), to better utilize the generated vehicle image-text pairs. We first relax the alignment on the non-target domain(generated data) to prevent overfitting and then fine-tune the model on the target domain with a more comprehensive alignment to enhance its capability. Through our method, Rank-1, Rank-5, and mAP have increased by 6.7%, 7.6%, and 4.9% respectively compared to the state-of-the-art model on commonly used human-annotated dataset T2I-VeRi[10], proving the effectiveness of our proposed method. Our main contributions are as follows:

1. We are the first to explore the potential application of MLLM in TIVReid. We propose an effective method to distill the knowledge of MLLM into the TIVReid model, which achieves significant improvement over existing models.
2. We propose a useful prompt design approach to guide MLLM to generate high-quality vehicle descriptions. We have demonstrated the effectiveness of this approach by designing different types of prompts and conducting systematic comparative experiments.
3. We devise an effective two-stage aligning strategy to better utilize the generated data. This strategy is validated through sufficient experiments and can be easily extendable to other multimodal models that need two-stage training.
4. We have demonstrated that the quality of the generated dataset TIVRD2 is comparable to or even superior to that of the human-annotated dataset T2I-VeRi, which offers a low-cost, simple and effective solution to the current data shortage problem in TIVReid. Our TIVRD2 dataset contains 23,780 high-quality vehicle image-text pairs, nearly ten times the size of T2I-VeRi

(2,458 pairs). We will open-source our dataset and code to facilitate research in TIVReid.

## 2 Related Work

### 2.1 Multimodal Large Language Models

In recent years, Large Language Models(LLM) have achieved tremendous success in the field of natural language processing. Early encoder-decoder models such as BERT[9], as well as models primarily based on the decoder, such as GPT-1[25], leveraging the Transformer[27] architecture, have achieved excellent performance on a variety of NLP tasks. GPT-3[4] was trained on a massive corpus of text data, and with further instruction-based fine-tuning, ChatGPT was developed. ChatGPT and subsequent GPT-4[1] have demonstrated powerful capabilities across multiple application scenarios. These LLMs compress and learn information from vast amounts of text data, possessing broad understanding and generative abilities.

Recently, the capability of LLM has been further extended to include image modalities. Latest LLMs introduce vision-language models and are trained on large amounts of image-text data, thereby gaining powerful multimodal understanding capabilities[30], known as Multimodal Large Language Models(MLLM), such as GPT4-V[24], Google Gemini, Claude-3, etc. These models have demonstrated excellent performance across multiple multimodal downstream tasks[12], providing new insights and methods for the research of multimodal tasks.

For example, [28] propose a framework using MLLM to filter image-text pairs, this method outperforms the existing popular methods CLIPScore via integrating the recent advances in MLLM. [16] investigate the capabilities of MLLM in detecting AI-generated human face images(called DeepFakes), They find the performance of GPT4-V is better than the early methods, and it is noteworthy that these methods are trained on a large-scale face image dataset. [5] use GPT4-V to generate high-quality captions for 100K diverse images, and replace the image-text pairs utilized in the Supervised Fine-Tuning(SFT) stage of several typical MLLMs(containing LLaVA[18], Qwen-VL[2]) with an equivalent quantity. Then they re-benchmark these models and the results show that they use a small equivalent substitution but get consistent performance improvements across various MLLMs and benchmarks. In the field of Autonomous Driving(AD), [8] and [29] systematically review the exploration of MLLM in AD.

The potential applications of MLLM in various multimodal tasks are gradually being explored, but that in TIVReid is still in its infancy.

### 2.2 Text-to-Image Vehicle Re-identification

Text-to-Image Vehicle Re-identification(TIVReid) faces challenges from both image-based vehicle re-identification [19–21, 31–33] and image-text retrieval

tasks[15,22]. The key point of TIVReid is to align the image and text features fine-grainedly and effectively, which typically requires plenty of high-quality image-text pairs.

As far as we know, there is only one dataset for Text-to-Image Vehicle Re-identification task[10], called T2I-VeRi. T2I-VeRi is constructed based on VeRi-776[20], a widely used dataset for Vehicle Re-identification(image-based), containing 776 different vehicles with 49,357 images in total, captured by 20 non-overlapping cameras. T2I-VeRi first selects 3-4 images for each vehicle in different views from VeRi-776, and then manually annotates the text descriptions for each image. Finally, T2I-VeRi contains 2458 vehicle-text pairs, partitioned into the training set and test set for the TIVReid task.

But T2I-VeRi has some limitations: (1) The number is small. The TIVReid task is challenging. The dataset is too small to train a robust deep-learning model. (2) Cost is high. The dataset is manually annotated, which is time-consuming, labor-intensive and expensive. (3) Quality has room to improve. Due to the limitation of manual annotation, most of the text descriptions are relatively short and some are insufficient to retrieve the target vehicle.

MCANet[10] is the current SOTA model for the TIVReid task. It uses ResNet-50[13] and BERT[9] as the image and text encoder and designs a multi-scale multi-view structure to align the image and text features both globally and locally. However, its performance on T2I-VeRi is still far from satisfactory, the Rank-1, Rank-5, and mAP are only 0.261, 0.571 and 0.195 respectively. Since similar model architectures can achieve a not bad performance in Text-to-Image Person Retrieval(TIPReid)[6], which is a similar task, the low performance of MCANet is likely due to the lack of large-scale, high-quality data.

### 3 Methodology

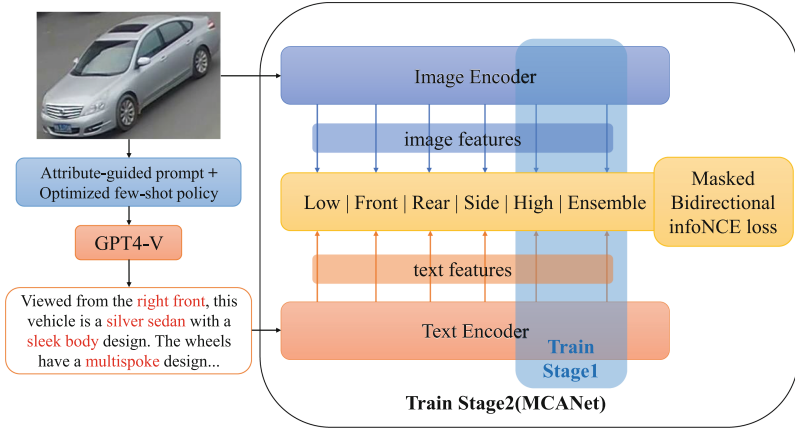
Aiming to explore the potential applications of MLLM in TIVReid, we propose an effective method to distill the knowledge of MLLM into the TIVReid model. Fig 1 shows the framework of our method, which will be introduced afterwards.

#### 3.1 Apply MLLM to TIVReid

In this section, we introduce how to distill the knowledge of MLLM into TIVReid.

MLLM possesses rich knowledge and powerful multimodal understanding capabilities. but we can only interact with MLLM through conversation, which precludes their direct use for the training and inference in TIVReid. So, we propose an effective method to distill the knowledge of MLLM into TIVReid. Hence, we have developed a useful prompt design method that introduces attribute-guided pre-prompt and optimized few-shot policy to enable MLLM to generate high-quality descriptions for vehicle images.

**Attribute-guided pre-prompt:** As shown in 2, the pre-prompt includes attribute tips such as vehicle type, color, view, etc. Furthermore, we design



**Fig. 1.** The framework of our method. We first guide MLLM to generate high-quality descriptions with the attribute-guided pre-prompt and optimized few-shot policy. Then we train the model with the proposed two-stage aligning strategy: We relax the alignment on the generated data in stage-1 and enhance it on the target domain in stage-2.


some attribute options in brackets. In this way, we can guide MLLM to generate descriptions with distinctive and sufficient attributes for vehicle images.

**Optimized few-shot policy:** Next, we carefully select five vehicle images of representative and different vehicle types, colors, and views, and then manually design detailed and distinctive descriptions based on these images. Moreover, the example descriptions all start with a similar sentence structure. In this way, under normal circumstances, the beginning of MLLM’s response will also follow a similar sentence structure, containing information about the view, color, and vehicle type attributes, followed by some specific detailed descriptions.

Through this prompt design approach, we guide MLLM to generate high-quality descriptions for vehicle images, during which the rich knowledge and powerful multimodal understanding capabilities of MLLM were distilled into high-quality vehicle descriptions, and further into the TIVReid model by training on the generated data with the proposed two-stage aligning strategy.

### 3.2 Dataset construction

In this section, we introduce the construction details of our dataset TIVRD2. Veri-776[20] contains 776 different vehicles with 49,357 images, captured by 20 cameras. ‘0001\_c001.00016450\_0.jpg’ is an example name: ‘0001’ is the vehicle ID(one vehicle has one ID. Different vehicles have different IDs.), ‘c001’ is the camera ID, and ‘16450’ is the timestamp, as shown in Fig 3. We use the word ID-Camera to represent the combination of vehicle ID and Camera ID, such as ‘0001\_c001’. Images with the same ID-Camera usually have a similar appearance. We count 5,145 different ID-Cameras in the train set and 1,677 different ID-Cameras in the test set. For each ID-Camera, we select about 4 images in



Viewed from the **right front**, this vehicle is a **yellow car** with a large front windshield, a rectangular glass **sunroof** on the roof, and a **little yellow paper** on the right window.

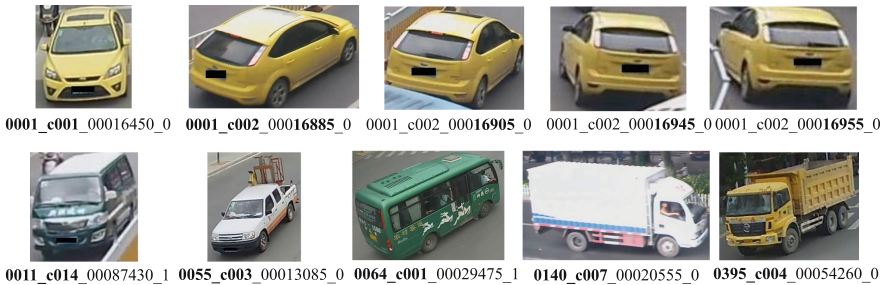
Viewed from the **left front**, this vehicle is a **white SUV** with a large rectangular glass **skylight** on the roof, **two luggage racks** on both sides, **two silver metal cross bars** in the middle of the air inlet, and the **Volkswagen logo** in the middle. We could see the **reflection of trees** in the front window and some **objects** inside through the window.

Viewed from the **right rear**, this vehicle is a **white pickup truck** with a **silver-white chrome bumper** on the back, a **frame of cargo** in the back, and **yellow strips** around the bottom.

Viewed from the **left rear**, this vehicle is a **dark gold minivan** with **roof stiffeners** on the roof, **three windows** on the left side. On the back of the car, there is a **red V-badge** in the center with **red and white taillights** on both sides.

Viewed from the **right front**, this vehicle is a **small white truck** with a **square enclosed share** in the back, a **blue sticker with three white letters** on the front of its roof, a line of **green spray paint** under the left side of the car body and **two people** on the seat.

**Fig. 2. 5-shot Prompt:** consists of pre-prompt and 5 optimized image-text pairs. In pre-prompt, we use **red** color to represent the attribute **tips**, and use **bold** to represent the attribute **options**. In the following 5 descriptions, we use **blue** color to represent the **key features** of the vehicle. **Pre-prompt:** Give a paragraph of about 60 words to describe the vehicle in detail, including the **view**(**front, left front, right front, left rear, right rear, rear**), **color, vehicle type**(**car, SUV, van, bus, pickup, truck**), and any **distinguishing features** on the **front, side, or rear** of the vehicle, etc. Here are five examples of descriptions.



**Fig. 3.** How to select images: For each ID-Camera, we select images with different timestamps, which ensures the diversity of the dataset TIVRD2: different vehicle types, colors, and views. ‘0001\_c001’ means the ID-Camera of the vehicle. ‘16450’ is the timestamp.

the train set and about 2 images in the test set, such that the timestamp difference between the selected images is as large as possible. ( One image with the minimum timestamp, one image with the maximum timestamp, one or two images with the median timestamp(if there are still images left). ) This is because images with big timestamp differences usually have different views. In this way, we get images with a remarkable diversity of ID, camera, and view. Furthermore, based on the diverse images, we use MLLM to generate text annotations for each image, which naturally leads to diverse text annotations.

Then, we use 5-shot prompt shown in 2 to request GPT-4-vision-preview API to generate the text descriptions. After generating, we manually check the quality of the descriptions and remove the unusable replies, which may contain refusals(“I am sorry...”), strange chars(“...json”), descriptions about multiple vehicles, etc. After repeating the request-check process several times, we finally get a large-scale high-quality dataset, named TIVRD2 since it is the second **T**ext-to-**I**mage **V**ehicle **R**e-identification **D**ataset. It contains 20,460 vehicle image-text pairs for training and 3,320 pairs for testing, Table 1 shows the comparison of T2I-VeRi, TIVRD2 and some commonly used datasets for vehicle re-identification(ReID) and Text-to-Image Person Reidentification(TIPReid).

**Table 1.** Comparison the datasets of ReID, TIPReid and TIVReid tasks.

task	dataset	average length	cameras	ids	images or pairs
Vehicle ReID	VeRi-776[20]	n/a	20	776	49,357
	CityFlow[31]	n/a	40	666	56,277
	VehicleID[19]	n/a	12	26,267	221,763
	VERIWild[21]	n/a	174	40,671	416,314
TIPReid	RSTPReid	23	-	4,101	20,505
	CUHK-PEDES	23.5	-	13,003	40,206
	ICFG-PEDES	37.2	-	4,102	54,522
TIVReid	T2I-VeRi[10]	27.6	-	776	2,458
	TIVRD2(Ours)	<b>64.6</b>	20	776	<b>23,780</b>

### 3.3 Two Stage Aligning Strategy

In this section, we introduce the proposed two-stage aligning strategy.

Text-to-Image Vehicle Re-identification aims to learn a function  $\mathcal{F}$  that maps the input image-text pair  $(T, I)$  ( $T$  donates the text,  $I$  donates the image) to a feature space, such that the distance of the positive pairs  $(f_T, f_I^+)$  is minimized and that of the negative pairs  $(f_T, f_I^-)$  is maximized as much as possible. Usually,  $\mathcal{F}$  is a deep neural network, trained by minimizing a loss function on a training set  $X = \{x_i = (T_i, I_i)\}_{i=1}^N$  with the annotate labels  $Y = \{(id_T, id_I)\}$  ( $id_T$  or  $id_I$



donates the corresponding vehicle ID of the image or text), that is:

$$\min_{\mathcal{F}} \sum_{i=1}^N \mathcal{L}(\mathcal{F}(T_i, I_i), y_i) \quad (1)$$

where  $\mathcal{L}(\cdot, \cdot)$  is the loss function,  $(f_T, f_I)$  and  $\mathcal{F}(T_i, I_i)$  are the feature representation of the image-text pairs.

The current SOTA method[10] designs a multi-view multi-scale cross-modal alignment network MCANet( $\mathcal{F}$ ) and a masked bidirectional infoNCE loss( $\mathcal{L}$ ) to align the image and text features both globally and locally. It uses ResNet-50[13] as the image encoder, where the fourth layer is multi-branched, and BERT[9] appended corresponding multi-branched convolutional layers as the text encoder. It gets six image-text feature pairs  $\{(f_T, f_I)\}_{i=1}^6$ , representing the front view, rear view, side view, low scale, high scale and ensemble feature, respectively. Fig 1 shows the brief structure of MCANet.

We take MCANet as our feature extraction model  $\mathcal{F}$ . To better align the image and text features by utilizing the large-scale high-quality vehicle image-text pairs in TIVRD2, we propose an effective two-stage aligning strategy, called **Global feature First, Local feature added After**(GFLA), that is:

**Stage-1:** In stage-1, we train the model on TIVRD2, the non-target domain. Unavoidably, the generated data has a little domain gap with the human-annotated data(target domain). So we relax the alignment requirement and only align the global feature(high scale and ensemble feature mentioned above), to learn a universal global feature alignment capability and prevent overfitting.

InfoNCE loss[23] is commonly used to align two feature spaces:

$$\mathcal{L}_{InfoNCE}(q, k) = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(q_i \cdot k_i / \tau)}{\sum_{j=1}^N \exp(q_i \cdot k_j / \tau)} \quad (2)$$

where  $q_i, k_i$  donates  $i$ -th feature pair,  $\tau$  is the temperature parameter,  $N$  is the batch size. Minimizing the InfoNCE loss means enlarging the similarity of positive pairs  $\exp(q_i \cdot k_i / \tau)$  and reduce the sum of similarities of negative pairs  $\sum_{j=1}^N \exp(q_i \cdot k_j / \tau)$ . We use global contrastive loss  $\mathcal{L}_{global}$  to train the model in stage-1:

$$S_{ij} = \exp(q_i \cdot k_j / \tau) \quad (3)$$

$$\mathcal{L}_{mb}(q, k) = -\frac{1}{N} \sum_{i=1}^N \log \frac{S_{ii}}{S_{ii} + \sum_{j \in M_i} S_{ij} + \sum_{j \in M_i} S_{ji}} \quad (4)$$

$$\mathcal{L}_{global} = \mathcal{L}_{mb}(f_T^5, f_I^5) + \mathcal{L}_{mb}(f_T^6, f_I^6) \quad (5)$$

where  $S_{ij}$  is the similarity between  $q_i$  and  $k_j$ , In a batch of image-text pairs,  $M_i$  denotes the set of index of the pairs that has the different ID with the  $i$ -th pair, so  $\sum_{j \in M_i} S_{ij}$  means the similarity of negative pairs from text to image, and  $\sum_{j \in M_i} S_{ji}$  means the similarity of negative pairs from image to text.  $\mathcal{L}_{mb}(q, k)$

is the masked bidirectional InfoNCE loss[10]. It masks  $k_j$  with the same ID as  $q_i$ , and  $q_j$  with the same ID as  $k_i$  because images and descriptions can have a huge difference from different views of the same vehicle. Bidirectional loss means we align both the image feature to the text feature and the text feature to the image feature simultaneously, which is more comprehensive and robust than unidirectional loss.  $(f_T^5, f_I^5)$  and  $(f_T^6, f_I^6)$  are the global features of the image and text mentioned above.

**Stage-2:** In stage-2, we fine-tune the model on the target domain(T2I-VeRi) by more fine-grained feature alignment to enhance its capability. We use multi-view multi-scale masked bidirectional contrastive loss[10]  $\mathcal{L}_{mvms}$  to train the model:

$$\mathcal{L}_{mvms} = \sum_{i=1}^3 \lambda_i \mathcal{L}_{mb}(f_I^i, f_T^i) + \sum_{i=4}^6 \mathcal{L}_{mb}(f_I^i, f_T^i) \quad (6)$$

We align all the global, local and multi-view feature pairs  $\{f_I^i, f_T^i\}_{i=1}^6$  via  $\mathcal{L}_{mvms}$  to learn a more comprehensive and detailed feature alignment.  $\lambda_1, \lambda_2, \lambda_3$ , are the self-adapting weights of the front view, side view and rear view, respectively.

## 4 Experiments

### 4.1 Experimental Setup

**Dataset:** We partition the dataset into training set and testing set by vehicle ID in the same way as VeRi-776, that is, given 576 IDs for training and the rest 200 IDs for testing. Specifically, T2I-VeRi has 1,826 pairs for training and 632 for testing. TIVRD2 has 20,460 pairs for training and 3,320 for testing.

**Evaluation Metrics:** We use the common metrics Rank-1, Rank-5, Rank-10 and mAP to evaluate the model. It is worth noting that in image-text retrieval tasks, Rank-k and Recall@k have the same meaning, that is, the proportion of the top-k retrieval results that contain the target[15]. AP reflects the area under the precision-recall curve. Refer to the highly-starred [repository](#) for detailed calculation. mAP is the mean of AP of all queries, which is more comprehensive.

**Implementation Details:** Input images are resized to 224x224. Random flipping and cropping are used for data augmentation. The maximum length of the text description is set to 60 for T2I-VeRi and 100 for TIVRD2 to ensure that at least 95% of the text descriptions are completely input into the model. The BERT model is fixed during training. We use the Adam optimizer with a learning rate of 0.002(0.001 in stage-2), warming up for 10(5) epochs, and decreasing to 80% at set epochs, 25(20) epochs in total. The batch size is 64. The temperature parameter  $\tau$  is set to 0.02. Our devices are NVIDIA TITAN RTX GPUs.

## 4.2 Comparison with State-of-the-Art Methods

Table 2 shows the comparison of our method with existing methods on T2I-VeRi.

Current research on TIVReid is limited(only one method[10] is proposed), so we compare not only with the SOTA method in TIVReid but also with some latest and representative methods in Text-Image Person Retrieval(TIPReid), Vision-Language Pre-training(VLP) and image-based object re-identification(ReID) tasks. LGUR[26], SSAN[11], HAT[3], TIPCB[6], TFAF[17] are the recent models for TIPReid. We re-train these models on T2I-VeRi and report the results. TransReID[14] is the representative model in object ReID and its performance is only marginally different from the SOTA method in both person and vehicle ReID tasks, so we choose it to compare with our method. However, this model can only be used for image-based retrieval tasks, so we add BERT as the text encoder and re-train it on T2I-VeRi. MCANet[10] is the current SOTA model for the TIVReid task, and we use it as the backbone of our method.

Our method outperforms the existing methods by a large margin: Rank-1, Rank-5, and mAP are **6.7%**, **7.6%**, **4.9%** higher than the current SOTA method, which indicates the effectiveness of the TIVRD2 and two-stage aligning strategy.

**Table 2.** Comparison with existing methods on T2I-VeRi.

Method	Ref	Rank-1	Rank-5	Rank-10	mAP
TransReID[14]	ICCV 21	7.5	24.3	35.1	7.4
LGUR[26]	ACM MM 22	11.3	29.0	45.9	9.2
SSAN[11]	arXiv 21	14.2	34.3	52.6	12.2
HAT[3]	ACM MM 23	16.0	40.7	56.2	13.7
TIPCB[6]	NeuroComputing 22	16.8	44.0	58.8	18.0
TFAF[17]	IEEE SPL 22	20.1	49.0	66.3	15.5
MCANet[10](baseline)	IEEE ITS 24	<u>26.1</u>	<u>57.1</u>	<u>72.3</u>	<u>19.6</u>
GFLA Stage-1(Ours)		19.3	43.0	56.2	16.4
GFLA Stage-2(Ours)		<b>32.8</b>	<b>64.7</b>	<b>79.1</b>	<b>24.5</b>

**Ablation Study: Data Scale.** Table 3 shows the performance of our method with different data scales: 5k, 10k, 15k, and 20.5k(all train data) of TIVRD2. As the data scale increases, the performance significantly improves in both stages.

**Ablation Study: Training Strategy.** Table 4 shows the performance on T2I-VeRi with different training strategies. Baseline method MCANet aligns global and local features in both two stages, which performs better in stage-1 but worse in stage-2 compared with our method GFLA, which relaxes the alignment in

**Table 3.** Performance of our method on T2I-VeRi with different training data scales.

Stage	Train Data	Rank-1	Rank-5	Rank-10	mAP
Stage-1	TIVRD2 5k	17.5	37.8	50.9	14.8
	TIVRD2 10k	17.6	38.6	55.1	14.8
	TIVRD2 15k	18.2	40.8	56.2	15.8
	TIVRD2 20.5k	<b>19.3</b>	<b>43.0</b>	<b>56.2</b>	<b>16.4</b>
baseline	T2I-VeRi	26.1	57.1	72.3	19.6
Stage-2 (based on Stage-1)	T2I-VeRi	27.1	58.0	75.5	20.7
		29.3	<u>64.0</u>	76.1	21.8
		<u>29.6</u>	60.3	<u>76.6</u>	<u>22.0</u>
		<b>32.8</b>	<b>64.7</b>	<b>79.1</b>	<b>24.5</b>

**Table 4.** Performance on T2I-VeRi with different training strategies.

Method	Stage	Train	Align	Loss	Rank-1	Rank-5	Rank-10	mAP	
MCANet-		T2I-VeRi	Global+Local	$\mathcal{L}_{mvms}$	6	26.1	57.1	72.3	19.6
MCANet 1	2	T2I-VeRi	Global+Local	$\mathcal{L}_{mvms}$	6	21.2	44.1	61.6	16.4
						+9.5	+20.1	+17.2	+6.0
		T2I-VeRi	Global+Local	$\mathcal{L}_{mvms}$	6	<u>30.7</u>	<u>64.2</u>	<u>78.8</u>	<u>22.4</u>
GFLA	1	TIVRD2	Global	$\mathcal{L}_{global}$	5	19.3	43.0	56.2	16.4
						+13.5	+21.7	+22.9	+8.1
	2	T2I-VeRi	Global+Local	$\mathcal{L}_{mvms}$	6	<b>32.8</b>	<b>64.7</b>	<b>79.1</b>	<b>24.5</b>

stage-1. The baseline method aligns the features fine-grained on the non-target domain(TIVRD2) in stage-1, which may lead to overfitting, and thus performs worse on the target domain(T2I-VeRi) in stage-2.

### 4.3 Ablation Study: Prompt design

In this section, we evaluate the effectiveness of the prompt design method.

The prompt word has a significant impact on the generation effects of MLLM. So we meticulously design four different prompts: 0-shot-v1(base version), 0-shot-v2(attribute-guided pre-prompt), 1-shot and 5-shot, shown in Table 5.

Based on these four prompts, we used GPT4-V to generate text descriptions for 2,458 vehicle images in T2I-VeRi respectively. To ensure the reliability of the results, all datasets are partitioned into training(1,826) and testing(632) sets in the same way as T2I-VeRi, and the model used is also the same, MCANet[10]. We cross-validated the trained models on T2I-VeRi and these generated datasets, yielding the following results and conclusions:

**Table 5.** Prompt comparison. Attribute **tips** and **options** are in **red** and **blue**.

Prompt name	Details
0-shot-v1	Give a paragraph of about 60 words to describe the vehicle in detail.
0-shot-v2	Give a paragraph of about 60 words to describe the vehicle in detail, including the <b>view</b> ( <b>front</b> , <b>left front</b> , <b>right front</b> , <b>left rear</b> , <b>right rear</b> , <b>rear</b> ), <b>color</b> , <b>vehicle type</b> ( <b>car</b> , <b>SUV</b> , <b>van</b> , <b>bus</b> , <b>pickup</b> , <b>truck</b> ), and any <b>distinguishing features on the front, side, or rear</b> of the vehicle.
1-shot	Prompt in 0-shot-v2 and the second pair in Fig 2.
5-shot	Prompt in 0-shot-v2 and five image-text pairs in Fig 2.

**Table 6.** Train on different prompt-guided generated datasets and test on T2I-VeRi.

Train	Attributes	Shots	Test	Rank-1	Rank-5	Rank-10	mAP
0-shot-v1	✗	✗	T2I-VeRi	9.2	25.2	39.7	8.5
0-shot-v2	✓	✗		10.6	25.9	34.2	9.2
1-shot	✓	1		<b>14.7</b>	<b>34.8</b>	<b>48.4</b>	<b>11.6</b>
5-shot	✓	5		<u>13.6</u>	<u>29.3</u>	<u>41.8</u>	<u>10.5</u>

Table 6 shows the test results of models trained on the generated dataset and tested on T2I-VeRi. Typically, the higher the quality of the generated dataset, the better the model is trained, and the higher the performance on T2I-VeRi.

Table 7 shows the test results of the model trained on T2I-VeRi and tested on the generated dataset. Typically, the higher the quality of the generated dataset, the better the performance of the model on it.

Therefore, the prompt with designed attributes and few optimized shot significantly improves the quality of the generated data and the performance of the model trained on them, which demonstrates the effectiveness of attribute-guided pre-prompt and optimized few-shot policy in guiding MLLM to generate high-quality descriptions.

**Table 7.** Train on T2I-VeRi and test on different prompt-guided generated datasets.

Train	Test	Attributes	Shots	Rank-1	Rank-5	Rank-10	mAP
T2I-VeRi	0-shot-v1	✗	✗	7.1	19.9	30.2	7.3
	0-shot-v2	✓	✗	9.7	26.4	38.4	9.7
	1-shot	✓	1	<u>12.0</u>	<u>32.4</u>	<u>43.4</u>	<u>10.9</u>
	5-shot	✓	5	<b>13.4</b>	<b>34.2</b>	<b>46.7</b>	<b>12.0</b>

#### 4.4 Dataset Evaluation

In this section, we conduct a detailed comparison of the T2I-VeRi and TIVRD2 datasets, to further verify the effectiveness of TIVRD2 generated by MLLM.

**Quantitative Analysis:** We conducted cross-validation using the current SOTA model MCANet[10] on TIVRD2 and T2I-VeRi in Table 8. The first two rows show that when models are tested on their respective source datasets, TIVRD2 performs better, indicating better consistency of the TIVRD2 dataset; the lower mAP is due to the larger number of TIVRD2 test datasets. The last two rows show that when models are tested on different source datasets, TIVRD2 performs better, indicating that the model trained on the TIVRD2 dataset has better generalization performance. Comparing rows 1 and 4, even without using T2I-VeRi, the model’s performance on the T2I-VeRi test set is not far off from the existing SOTA model trained on T2I-VeRi, which fully demonstrates the effectiveness of the TIVRD2 dataset.

**Table 8.** cross-validation on TIVRD2 and T2I-VeRi dataset.

Train	Test	Rank-1	Rank-5	Rank-10	mAP
T2I-VeRi		26.1	57.1	72.3	<b>19.6</b>
TIVRD2		<b>33.6</b>	<b>61.6</b>	<b>74.2</b>	14.5
T2I-VeRi	TIVRD2	12.2	28.7	39.3	7.9
TIVRD2	T2I-VeRi	<b>21.2</b>	<b>44.1</b>	<b>61.6</b>	<b>16.4</b>

**Qualitative Analysis:** We randomly selected 100 image-text pairs from the T2I-VeRi and TIVRD2 datasets and manually counted the number of noisy pairs (inappropriate, incorrect, or insufficient annotations). Results show that the proportion of noisy pairs in T2I-VeRi is about 3%, while that in TIVRD2 is less than 1%. We present some pairs from T2I-VeRi and TIVRD2 for comparison in Fig 4 and the matching result comparison in 5.

Overall, through quantitative and qualitative analysis, we demonstrate that the quality of the dataset TIVRD2, generated by guiding GPT4-V with proposed attribute-guided pre-prompt and optimized few-shot policy, is not only comparable to but even exceeds the quality of manually annotated datasets.

**GPT4-V:** Viewed from the **left side**, this vehicle is an **orange car**[**more appropriate color**] with a compact body and **hatchback design**, featuring a **clear headlight** on the left and a side **mirror** that’s likely **black**. It has a **noticeable roofline** that slopes down towards the rear of the car, which suggests a fairly **aerodynamic shape**. [**more features**]

**Manual:** This is a **brown car**[**inappropriate color**] with a paper-pumping **white object** on the front of the cockpit and **black trim** on the side of the body. [**less features**]

**GPT4-V:** Viewed from the **right front**, this vehicle is a **small white truck**[**correct type**] with a **blue tarp** covering over the cargo bed. The cab features a **blue sticker** with **three white letters** on top and the word ‘**JAC**’ on the grille. Along the bottom left of its body, the truck has decorative **blue and red stripes**, and there are **two occupants** visible in the cab. [**more**]

**Manual:** This is a white **JIANGhuai van**[**ambiguous type**], the **driver** wearing a **white shirt**, the carriage covered with **dark blue cloth**, the bottom of the carriage has a **light blue paint** decorative strip. [**less**]



**Fig. 4.** Examples from TIVRD2 and T2I-VeRi. We use red to mark the noisy annotations in T2I-VeRi and corresponding better annotations in TIVRD2, use ‘[ ]’ to explain the reasons, and use blue to mark the distinctive features in descriptions.

**Manual:** This is a **black SUV** with 4 **roof stiffeners** on the roof and a non-fitting **luggage rack**. There are 4 horizontally arranged **air intakes** in the front of the car.

**GPT4-V:** Viewed from the **front**, this vehicle is a **dark-colored SUV** with a reflective window shield and **silver frontal grille** featuring **horizontal slats**. It has a pair of visible **roof racks** on top, suggesting utility and additional cargo capacity. The **headlights** appear to be a clear lens, **rectangular style**, typical of **older SUV designs**, while the hood shows a slight protrusion at the center, giving it a more robust appearance. The car’s **overall stance is bulky and squared**, which contributes to its sturdy and durable look.



**Top-5 matching results**

Trained on **T2I-VeRi:**



Trained on **TIVRD2:**



( Red: target vehicle, Green: same ID )

**Fig. 5.** Results comparison of models trained on T2I-VeRi and TIVRD2. The following description example shows that the manual description is insufficient to retrieve the target vehicle since there are many vehicles matching these features(marked in blue), while the generated description contains more representative and discriminative features (marked in red). The matching results show the limitation of the manual annotation and the effectiveness of the generated description.

## 5 Conclusion

In this paper, we have explored an effective method to distill the knowledge of MLLM into the TIVReid model. We propose a useful prompt design approach, which introduces the attribute-guided pre-prompt and optimized few-shot policy to guide MLLM to generate high-quality descriptions for vehicle images, and thus construct a large-scale high-quality vehicle image-text pairs dataset TIVRD2. Then, We devise an effective two-stage aligning strategy to better utilize the generated data. systematic and extensive experiments show the effectiveness of our prompt design approach and the two-stage aligning strategy, and demonstrate that the quality of the TIVRD2 dataset is of equal or even better quality than manually annotated datasets.

However, there are still some limitations in our work. We only consider the representative MLLM GPT4-V to test our ideas. Future research can test some open-source MLLMs to further verify the effectiveness of our method and compare their performance, such as LLaVA[18], InternVL[7], Qwen-VL[2], etc. Besides, we only focus on the text-to-image vehicle re-identification task, The similar method can be applied to text-to-image person re-identification or other fine-grained text-image retrieval tasks. What’s more, 2D visualization of learnt space can be used to analyze the feature alignment effect. We wish that our work and limitations could inspire more research on the relevant fields.

**Acknowledgement.** This work is supported in part by the Beijing Municipal Natural Science Foundation under Grant QY23186 and the Natural Science Foundation of China under Grant 62072457. Meanwhile, thanks are given to Leqi Ding for valuable discussions.

## References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint [arXiv:2303.08774](https://arxiv.org/abs/2303.08774) (2023)
2. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. arXiv preprint [arXiv:2308.12966](https://arxiv.org/abs/2308.12966) (2023)
3. Bin, Y., Li, H., Xu, Y., Xu, X., Yang, Y., Shen, H.T.: Unifying two-stream encoders with transformers for cross-modal retrieval. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 3041–3050 (2023)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *Adv. Neural. Inf. Process. Syst.* **33**, 1877–1901 (2020)
5. Chen, L., Li, J., Dong, X., Zhang, P., He, C., Wang, J., Zhao, F., Lin, D.: Sharegpt4v: Improving large multi-modal models with better captions. arXiv preprint [arXiv:2311.12793](https://arxiv.org/abs/2311.12793) (2023)
6. Chen, Y., Zhang, G., Lu, Y., Wang, Z., Zheng, Y.: Tipcb: A simple but effective part-based convolutional baseline for text-based person search. *Neurocomputing* **494**, 171–181 (2022)



7. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24185–24198 (2024)
8. Cui, C., Ma, Y., Cao, X., Ye, W., Zhou, Y., Liang, K., Chen, J., Lu, J., Yang, Z., Liao, K.D., et al.: A survey on multimodal large language models for autonomous driving. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 958–979 (2024)
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](https://arxiv.org/abs/1810.04805) (2018)
10. Ding, L., Liu, L., Huang, Y., Li, C., Zhang, C., Wang, W., Wang, L.: Text-to-image vehicle re-identification: Multi-scale multi-view cross-modal alignment network and a unified benchmark. *IEEE Transactions on Intelligent Transportation Systems* (2024)
11. Ding, Z., Ding, C., Shao, Z., Tao, D.: Semantically self-aligned network for text-to-image part-aware person re-identification. arXiv preprint [arXiv:2107.12666](https://arxiv.org/abs/2107.12666) (2021)
12. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., Wu, Y., Ji, R.: Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint [arXiv:2306.13394](https://arxiv.org/abs/2306.13394) (2023)
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, S., Luo, H., Wang, P., Wang, F., Li, H., Jiang, W.: Transreid: Transformer-based object re-identification. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 15013–15022 (2021)
15. Hodosh, M., Young, P., Hockenmaier, J.: Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* **47**, 853–899 (2013)
16. Jia, S., Lyu, R., Zhao, K., Chen, Y., Yan, Z., Ju, Y., Hu, C., Li, X., Wu, B., Lyu, S.: Can chatgpt detect deepfakes? a study of using multimodal large language models for media forensics. arXiv preprint [arXiv:2403.14077](https://arxiv.org/abs/2403.14077) (2024)
17. Li, S., Lu, A., Huang, Y., Li, C., Wang, L.: Joint token and feature alignment framework for text-based person search. *IEEE Signal Process. Lett.* **29**, 2238–2242 (2022)
18. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint [arXiv:2310.03744](https://arxiv.org/abs/2310.03744) (2023)
19. Liu, H., Tian, Y., Yang, Y., Pang, L., Huang, T.: Deep relative distance learning: Tell the difference between similar vehicles. In: Proceedings of Conference on Computer Vision and Pattern Recognition. pp. 2167–2175 (2016)
20. Liu, X., Liu, W., Mei, T., Ma, H.: Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Trans. Multimedia* **20**(3), 645–658 (2018). <https://doi.org/10.1109/TMM.2017.2751966>
21. Lou, Y., Bai, Y., Liu, J., Wang, S., Duan, L.: Veri-wild: A large dataset and a new method for vehicle re-identification in the wild. In: Proceedings of Conference on Computer Vision and Pattern Recognition. pp. 3235–3243 (2019)
22. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4004–4012 (2016)
23. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint [arXiv:1807.03748](https://arxiv.org/abs/1807.03748) (2018)

24. OpenAI: Gpt-4v(ision) system card (2023), <https://openai.com/research/gpt-4v-system-card>
25. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018), <https://www.mikecaptain.com/resources/pdf/GPT-1.pdf>
26. Shao, Z., Zhang, X., Fang, M., Lin, Z., Wang, J., Ding, C.: Learning granularity-unified representations for text-to-image person re-identification. In: Proceedings of the 30th acm international conference on multimedia. pp. 5566–5574 (2022)
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
28. Wang, W., Mrini, K., Yang, L., Kumar, S., Tian, Y., Yan, X., Wang, H.: Fine-tuned multimodal language models are high-quality image-text data filters. arXiv preprint [arXiv:2403.02677](https://arxiv.org/abs/2403.02677) (2024)
29. Wen, L., Yang, X., Fu, D., Wang, X., Cai, P., Li, X., Ma, T., Li, Y., Xu, L., Shang, D., et al.: On the road with gpt-4v (ision): Early explorations of visual-language model on autonomous driving. arXiv preprint [arXiv:2311.05332](https://arxiv.org/abs/2311.05332) (2023)
30. Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., Chen, E.: A survey on multimodal large language models. arXiv preprint [arXiv:2306.13549](https://arxiv.org/abs/2306.13549) (2024)
31. Zheng, T., Milind, N., Ming-Yu, L., Xiaodong, Y., Stan, B., Shuo, W., Ratnesh, K., David, A., Jenq-Neng, H.: Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In: CVPR (2019)
32. Zheng, Z., Jiang, M., Wang, Z., Wang, J., Bai, Z., Zhang, X., Yu, X., Tan, X., Yang, Y., Wen, S., et al.: Going beyond real data: A robust visual representation for vehicle re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 598–599 (2020)
33. Zheng, Z., Ruan, T., Wei, Y., Yang, Y., Mei, T.: Vehiclenet: Learning robust visual representation for vehicle re-identification. *IEEE Trans. Multimedia* **23**, 2683–2693 (2020)



# Audio-Visual Navigation with Anti-Backtracking

Zhengkao Zhao<sup>1</sup>(✉), Hao Tang<sup>2</sup>, and Yan Yan<sup>1</sup>

<sup>1</sup> Department of Computer Science, University of Illinois Chicago, Chicago, USA  
zzhao48@hawk.iit.edu

<sup>2</sup> Robotics Institute, Carnegie Mellon University, Pittsburgh, USA

**Abstract.** Embodied navigation, which involves robotic agents exploring an unknown environment to reach target locations with egocentric observation, is a complex problem in the field of embodied AI. Audio-visual navigation extends this concept by equipping agents with both visual and auditory sensors. Recent studies have explored the audio-visual navigation task, exploring its potential and intricacies. Current methodologies, despite their achievements, often fail to fully utilize the capabilities of these sensory modalities, leading to sub-optimal designs and inefficiencies. In addition, the prevalent backtracking problem in navigation tasks often leads to redundant movements by robots. To address these challenges, we introduce the Audio-Visual Guided Navigation (AVGN) model. At its core, AVGN champions the fusion of visual and auditory data through sophisticated modality fusion layers. Our model represents a significant advancement by employing transformers for visual encoding and deploying a novel Dual Stage Feature Integration (DSFI) to decode latent interrelationships between the visual and audio realms. With AVGN, the backtracking issue is mitigated, as the acoustic map synergizes more effectively with visual features for informed action decisions. Furthermore, targeting the backtracking problem, we propose a unique set of reward structures to guide and refine the actions of the agent. Benchmark evaluations on Replica and Matterport3D datasets validate our claims, and AVGN notably surpasses existing methodologies in audio-visual navigation tasks.

**Keywords:** Visual Navigation · Reinforcement Learning · Audio Visual Navigation.

## 1 Introduction

Embodied navigation [1–5] is a critical area of research in the field of embodied intelligence [6–9], where robotic agents navigate unknown environments to reach their target destinations using egocentric observations. In addition to visual observations, hearing is an essential sense as it provides both temporal and spatial information, allowing visually impaired subjects to navigate efficiently. To

get a better understanding of physical space and localizing sound-emitting targets, Chen et al. [10] introduced the concept and standard of the audio-visual navigation task. Concurrently, research [10–15] have been proposed for the audio-visual navigation task. Among recent audio-visual navigation research, LLA [12] presents a traditional phase-based navigation approach, leveraging a topological graph to streamline path planning. SoundSpaces [10] distinguishes itself as a trailblazing end-to-end solution, uniquely foregoing reliance on topological graphs and specific auditory meta-data such as sound source categories (e.g., telephone, doorbell, alarm). AV-WaN [15] utilizes waypoints (depicted via a topological graph) to enhance long-distance navigation efficacy. SAVi [14] addresses environments with sporadic sound emissions by incorporating auditory meta-data.

However, a significant limitation of these methodologies lies in their approach to fusing visual and auditory data. Their strategies for modalities integration often use concatenation. This simplistic merging often loses the details that each type of sensor can provide, potentially missing opportunities for deeper insights and enhanced navigation cues. Furthermore, existing methods suffer from the pervasive backtracking problem. This challenge manifests itself in multiple ways: it not only introduces redundancy trajectories, inflating the navigation time, but it can also lead to agents being sent astray. Such misdirection can be particularly detrimental in complex or dynamic environments where efficient navigation is paramount. Agents often find themselves retracing their steps or oscillating between points, increasing the risk of navigational failure.

To enhance the effectiveness of information utilization and reduce the backtracking issue, our proposed methodology aims to effectively integrate both audio and visual cues and examine their interrelationships. We have designed a model that comprises two dedicated encoders, tailored for visual and audio inputs, complemented by an advanced suite of fusion techniques on different levels of features. In contrast to conventional Convolutional Neural Networks (CNNs), vision transformers [16–18] demonstrate superior capability in capturing intricate spatial relationships and global context from visual data. Leveraging the prowess of transformers, our model becomes adept at extracting and assimilating pivotal features from visual streams. To further accentuate this synergy, we introduce a multi-stage fusion strategy termed Dual Stage Feature Integration (DSFI). This approach allows our model to discern and represent the nuanced connections between visual scenes and concurrent audio cues. The DSFI procedure is bifurcated into two fusion strata. The Early Fusion integrates early visual and audio features, offering a foundational context and immediate sensory correlations. On the other hand, the Acoustic Fusion branch refined visual and auditory features in a more context-aware manner. By introducing DSFI into our architecture, our model can seamlessly blend insights from both sensory channels, providing a holistic and enriched understanding for the navigating agent. Since the acoustic map contains information about the direction of the sound source, which is also the navigation target, the backtracking issue is alleviated. Moreover, in addressing the common backtracking issue inherent in many

navigation algorithms, we introduce the directional continuity reward and the historical trajectory reward to constrain the agent’s actions. The directional continuity reward penalizes abrupt changes in the agent’s direction of movement. The historical trajectory reward imposes penalties if the agent revisits a location to which it has already been on its trajectory.

In conclusion, our contributions can be summarized as three-fold: **(i)** We introduced a novel audio-visual navigation method, AVGN, which employs a robust fusion strategy, DSFI, adeptly integrating visual and audio features to optimize action decisions; **(ii)** We have innovatively devised a triad of reward mechanisms, specifically targeting and mitigating the persistent backtracking challenges inherent to navigation tasks; **(iii)** Benchmarking on the SoundSpaces platform, our model has demonstrated superior performance over existing methods on Matterport3D and Replica datasets.

## 2 Related Work

### 2.1 Vision-Based Navigation

The significance of vision in cognitive mapping for human navigation has been extensively investigated in early research [19,20]. Similarly, recent AI agents process egocentric visual input for navigation purposes [8,19,21–26]. NTS [21] presents a method for image-goal navigation in uncharted environments by leveraging topological space representations that intricately merge semantics with proximate geometry, hinged on nodes and their relationships. Meanwhile, LB-WayPtNav [19] offers an inventive blend of model-based control and learning-oriented perception for robot navigation, delivering consistent and efficient trajectories in uncharted terrains. Bansal et al. [22] introduces a technique that synergizes model-based control with learning-driven perception to navigate robots, resulting in reliable and efficient trajectories in unfamiliar environments, even with minimal 3D environmental mapping and lower frame rates, showcasing effective sim-to-real generalization.

Vision-based navigation is frequently augmented with textual input in the form of dialogs. Pioneering studies [27–29] have investigated the feasibility of vision-dialog navigation. Specifically, Jesse et al. [27] curate a dataset tailored for Cooperative Vision-and-Dialog Navigation set within photorealistic residential environments, and introduce a task where agents rely on human dialog history for navigation. Their insights reveal that a richer dialog history boosts navigation outcomes. VLN-BERT [28] refines the BERT [30] architecture by incorporating a time-sensitive recurrent function for the vision-and-language navigation challenge, showcasing adaptability over multiple transformer models and adeptness in managing both navigation and expression referral tasks. Furthermore, Hao et al. [29] propose a pre-training and fine-tuning strategy for vision-language navigation tasks, which leverages a plethora of image-text-action datasets in a self-supervised manner to augment performance in established vision-language navigation setups.

Although vision-based and vision-language navigation has achieved considerable success, audiovisual navigation is emerging as a novel challenge in the domain.

## 2.2 Audio-Visual Navigation

Audio-Visual Navigation pertains to tasks where agents process visual and auditory cues. Historically, most audio-visual navigation research has been confined to single-sound environments. Notable examples include LLA (Look, Listen, and Act) [12], SoundSpaces [10], AV-WaN (Audio-Visual Waypoint Navigation) [15], and SAVi (Semantic Audio-Visual Navigation) [14]. LLA [12] offers a conventional phase-based navigation method, relying on a topological graph to facilitate optimal path planning. SoundSpaces [10], a pioneering end-to-end method, stands out by not depending on any topological graph or specific auditory meta-information, such as the category of the sound source (e.g., telephone, doorbell, alarm). AV-WaN [15] adopts waypoints (represented through a topological graph) to optimize navigation over longer distances. SAVi [14] considers scenarios with intermittent sound emissions, enriching their model with auditory meta-information. For more acoustically intricate settings, where the target sound blends with multiple ambient noises, SAAVN (Sound Adversarial Audio-Visual Navigation) [13] offers an end-to-end framework.

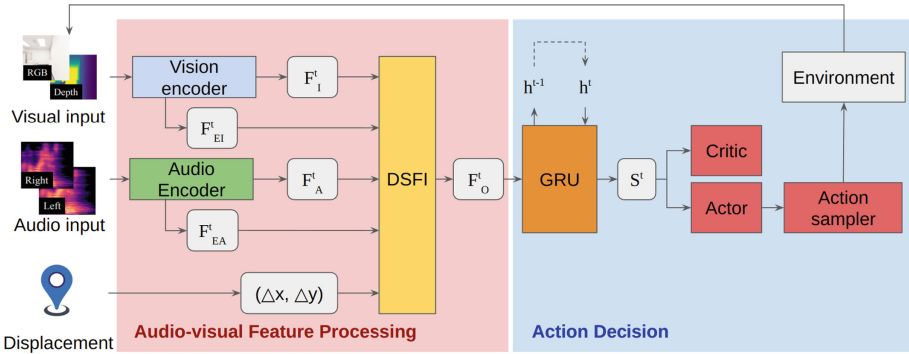
Although significant strides have been made in audiovisual navigation, the audio and visual modalities are recognized as underutilized, a gap that we aim to bridge in our proposed methods.

## 3 The Proposed Method

### 3.1 Method Overview

We frame the navigation challenge as a reinforcement learning problem, where a robot, driven by an audio cue, learns an optimal policy to swiftly navigate to a target within an unfamiliar environment. We focus on two tasks: AudioGoal Navigation and AudioPointGoal Navigation. In the AudioGoal scenario, the agent receives both audio and visual inputs. AudioPointGoal is an audio extension of the PointGoal task [31–33], where the agent navigates with an egocentric view and audio input, and its displacement to the target. The action space of the agent comprises {MoveForward, TurnLeft, TurnRight, Stop}. Following [10], the sensory inputs include binaural sound simulated by room impulse response (RIR) [34], egocentric RGB images, and the displacement vector to the goal (exclusive in the AudioPointGoal setting). We introduce our solution as AVGN, which stands for Audio-Visual Guided Navigation. The architecture of AVGN consists of four key components, as illustrated in Fig. 1. Specifically, AVGN processes given egocentric vision and audio inputs by encoding them into distinct features. Visual data is encoded using transformers, while audio data is processed with convolutional neural networks (CNNs). These separate features

are then fused through the Dual Stage Feature Integration (DSFI), resulting in a comprehensive audio-visual embedding. This fused embedding is subsequently transformed into a temporally-sensitive state representation by a Gated Recurrent Unit (GRU). Finally, an actor-critic framework is applied to predict, evaluate, and refine subsequent actions. The robot agent iteratively undergoes this sequence until it successfully locates the target. Each of these components will be elaborated upon in the subsequent sections.



**Fig. 1. The AVGN structure.** The visual and audio inputs are initially processed by their respective encoders. Each type of input is encoded to generate the corresponding features, which include both primary and early-stage characteristics. These features, along with a displacement vector, are then input into the Dual Stage Feature Integration (DSFI) to derive an observation feature. Subsequently, this observation feature is fed into a Gated Recurrent Unit (GRU) to capture the state feature at step  $t$ . Finally, this state feature is used by an actor-critic algorithm to determine the appropriate action.

### 3.2 Audio-Visual Model

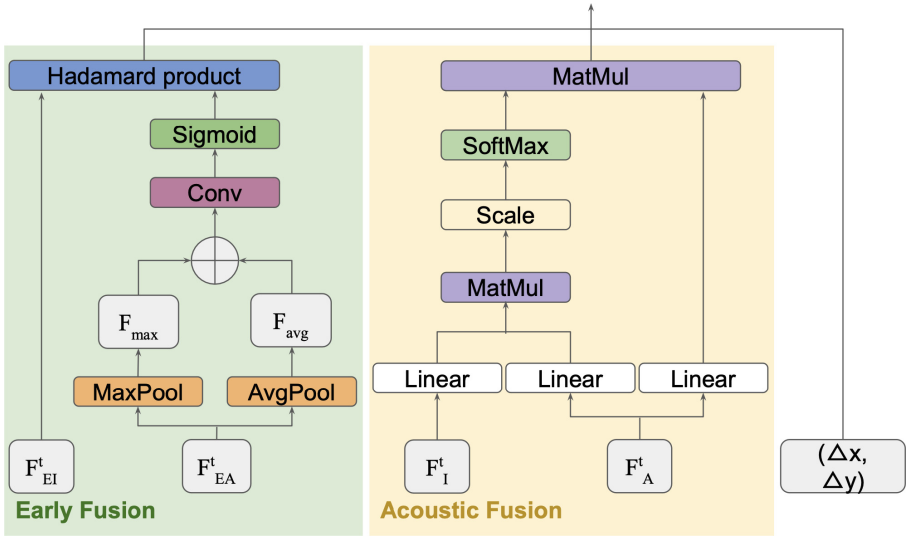
At  $t$ -th step, the agent receives a visual input, denoted as  $I^t$ , and an audio input, represented by  $A^t$ . The visual input  $I^t$  undergoes processing via a Swin transformer [17], resulting in the derivation of an early image feature,  $\mathbf{F}_{EI}^t$ , and an image feature,  $\mathbf{F}_I^t$ . Similarly, the audio input  $A^t$  is processed through a Convolutional Neural Network (CNN) to obtain an early audio feature,  $\mathbf{F}_{EA}^t$ , and an audio feature,  $\mathbf{F}_A^t$ . Additionally, the GPS sensor of the agent collects the displacement input,  $\Delta^t$ .

To gauge the real-time contribution of each modality, factoring in the fluctuating contexts, we have devised a trainable audio-visual fusion mechanism, termed Dual Stage Feature Integration (DSFI). This mechanism is adept at converting the encoded features into a consolidated embedding vector,  $\mathbf{F}_O^t$ :

$$\mathbf{F}_O^t = DSFI(\mathbf{F}_I^t, \mathbf{F}_A^t, \mathbf{F}_{EI}^t, \mathbf{F}_{EA}^t, \Delta^t) \quad (1)$$

The detailed structure of DSFI is illustrated in Fig. 2. The early fusion branch is shown on the left, and we pay special attention to the early visual feature  $\mathbf{F}_{EI}^t$  with the early audio feature  $\mathbf{F}_{EA}^t$  as follows. For the early audio feature  $\mathbf{F}_{EA}^t$ , we apply the max pooling and average pooling along the channels. We obtain max pool  $\mathbf{F}_{max} \in \mathbb{R}^{1 \times W \times H}$  and the average pool  $\mathbf{F}_{avg} \in \mathbb{R}^{1 \times W \times H}$ , respectively. Then we concatenate the obtained features and process a  $1 \times 1$  convolutional layer followed by a sigmoid function, to obtain a weighted early audio feature. Then we proceed with a Hadamard product between the weighted early audio feature and the early visual feature to obtain an early fusion feature. The procedure above can be concluded as follows:

$$\mathbf{F}_E^t = \text{sigmoid}(\text{Conv}(\mathbf{F}_{max} \oplus \mathbf{F}_{avg})) \odot \mathbf{F}_{EI}^t \quad (2)$$



**Fig. 2. The structure of DSFI.** The early audio feature  $\mathbf{F}_{EA}^t$  is first enhanced with a spatial attention mechanism and then combined with the early vision features  $\mathbf{F}_{EI}^t$  using the Hadamard product. For the image and audio features, we apply cross-attention to process their interaction. The resulting output is then concatenated with the early fused features and the displacement vector to form the observation feature.

The other branch is the acoustic fusion branch, which we process with visual feature  $\mathbf{F}_I^t$  and audio feature  $\mathbf{F}_A^t$ . We apply the function as follows:

$$\mathbf{F}^t = \text{softmax}\left(\frac{\mathbf{W}_1 \mathbf{F}_I^t (\mathbf{W}_2 \mathbf{F}_A^t)^T}{\sqrt{d}}\right) \mathbf{W}_3 \mathbf{F}_A^t \quad (3)$$

where  $\mathbf{W}_1$  and  $\mathbf{W}_2$  are both weight matrices to be optimized;  $d$  is a scalar factor with a value of 256; and  $F^t$  is the resulting fused embedding vector. We use the



right branch to extract useful information from the audio feature  $\mathbf{F}_A^t$  and fuse with the visual feature  $\mathbf{F}_I^t$ . In this way, the audio information, which delivers the sound source orientation, is combined with the visual feature and the agent can make better decisions accordingly.

The primary advantage of the DSFI design lies in both the acoustic fusion branch and the early fusion branch. Since the acoustic reveals where the sound comes from, it is important to transfer the audio information into visual features. By fusing the visual feature  $\mathbf{F}_I^t$  and the audio feature  $\mathbf{F}_A^t$ , it ensures a robust correlation between the two, efficiently extracting valuable information from the audio data and enhancing the visual feature in the process. Importantly, the fusion of early visual and audio features delves into finer details of the environment, capturing nuanced environmental cues. Such multi-level feature fusion greatly aids in making more informed decisions regarding actions.

With the fused early feature  $\mathbf{F}_E^t$ , the fused feature  $\mathbf{F}^t$ , and the displacement vector  $\Delta$ , the DSFI concatenates them together to obtain the feature of current observation at step  $t$ :

$$\mathbf{F}_O^t = \mathbf{F}_E^t \oplus \mathbf{F}^t \oplus \Delta. \quad (4)$$

We employ a bidirectional GRU [35] with a 512-dimensional hidden layer to enhance a series of observation features, namely  $\mathbf{F}_O^{t_0}$  through  $\mathbf{F}_O^{t_n}$ , culminating in a temporally-aware state representation denoted by  $s^t$ . Specifically, at time  $t$ , the GRU cell takes in both the current embedding  $\mathbf{F}_O^t$  and the previous cell state  $h^{t-1}$  to produce  $s^t$  and  $h^t$ , which can be formulated as

$$s^t, h^t = GRU(\mathbf{F}_O^{t-1}, h^{t-1}). \quad (5)$$

The state vectors, denoted as  $s^1, \dots, s^t$ , are processed through an actor-critic network for two primary purposes: firstly, to predict the conditioned action probability distribution  $\pi_{\theta_1}(a^t|s^t)$  and secondly, to estimate the state value  $V_{\theta_2}(s^t)$ . The actor and the critic are implemented with a single linear layer parameterized by  $\theta_1$  and  $\theta_2$ , respectively. The action sampler shown in Fig.1 samples the actual action denoted  $a^t$ , based on  $\pi_{\theta_1}(a^t|s^t)$ . The training aims to maximize the expected discounted return  $R$ :

$$R = \mathbb{E}_{\pi}[\sum_{t=1}^T \Gamma^t r(s^{t-1}, a^t)], \quad (6)$$

where  $\Gamma$  represents the discount factor, while  $T$  is the maximum number of time steps. Additionally,  $\pi$  denotes the policy that governs the robot agent.  $r(s^{t-1}, a^t)$  is the reward given by the environment at the time step  $t$ . The reward is detailed in Section 3.3. Proximal Policy Optimization (PPO) [36] is adopted in this work to optimize Equation (6). The entire procedure is described in Algorithm 1.

---

**Algorithm 1** Audio Visual Guided Navigation.

---

**Require:** Environment  $E$ , stochastic policies  $\pi$ , initial actor-critic weights  $\theta_0$ , initial encoder and DSFI weights  $\mathbf{W}_0$ , the number of updates  $M$ , the number of episodes  $N$ , max time steps  $T$ .

**Ensure:** Trained weights:  $\theta_M$  and  $\mathbf{W}_M$

- 1: **for**  $i = 1$  to  $M$  **do**
  - 2:   # Run policy  $\pi_{\theta_{i-1}}$  in environment for  $N$  episodes and  $T$  timesteps
  - 3:    $\{(o^t, h^{t-1}, a^t, r^t)\}_{t=1}^T \leftarrow \text{roll}(E, \pi_{\theta}, T)$  for  $i$ -th update
  - 4:   Compute advantage estimates
  - 5:   # Optimize w.r.t.  $\theta$  and  $\mathbf{W}$
  - 6:    $(\theta_i, \mathbf{W}_i) \leftarrow$  update the weights using the PPO algorithm to maximize Equation (6)
- 

### 3.3 Reward and Training

To address the backtracking issue discussed in Section 1, besides integrating the audio feature and the visual feature to guide the agent, as mentioned in Section 3.2, we implement a combination of fundamental rewards that facilitate navigation. These include rewards for stopping at the designated goal, time-related penalties, rewards for decreasing geodesic distance, and penalties for increasing geodesic distance. Additionally, we introduce specialized rewards, termed the directional continuity reward and historical trajectory reward.

**Directional continuity reward** penalizes the agent for abrupt changes in its direction of movement. It encourages the agent to move smoothly and continuously. Let  $\theta_{prev}$  be the agent’s previous movement direction and  $\theta_{curr}$  be its current movement direction. Then the directional continuity reward  $r_{dcr}$  can be defined as:

$$r_{dcr} = -\beta \times |\theta_{curr} - \theta_{prev}| \tag{7}$$

where  $\beta$  is a constant that determines the penalty’s strength for changing direction. Backtracking often requires an agent to make a U-turn or change its direction drastically. By penalizing sharp directional changes, the agent is less likely to make such U-turns, thus reducing backtracking.

**Historical trajectory reward** is based on storing the recent positions the agent has visited. If the agent revisits a location that has been in the recent past, it receives a penalty. Let  $H_i^t$  be the set of positions visited by the agent in the past  $i$  timesteps, and  $p_{curr}$  be the agent’s current position. Then the historical trajectory reward  $r_{htr}$  at time  $t$  can be defined as:

$$r_{htr} = \begin{cases} -\alpha, & \text{if } p_{curr} \in H_i^t, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where  $\alpha$  is a positive constant representing the penalty for revisiting a recent position. By penalizing the agent for revisiting recent locations, the agent is

discouraged from taking routes that lead it back to where it has just come from, thus reducing the chances of backtracking.

In our experimental framework, we amalgamate the aforementioned rewards with three straightforward incentives, which follow SoundSpaces [10]: (1) a reward of +10 points when the robot successfully navigates to the target and initiates the "Stop" action; (2) a reward of +0.25 points whenever the Manhattan distance between the robot and the target decreases; and (3) a time penalty of -0.01 for each action taken, promoting more efficient navigation.

## 4 Experiments

### 4.1 Dataset and Benchmark

In this study, we employ our method in conjunction with the 3D environment benchmark, SoundSpaces [10, 11]. SoundSpaces stands out as an advanced and lifelike acoustic simulation platform tailored for audio-visual embodied AI research. This platform is built based on Habitat [37, 38], offering a wide spectrum of research possibilities, ranging from audio-visual navigation and exploration to echolocation, as well as audio-visual floor plan reconstruction. This broad array of capabilities enables researchers to delve deeper into the nuances of embodied vision in various dimensions. For our experiments and analysis, we have specifically chosen datasets from Matterport3D [39] and Replica [40]. We use the AudioGoal setting for our experiments. We mainly use the telephone as a sound source for the experiments.

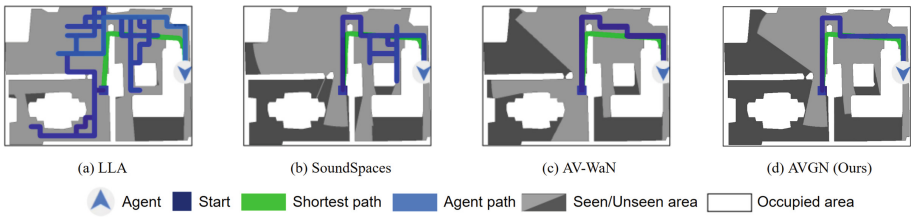
**Table 1. Quantitative comparisons with navigation methods.** We compare our model with existing methods on the Replica dataset and the Matterport3D dataset under both Heard and Unheard settings. SPL, SR, and SNA denote the success rate, success weighted by path length, and success weighted by number of actions, respectively.

Method	Replica						Matterport3D					
	Heard			Unheard			Heard			Unheard		
	SPL	SR	SNA	SPL	SR	SNA	SPL	SR	SNA	SPL	SR	SNA
RandomAgent	4.9	18.5	1.8	4.9	18.5	1.8	2.1	9.1	0.8	2.1	9.1	0.8
Direction Follower	54.7	72.0	41.1	11.1	17.2	8.4	32.3	41.2	23.8	13.9	18.0	10.7
Frontier Waypoints	44.0	63.9	35.2	6.5	14.8	5.1	30.6	42.8	22.2	10.9	16.4	8.1
Supervised Waypoints	59.1	88.1	48.5	14.1	43.1	10.1	21.0	36.2	16.2	4.1	8.8	2.9
LLA [12]	57.6	83.1	47.9	7.5	15.7	5.7	22.8	37.9	17.1	5.0	10.2	3.6
SoundSpaces [10]	78.2	94.5	52.7	<u>34.7</u>	50.9	16.7	55.1	71.3	32.6	25.9	40.1	12.8
AV-WaN [15]	<u>86.6</u>	<b>98.7</b>	<u>70.7</u>	34.7	<u>52.8</u>	<u>27.1</u>	<u>72.3</u>	<b>93.6</b>	<u>54.8</u>	<u>40.9</u>	<u>56.7</u>	<u>30.6</u>
AVGN (Ours)	<b>88.5</b>	<u>97.3</u>	<b>71.9</b>	<b>34.9</b>	<b>59.1</b>	<b>29.0</b>	<b>84.3</b>	<u>88.3</u>	<b>57.1</b>	<b>53.2</b>	<b>62.2</b>	<b>38.6</b>

For evaluation, we apply three metrics in our experiments: 1) success rate (SR), the fraction of successful episodes, i.e., episodes in which the agent stops exactly at the audio goal location on the grid; 2) success weighted by path length (SPL) [41], the standard metric that weighs successes by their adherence to the shortest path; 3) success weighted by number of actions (SNA), which penalizes rotation in place actions, which do not lead to path changes.

## 4.2 Implementation Details

In the implementation of the AVGN model, raw egocentric vision images of size  $256 \times 256 \times 3$  are processed through 6 transformer blocks, each with multi-head self-attention mechanisms, producing a visual feature map of  $64 \times 64 \times 128$ . Currently, the audio data, represented as spectrograms with dimensions  $64 \times 128$ , undergoes processing through a CNN composed of three convolutional layers (each followed by batch normalization and ReLU activation), resulting in a feature map of  $8 \times 16 \times 64$ . These visual and audio features are fused using the Dual Stage Feature Integration (DSFI) mechanism, yielding a combined embedding of dimension 128. This embedding undergoes temporal refinement in a bidirectional GRU with a hidden state of 512, feeding into the actor-critic framework. The actor and critic networks, each built from two dense layers with 256 neurons and ReLU activations, optimize actions using the Proximal Policy Optimization (PPO) algorithm [36] at a learning rate of  $1 \times 10^{-4}$ .



**Fig. 3. Comparison with state-of-the-art methods.** Our results showcase smooth navigation without backtracking problems.

## 4.3 Comparison with Baselines

The experiment results are shown in Table 1. We compare our methods with baselines on two datasets: Replica and Matterport3D. On each dataset, we evaluated the methods under two settings: Heard and Unheard, where Heard stands for the standard AudioGoal setting with both visual and audio sensors equipped on the agent, while in the Unheard setting, only the visual sensor is attached. The existing methods and baselines with which we compared are as follows:

1. **Random Agent:** An agent that randomly selects each action and selects *Stop* when it reaches the goal.

2. **Direction Follower:** A hierarchical model that sets intermediate goals  $K$  meters in the predicted direction of arrival (DoA) of the audio. Following [15], we set  $K = 2$  in Replica and  $K = 4$  in Matterport3D.
3. **Frontier Waypoints:** A hierarchical approach that combines the predicted DoA with the boundaries of the explored areas to designate the next waypoint.
4. **Supervised Waypoints:** This model employs supervised learning to predict waypoints within its field of view (FoV), using RGB frames and audio spectrograms.
5. **SoundSpaces [10]:** This model applies CNNs to both acoustic and visual inputs from sensors, using a simple concatenation to combine different modalities. The output is passed to a GRU, and an actor-critic model estimates the value of the state and policy distribution.
6. **LLA [12]:** A conventional phase-based navigation method that uses a topological graph for enhanced path planning.
7. **AV-WaN [15]:** This method utilizes waypoints, represented through a topological graph, to improve navigation efficiency over longer distances. Similar to SoundSpaces, the model uses a simple concatenation to combine different modalities.

In the Replica dataset, under the Heard setting, the proposed AVGN method conspicuously stands out. It achieves an SPL score of 88.5, suggesting optimal path utilization. While its success rate of 97.3 slightly lags behind the state-of-the-art method AV-WaN (98.7), AVGN leads the SNA metric with a dominant score of 71.9, a roughly 1.2 point improvement. In the Unheard setting, AVGN achieves the best performance, with 34.9 in SPL, 59.1 in SR, and 29.0 in SNA. Transitioning to the Matterport3D dataset, under Heard conditions, the performance of AVGN is unparalleled. The SPL, SR, and SNA are 84.3, 88.3, and 57.1 respectively. This places AVGN firmly at the top, especially in the SPL metric, where it outperforms the nearest competitor by a substantial 12 points. Under the Unheard setting, AVGN's excellence persists. It registers the highest scores across all three metrics: 53.2 in SPL, 62.2 in SR, and 38.6 in SNA. To put this into perspective, in the SPL metric alone, AVGN exceeds the closest rival by an impressive 12.3 points.

When the performance of AVGN is compared with LLA, SoundSpaces and AV-WaN, the progress is evident. Particularly in the Matterport3D dataset under the Heard setting, the performance improvements of AVGN are significant. It not only tops the charts, but does so with a clear margin in the SPL, highlighting the advancements made from baseline methodologies to our proposed methods. In essence, the AVGN method introduced in this study signifies a landmark in embodied AI navigation research. Its consistent top-tier performance, especially when evaluated against the backdrop of prior methods, underscores its potential and the considerable progress made in the field.

We also include qualitative results for our model and comparisons with LLA [12], SoundSpaces [10] and AV-WaN [15] in Fig.3. These qualitative results intuitively highlight the strengths and advantages of our methods.

#### 4.4 Ablation Study

In our research, we conduct an ablation study using three distinct settings: (1) without the DSFI module and backtracking rewards; (2) without backtracking rewards; and (3) without the DSFI module. To evaluate the efficacy of our proposed model, we compare these settings within the Heard configurations across both datasets. Detailed outcomes of these evaluations are presented in Table 2. We denote the Directional continuity reward and Historical trajectory reward as  $R_b$  for simplicity.

**Table 2. Ablation study for AVGN.** We evaluate our model under three configurations: 1. AVGN without both DSFI and backtracking rewards; 2. AVGN excluding backtracking rewards; and 3. AVGN without the DSFI module.

4emMethod	Replica			Matterport3D		
	SPL	SR	SNA	SPL	SR	SNA
AVGN w/o DSFI and $R_b$	74.9	88.5	51.8	54.9	78.5	31.8
AVGN w/o $R_b$	84.7	92.0	61.1	81.1	87.2	48.4
AVGN w/o DSFI	74.0	83.9	55.2	56.5	74.8	35.1
AVGN (Ours)	<b>88.5</b>	<b>97.3</b>	<b>71.9</b>	<b>84.3</b>	<b>88.3</b>	<b>57.1</b>

Table 2 elucidates the pivotal roles of the Dual Stage Feature Integration (DSFI) mechanism and the backtracking rewards in our AVGN model through an ablation study. By assessing the model’s performance across different configurations on the Replica Heard and Matterport3D Heard datasets, we draw several conclusions as follows. When both components are removed from the model (AVGN w/o DSFI and  $R_b$ ), there is a pronounced decline in performance across all metrics. The SPL values for Replica and Matterport3D are notably reduced to 74.9 and 54.9, respectively, indicating that both the DSFI and backtracking rewards are key contributors to the model’s efficacy. By retaining only the DSFI and omitting backtracking rewards (AVGN w/o  $R_b$ ), the model achieves an SPL of 84.7 on the Replica dataset. This substantial improvement from the first configuration underlines the DSFI’s indispensable role in synergistically fusing visual and audio inputs. The version without the DSFI but including backtracking rewards (AVGN w/o DSFI) performs similarly to the model lacking both components. However, when comparing the full AVGN with the model that only has DSFI, we see better results with the complete AVGN. This suggests that backtracking rewards contribute positively to the DSFI mechanism. Our fully-fledged AVGN model, which integrates both the DSFI and backtracking rewards, unsurprisingly registers the highest performance metrics on both datasets. With outstanding SPL scores of 88.5 and 84.3 for Replica and Matterport3D, it reiterates the combined strength of both the DSFI and the backtracking rewards. Collectively, this ablation study reaffirms the centrality of the DSFI in our model. Its prowess in amalgamating visual and audio information proves vital in enhancing the robot’s navigation capabilities.

## 5 Conclusions

In this work, we introduced the Audio-Visual Guided Navigation (AVGN) approach, a novel audio-visual navigation methodology. Central to AVGN is the Dual Stage Feature Integration (DSFI), an advanced feature fusion module adept at amalgamating audio and visual cues. Furthermore, we incorporated a triad of rewards specifically designed to address the backtracking issue. Comparative evaluations have demonstrated the superior performance of AVGN over existing methods. As a future direction, we aim to explore audio-visual navigation challenges in more complex scenarios, such as environments with moving sound sources or those with background noise.

**Acknowledgments.** This research is supported by NSF IIS-2309073 and ECCS-2123521. This article solely reflects the opinions and conclusions of authors and not funding agencies.

## References

1. P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018
2. S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017
3. J. Truong, S. Chernova, and D. Batra, "Bi-directional domain adaptation for sim2real transfer of embodied navigation agents," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, 2021
4. Z. Zhao, H. Tang, J. Wan, and Y. Yan, "Monocular expressive 3d human reconstruction of multiple people," in *Proceedings of the 2024 International Conference on Multimedia Retrieval*, 2024, pp. 423–432
5. H. Wang, Z. Yu, Y. Yue, A. Anandkumar, A. Liu, and J. Yan, "Learning calibrated uncertainties for domain shift: A distributionally robust learning approach." in *IJCAI*, 2023, pp. 1460–1469
6. J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023
7. J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence (TETCI)*, vol. 6, no. 2, 2022
8. G. Zhang, H. Tang, and Y. Yan, "Versatile navigation under partial observability via value-guided diffusion policy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 943–17 951
9. Y. Shang, D. Xu, G. Liu, R. R. Kompella, and Y. Yan, "Efficient multitask dense predictor via binarization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 899–15 908

10. C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman, "Soundspaces: Audio-visual navigation in 3d environments," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, Springer, 2020
11. C. Chen, C. Schissler, S. Garg, P. Kobernik, A. Clegg, P. Calamia, D. Batra, P. Robinson, and K. Grauman, "Soundspaces 2.0: A simulation platform for visual-acoustic learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022
12. C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum, "Look, listen, and act: Towards audio-visual embodied navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020
13. Y. Yu, W. Huang, F. Sun, C. Chen, Y. Wang, and X. Liu, "Sound adversarial audio-visual navigation," arXiv preprint [arXiv:2202.10910](https://arxiv.org/abs/2202.10910), 2022
14. C. Chen, Z. Al-Halah, and K. Grauman, "Semantic audio-visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021
15. C. Chen, S. Majumder, Z. Al-Halah, R. Gao, S. K. Ramakrishnan, and K. Grauman, "Learning to set waypoints for audio-visual navigation," arXiv preprint [arXiv:2008.09622](https://arxiv.org/abs/2008.09622), 2020
16. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929), 2020
17. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2021
18. J. Wu, B. Duan, W. Kang, H. Tang, and Y. Yan, "Token transformation matters: Towards faithful post-hoc explanation for vision transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 926–10 935
19. A. D. Ekstrom, "Why vision is important to how we navigate," *Hippocampus*, vol. 25, no. 6, 2015
20. E. C. Tolman, "Cognitive maps in rats and men." *Psychological review*, vol. 55, no. 4, 1948
21. D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020
22. S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," in *Conference on Robot Learning*, PMLR, 2020
23. Z. Al-Halah, S. K. Ramakrishnan, and K. Grauman, "Zero experience required: Plug & play modular transfer learning for semantic visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022
24. A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1–10
25. U. Jain, L. Weihs, E. Kolve, M. Rastegari, S. Lazebnik, A. Farhadi, A. G. Schwing, and A. Kembhavi, "Two body problem: Collaborative visual task completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6689–6699



26. M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6750–6759
27. J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," in *Conference on Robot Learning (CoRL)*. PMLR, 2020
28. Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould, "Vln bert: A recurrent vision-and-language bert for navigation," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, 2021
29. W. Hao, C. Li, X. Li, L. Carin, and J. Gao, "Towards learning a generic agent for vision-and-language navigation via pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020
30. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805), 2018
31. M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9339–9347
32. D. S. Chaplot, S. Gupta, D. Gandhi, A. K. Gupta, and R. Salakhutdinov, "Learning to explore using active neural mapping," in *International Conference on Learning Representations*, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:204770375>
33. D. Gordon, A. Kadian, D. Parikh, J. Hoffman, and D. Batra, "Splitnet: Sim2sim and task2task transfer for embodied visual navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1022–1031
34. H. Kuttruff, *Room acoustics*. Crc Press, 2016
35. J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *Advances in neural information processing systems*, vol. 28, 2015
36. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint [arXiv:1707.06347](https://arxiv.org/abs/1707.06347), 2017
37. M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2019
38. A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021
39. A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," arXiv preprint [arXiv:1709.06158](https://arxiv.org/abs/1709.06158), 2017
40. J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, "The replica dataset: A digital replica of indoor spaces," arXiv preprint [arXiv:1906.05797](https://arxiv.org/abs/1906.05797), 2019
41. P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," arXiv preprint [arXiv:1807.06757](https://arxiv.org/abs/1807.06757), 2018



# Towards Building Secure UAV Navigation with FHE-Aware Knowledge Distillation

Arjun Ramesh Kaushik<sup>1(✉)</sup>, Charanjit Jutla<sup>2</sup>, and Nalini Ratha<sup>1</sup>

<sup>1</sup> University at Buffalo, The State University of New York, Getzville, USA  
{kaushik3,nratha}@buffalo.edu

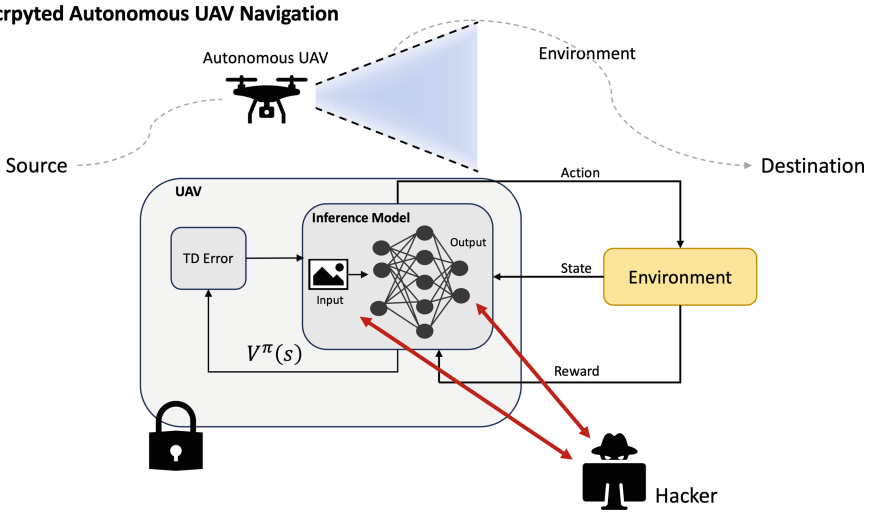
<sup>2</sup> IBM Research, Yorktown Heights, USA  
csjutla@us.ibm.com

**Abstract.** In safeguarding mission-critical systems, such as Unmanned Aerial Vehicles (UAVs), preserving the privacy of path trajectories during navigation is paramount. While the combination of Reinforcement Learning (RL) and Fully Homomorphic Encryption (FHE) holds promise, the computational overhead of FHE presents a significant challenge. This paper proposes an innovative approach that leverages Knowledge Distillation to enhance the practicality of secure UAV navigation. By integrating RL and FHE, our framework addresses vulnerabilities to adversarial attacks while enabling real-time processing of encrypted UAV camera feeds, ensuring data security. To mitigate FHE's latency, Knowledge Distillation is employed to compress the network, resulting in an impressive 18x speedup without compromising performance, as evidenced by an R-squared score of 0.9499 compared to the original model's score of 0.9631. Our methodology underscores the feasibility of processing encrypted data for UAV navigation tasks, emphasizing security alongside performance efficiency and timely processing. These findings pave the way for deploying autonomous UAVs in sensitive environments, bolstering their resilience against potential security threats.

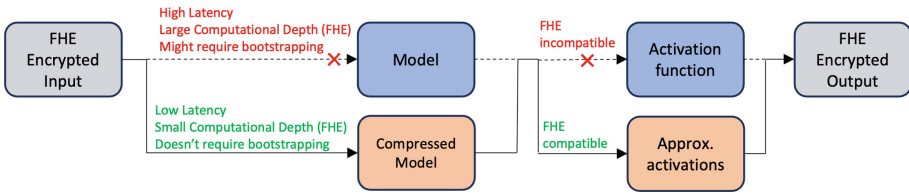
**Keywords:** Autonomous Unmanned Aerial Vehicles · Reinforcement Learning · Fully Homomorphic Encryption · Privacy · Knowledge Distillation

## 1 Introduction

In recent years, the integration of autonomous Unmanned Aerial Vehicles (UAVs) has revolutionized various industries, offering unparalleled capabilities in surveillance, reconnaissance, disaster response, and product delivery [22]. However, ensuring secure navigation of UAVs, particularly in critical scenarios, has become a paramount concern due to the inherent vulnerabilities associated with Deep Learning (DL) techniques and potential adversarial attacks [21][11]. While previous research has made strides in enhancing UAV security [1][19], the computational demands of existing solutions often render them impractical for real-world



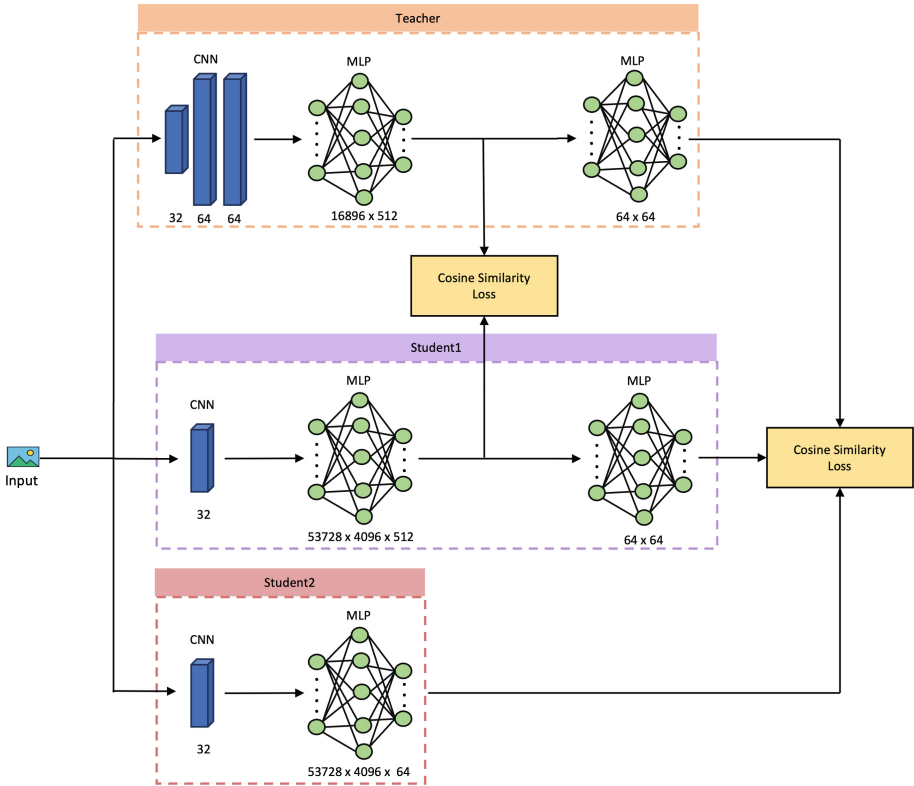
**Fig. 1. Overview:** In an ordinary scenario the UAV is vulnerable to attacks, as the attacker can directly steal the information. FHE-encrypted input and inference prevent this. But, currently, FHE is computationally infeasible.



**Fig. 2.** An overview of the need for an FHE optimized model.

deployment. This paper addresses the pressing need for a secure and feasible architecture for UAV navigation.

While traditional approaches to UAV navigation have relied on vision-based systems incorporating visual mapping, obstacle detection, and path planning [31], recent advancements have shifted towards leveraging Deep Learning and Reinforcement Learning methodologies [24, 27, 28]. In response to the increasing importance of security, recent works have explored various security schemes [1, 4, 13]. However, many existing solutions either prioritize maximum security at the expense of computational feasibility or offer compromised security with practical implementation. Our contribution introduces a secure Reinforcement Learning framework, utilizing the Actor-Critic policy within the Proximal Policy Optimization (PPO) algorithm, capable of seamlessly operating on encrypted real-time video feeds captured by UAV cameras, while remaining resilient to adversarial attacks (Fig. 1). Building upon prior research [1], we present a significantly more feasible architecture in terms of computational efficiency.



**Fig. 3.** We propose a smaller model through Knowledge Distillation to suit FHE needs while maintaining security and accuracy.

In the subsequent sections, we provide a comprehensive overview of how each component of our deep learning model is uniquely adapted to handle encrypted data. Key aspects of our approach include transforming convolutional layers into spectral domain operations, employing generalized matrix multiplication in fully connected layers, and customizing activation functions for the FHE domain through polynomial approximations and comparators. Additionally, navigational steps are extracted through a neural network trained to replicate the OpenAI Gym library. Despite the maximum security provided by FHE, its computational overhead remains significant even after adaptation. To address this challenge, we propose a smaller model through Knowledge Distillation, ensuring feasibility within the FHE framework. Importantly, our research demonstrates the minimal loss of accuracy when mapping teacher and student models to the FHE domain, validating the feasibility of processing encrypted data for UAV navigation tasks.

This work not only addresses immediate security concerns associated with UAVs, but also lays the groundwork for a new era in autonomous aerial systems. By prioritizing security and privacy through FHE integration, our approach

opens avenues for deploying UAVs in sensitive domains where data confidentiality is paramount. The implications extend to applications in military operations, surveillance, and disaster response, where enhanced security measures are essential for the successful execution of critical missions.

## 2 Threat Model

Unmanned Aerial Vehicles (UAVs) deployed in critical scenarios are exposed to various adversarial threats, including (i) Data Poisoning [29], (ii) Model Inversion [17], and (iii) White-box attacks [23, 26]. In our research, we specifically address the scenario where an attacker can intercept communication between the drone and its navigation server, posing a potential risk to the UAV's secure operation. Our primary focus is on establishing secure communication channels between the drone and its navigation server, thereby safeguarding it against Targeted Attacks.

Our solution not only mitigates the risk of Targeted Attacks but also protects against Model Inversion attacks. This is achieved by intelligent adaptation of different components of the model architecture to the encrypted domain. The server can be assumed to hold the weights of the model as matrices, and activation functions as polynomial approximations, instead of the true model architecture in sequence. Consequently, even with full knowledge of such weights, an attacker would be unable to configure the architecture, enhancing the security posture of the UAV system. Moreover, the overall execution of the algorithm takes place on encrypted data. Thus one with access to the secret key can only consume the results. However, adversarial image attacks are not protected by this approach.

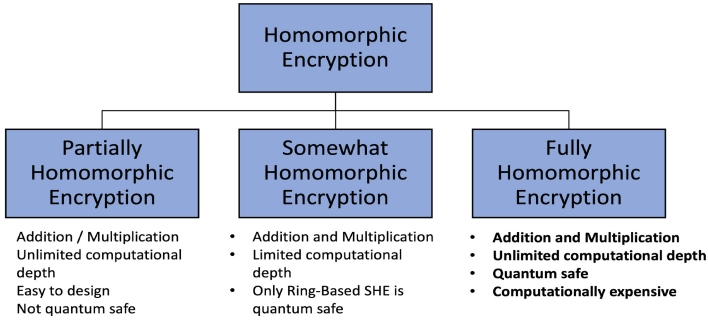
## 3 FHE basics

**Homomorphic encryption (HE) is a cryptographic system that enables computations on encrypted data without the need for decryption, unlike other encryption methods.** In this system, two key components are utilized: public key  $p_k$  and secret key  $s_k$ . Encryption and decryption operations are denoted by  $E$  and  $D$ , respectively. Consider the plaintext values  $x$  and  $y$ , and their corresponding encrypted versions, denoted as  $x' = E(x, p_k)$  and  $y' = E(y, p_k)$ .

Homomorphic Encryption allows for the computation of various operations directly on encrypted ciphertexts. For instance, the addition of encrypted values ( $x' + y'$ ) corresponds to the addition of the original plaintext values ( $x + y$ ). Likewise, the multiplication of encrypted values ( $x' * y'$ ) is equivalent to the multiplication of original plaintext values ( $x * y$ ).

While there exist various Homomorphic Encryption schemes, **FHE stands out as the only one capable of supporting computations on ciphertexts of any depth and complexity** as shown in Fig. 4. Various FHE cryptosystems have been proposed - BFV, BGV, and CKKS schemes [9]. Notably, BFV and

BGV schemes support integers. **In our research, we have employed the CKKS scheme as it supports floating-point decimals.**



**Fig. 4.** Types of Homomorphic Encryption (HE) and their features.

HEAAN, a CKKS FHE scheme, restricts data encryption, allowing only sizes in powers of 2. Hence, we pack our input into arrays of size  $2^n$  before encryption. If the input sizes are not perfect powers of 2, we pad the data with 0s. Although these ciphertexts support Single Instruction Multiple Data (SIMD) operations, they do not provide direct access to individual elements within the ciphertext.

Our research utilizes FHE, specifically the CKKS scheme, to enable secure autonomous UAV navigation using Deep Learning. While FHE allows computations on encrypted data without compromising privacy, certain essential computational operators are yet to be fully implemented in the FHE framework. To address this, we resort to polynomial approximations for these operations. **In this paper, we have developed FHE-compatible operators tailored for autonomous UAV navigation tasks, leveraging a fully learned deep learning network for inference.**

## 4 Related Work

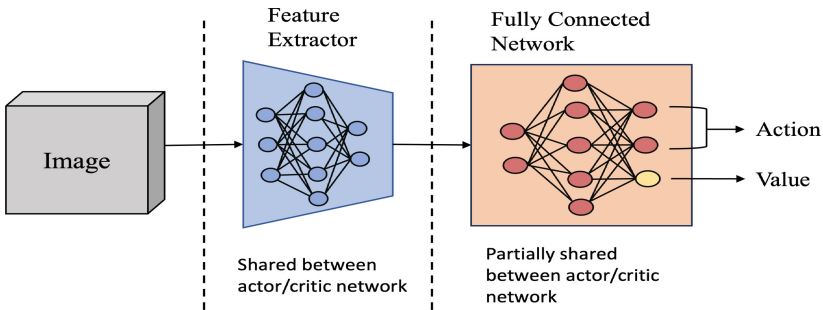
Numerous surveys have delved into the privacy and security challenges specific to UAVs. Works such as [30] and [14] highlight the vulnerability landscape in UAV communication networks, emphasizing the delicate trade-off between robust security and the imperative for lightweight, efficient operations. These discussions underscore the crucial role of encryption in fortifying UAV systems against multifaceted threats, as presented by the authors in [18]. Our research aims to build upon these foundational insights, contributing to the ongoing discourse on UAV security.

Homomorphic Encryption has been employed in prior work to secure computations in the context of UAV navigation. For instance, in [2], the authors propose an extra key generation encryption technique using the Paillier Cryptosystem to prevent cipher data from being compromised. Further, Cheon et al.

[5] explores the development of secure UAVs using a homomorphic public-key encryption method, enabling both secret communication and confidential computation. Another approach focuses on providing a secure and efficient method for third-party UAV controllers to collect and process client data, as demonstrated in [20]. The authors propose a Secure Homomorphic Encryption (SHE) framework, which transfers the FHE encryption to UAVs through an encryption protocol.

Despite notable progress in advancing autonomous systems and encryption methodologies for various applications [13][4][1], achieving a comprehensive and practical solution for secure drone systems has proven elusive. While previous works, such as [4], offer feasible frameworks for drone controllers, they do not address drone security, leaving them vulnerable to attacks when operating autonomously. Similarly, [1] presents a secure Reinforcement Learning-based framework for drone navigation, yet its practical implementation remains infeasible. In contrast to the innovative approach of AutoFHE [3] for accelerating inference in encrypted domain of large CNN models (with a focus on ReLU amongst other activations), our work uses a small model with minimal activation functions.

Among various model compression techniques, including Pruning, Quantization, Decomposition, and Knowledge Distillation [15], our research finds Knowledge Distillation to be particularly effective for FHE. Pruning involves eliminating network components to create sparse models, which, although useful for acceleration and compression, does not significantly reduce computational time for CNNs in FHE. While Quantization typically operates in the BGV scheme, our research focuses on the CKKS scheme [9]. Although Decomposition shows promise, it does not match the effectiveness of reducing network depth through Knowledge Distillation.



**Fig. 5.** Architecture overview of our framework implementing the Actor-Critic algorithm.

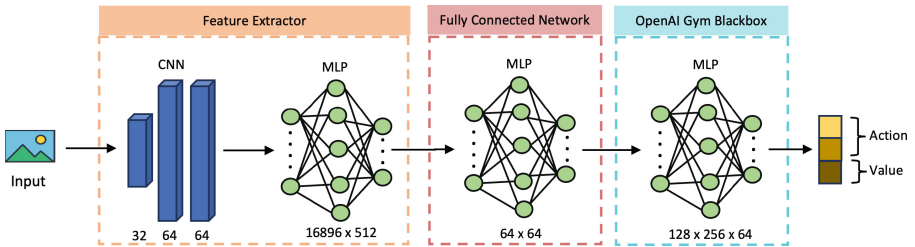
## 5 Proposed Method

The drone is trained using the Actor-Critic Reinforcement Learning algorithm [25]. During training, both the Actor and Critic networks are utilized, whereas, during inferencing, only the Actor network is leveraged. The network architecture can be divided into two segments - Feature Extractor and Fully Connected Network as shown in Fig. 5. The Feature Extractor consists of three convolution blocks and one linear block as shown in Fig. 6. Each convolution block consists of a Convolution layer, Batch Normalization layer, and ReLU activation layer. The linear block consists of a Dense Layer, Batch Normalization layer, and ReLU activation layer. The Fully Connected Network segment consists of two shared linear blocks (shared between Actor and Critic) and an output linear block as in Fig. 6. The shared linear blocks are made up of a dense layer and utilize the TanH activation function.

Computation within the Fully Homomorphic Encryption (FHE) domain introduces several significant limitations, including the absence of individual element access in encrypted arrays, restricted computation depth, heightened time complexity, and the absence of inherent support for operators like comparators. Consequently, we choose to train the Actor-Critic model in the unencrypted domain with data generated in a simulated environment, employing Microsoft's AirSim library and Unreal Engine. Subsequently, leverage the model weights for inference within the encrypted domain. To achieve this, we carefully adapt each component of the Actor-Critic network to seamlessly operate within the FHE domain, addressing specific challenges presented by FHE.

In addition to computational constraints, currently, operations in the FHE domain consume significant time. We must have an efficient model with low inference times and high accuracy. We achieve this with the help of Knowledge Distillation in 2 steps.

Key adaptations within the FHE domain encompass the following components: (i) Model Compression via Knowledge Distillation; (ii) 2-D strided Convolution; (iii) ReLU activation function; (iv) Dense Layer; (v) TanH activation function; and (vi) OpenAI Gym Library. In this section, we provide an in-depth exploration of these adaptations in each layer.



**Fig. 6.** Architecture of the original model (Teacher Network).



## 5.1 Input Adaptations for FHE

The drone’s input comprises of three consecutive images, each captured from the AirSim simulator, with dimensions 50x50. These images are concatenated to form a single input image with dimensions 50x150. In HEAAN, we adopt a strategy where each row of the image is encrypted as a single ciphertext. This approach enables the utilization of SIMD operations, enhancing computational efficiency [16].

Given that HEAAN exclusively supports the encryption of data with sizes as powers of 2, we address this constraint by padding each row of the image with zeros, extending the width to 256. Consequently, the padded input image, now of size 50x256, is encrypted, resulting in a vector of ciphertexts. To facilitate efficient computation, the plaintext weights or filters undergo similar zero-padding, aligning with the dimensions of the padded input image. Importantly, the increase in input size from 50x150 to 50x256 does not impose a significant computational overhead, thanks to the SIMD nature of operations inherent in HEAAN.

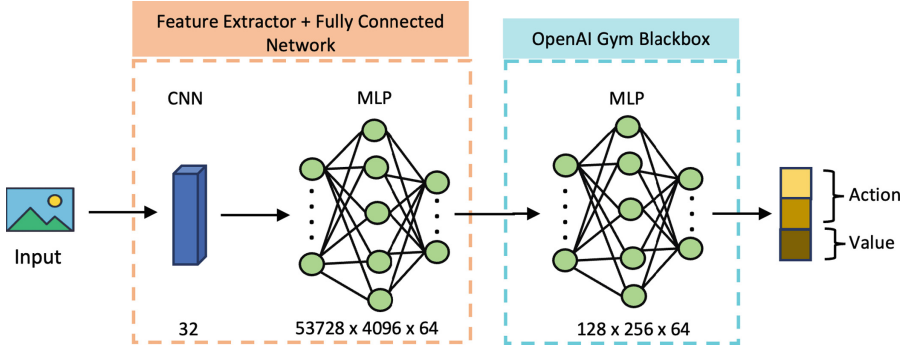
## 5.2 Knowledge Distillation

Knowledge distillation, a representative type of model compression and acceleration, effectively learns a small student model from a large teacher model [10]. In our work, we employ feature-based Knowledge Distillation to compress our original model (Teacher network) to a smaller and FHE-friendly model (Student2 network). We achieve this in 2 steps as shown in Fig. 3, achieving Student1 network first and then using Student1 to further compress the model to Student2. It is important to note that, we perform distillation only on the feature extractor network of while training Student1. As shown in Fig. 3, we train the student networks on the Cosine Similarity Loss between the extracted features. This significantly reduces the inference time, thereby making the FHE implementation more feasible.

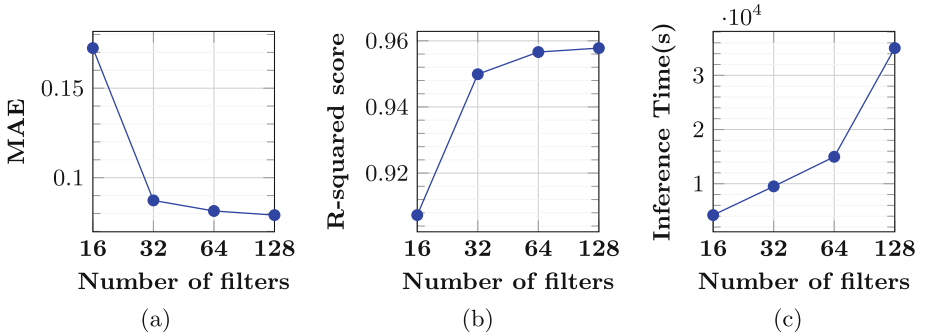
## 5.3 Convolutional Layer

Performing regular convolution in the encrypted domain is extremely computationally inefficient as shown in Table 1 . In our research, we adopt a frequency-domain approach for convolution leveraging the Discrete Fourier transform (DFT). Following are steps performed to achieve 2D convolution in an efficient manner: (i) Perform Homomorphic Fourier Transform (HFT) for each row of 2D Ciphertext using the method in [12]; (ii) Take the transpose of 2D Ciphertext using the method proposed in [32]; (iii) Perform row wise HFT of the new transposed Ciphertext; (iv) Transpose back the 2D Ciphertext (v) Compute the convolution output  $y[n]$  using element-wise multiplication in the frequency domain, as expressed in Equation 1, where  $\mathcal{G}^{-1}$  denotes the inverse Fourier transform, and  $H(u)$  and  $F(u)$  are the DFT of the row of input image and filter, respectively.

$$y[n] = \mathcal{G}^{-1} \{H(u) \cdot F(u)\} \quad (1)$$



**Fig. 7.** Architecture of the final compressed model (Student2 Network) to comply with FHE’s time constraints.



**Fig. 8.** (a) Mean Absolute Error (MAE) for various filter counts in the feature-extractor of the Student network (b) R-squared score for various filter counts in the feature-extractor of the Student network (c) Inference time in seconds for various filter counts in the feature-extractor of the Student network.

The DFT of each input value  $h[v]$  is computed using Equation 2, where  $H[v]$  represents the DFT coefficient at frequency bin  $v$ , and  $N$  is the size of the input.

$$H[u] = \sum_{v=0}^{N-1} h[v] \cdot e^{-j \frac{2\pi}{N} uv} \tag{2}$$

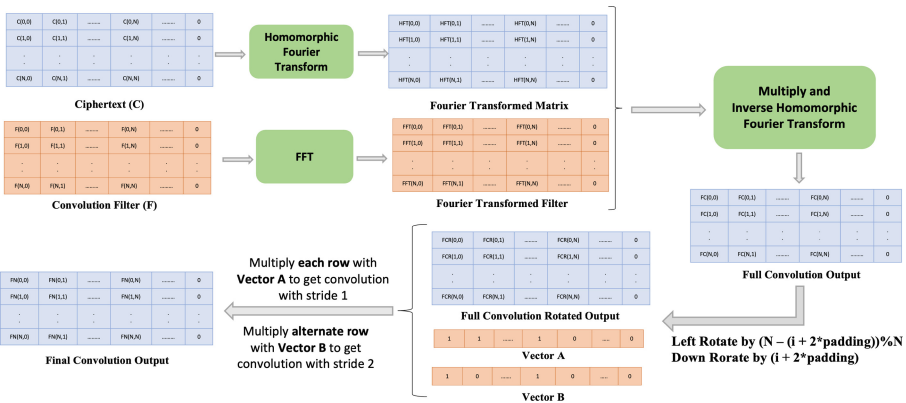
To address the time inefficiency associated with computing the DFT of encrypted data using standard plaintext methods, we employ the Homomorphic Fourier transform. This approach, inspired by Cooley-Tukey matrix factorization [8], facilitates an efficient algorithm for computing the 1-D DFT of encrypted data.

For transforming the plaintext filter into the frequency domain, we utilize the standard Fast Fourier Transform (FFT). The element-wise multiplication between the input and filter in the frequency domain, followed by the inverse DFT, yields the complete convolution output. To achieve a strided convolution,

**Table 1.** Time complexity analysis of convolution in spatial domain and frequency domain, for an image of size  $m \times m$  and filter of size  $n \times n$ . The time complexities below reflect multiplication complexities.

Convolution domain	spatial domain	frequency domain
Time complexity	$O(m^2 * n^2)$	$O(m^2 + 2 * n * \log n)$

a rotational manipulation is applied to the resulting ciphertext. We introduce a leftward rotation of the resulting ciphertext by  $(N - (2 * padding)) \% N$  and downward rotation by  $2 * padding$ , where  $N$  represents the size of the Ciphertext and  $padding$  represents the padded value used to extract DFT convolution output. Additionally, this result is multiplied by an array containing 1s and 0s to obtain appropriate convolution based on the stride value, as illustrated in Fig. 9.



**Fig. 9.** 2D Convolution in FHE Domain. Input ciphertext and weights are multiplied in the frequency domain to obtain full convolution. Final convolution output is obtained by rotating the full convolution as shown above. Different stride-based convolutions can be extracted by multiplying appropriate vectors.

### 5.4 Activation functions

Activation functions play a crucial role in neural networks, but their implementation in the context of FHE presents unique challenges [7]. FHE libraries lack native support for comparison operations, necessitating the use of approximations like CompG for the sign function [6]. Normalization is essential to align input values within the required range, achieved by scaling the outputs of convolutional layers based on the maximum observed absolute values during training. This scaling factor is determined by the maximum of the absolute values of the

inputs' observed range. Following the application of the approximations, positive input values are rescaled to their original range using the inverse of the scaling factor.

In our research, we adopt a composite approximation technique for comparison in ReLU implementation. This method evaluates the input value  $a$  against zero, encoding the output as 1 for  $a > 0$ , 0 for  $a < 0$ , and 0.5 for  $a = 0$ , and subsequently calculates the final ReLU output by multiplying this result by the input value  $a$ . Additionally, we address the challenges of implementing exponential functions in FHE by employing an 8-degree polynomial approximation of TanH restricted to the range  $[-2, 2]$ . This approach allows for a closer approximation while mitigating the limitations of FHE in handling exponential functions. The performance of our approximation is evaluated through the relative error of 2000 points within the specified range, providing insights into its effectiveness and accuracy as shown in Fig . 10.

### 5.5 Flattening layer

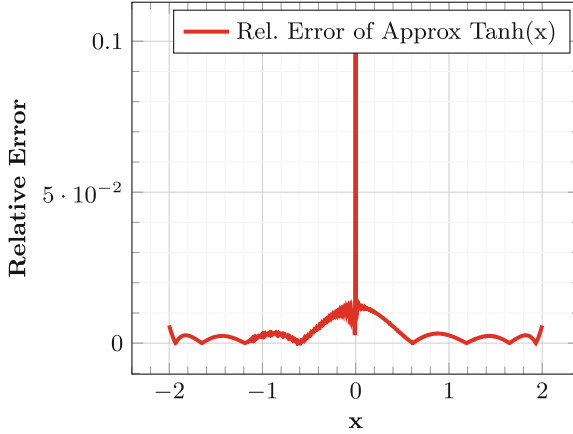
The flattening operation is usually performed on the convolution outputs. Flattening operation is not possible in FHE without decrypting and re-encrypting the ciphertexts, as it involves changing the length of ciphertexts. To circumvent this issue, we perform element-wise multiplication of the weights and convolution output. Element-wise multiplication is an extremely time-consuming operation as it involves multiplication, addition, and left rotation. We multiply each ciphertext with its corresponding weight vector and add it to a temporary ciphertext initialized to zeros. Then, we perform a summation of the ciphertext elements through repetitive left rotation and addition  $N-1$  times.

### 5.6 Fully-Connected Layer

A Fully Connected Layer is adapted to FHE as the matrix multiplication of ciphertext inputs and plaintext weight matrices. Each row of weight matrix is multiplied with the ciphertext and the elements of the ciphertext are summed through left rotation.

### 5.7 OpenAI Gym Library

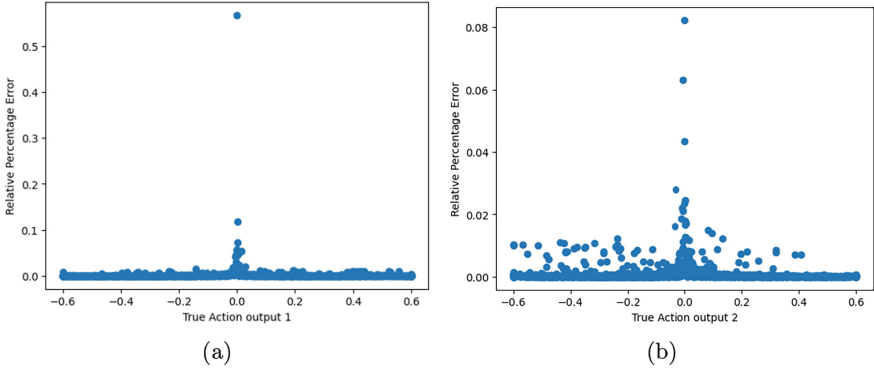
We have adapted the OpenAI Gym Library to FHE through a 3-layer neural network as in Fig. 6 and Fig. 7. This is due to the limitations of FHE in modeling probability distributions. The neural network learns the probability distribution and maps the final 64-dimension latent vector to the action output. The model is trained in the unencrypted domain and its weights are used for inferencing in FHE.



**Fig. 10.** Relative error of  $f(x)$  over the interval  $[-2, 2]$ , where  $f(x)$  is the polynomial approximation of  $\text{Tanh}(x)$ . Relative error of  $f(x) = \frac{|f(x) - \tanh(x)|}{|\tanh(x)|}$ .

## 6 Results

Experiments were performed in the encrypted domain on a subset of randomly selected samples from the testing set of the unencrypted domain. We evaluated our results from the FHE-adapted Reinforcement Learning framework against the expected results from the Reinforcement Learning framework in the unencrypted domain. Table 2 depicts the mean absolute error (MAE) across each block in the Teacher and Student networks within the encrypted domain. Crucially, the regression-based prediction output remained consistent between the FHE version and the plaintext counterpart for the tested samples, indicating coherence in predictive outcomes. We have also achieved an **R-squared score of 0.9631 for the Teacher network** and **0.9499 for the Student2 network** with the end-to-end FHE-based Reinforcement Learning framework, in comparison with results in the unencrypted domain. Additionally, Table 3 presents the average processing time across each block in the Teacher and Student networks. We achieve an 18x improvement in inference speed with Knowledge Distillation. These findings substantiate the efficacy of our FHE-adapted network, showcasing the viability of FHE in preserving model accuracy while ensuring data confidentiality.



**Fig. 11.** Relative percentage errors of actions on adaption of OpenAI Gym Library to FHE.

**Table 2.** Layerwise average Mean Absolute Error (MAE) between plain-text and FHE model intermediate outputs in Teacher and Student networks.

Layer	Average MAE		
	Teacher	Student1	Student2
Convolution	0.0779	0.0860	0.0873
Linear	0.0129	0.0185	0.0203
OpenAI Gym Library Blackbox	0.0210	0.0206	0.0201

**Table 3.** Time taken by the Teacher and Student networks.

Layer	Inference Time (seconds)		
	Teacher	Student1	Student2
Convolution	1,006,337.18	9,508.44	9,510.22
Linear	13,662.48	43,670.76	41,989.52
OpenAI Gym Library Blackbox	4,574.82	4,725.92	4,668.19
Total	1,024,754.48	57,905.12	56,167.93

## 7 Conclusion

This paper introduces a groundbreaking end-to-end homomorphically encrypted Unmanned Aerial Vehicle (UAV) navigation system, leveraging a fusion of reinforcement learning and deep neural networks. Given Fully Homomorphic Encryption’s (FHE) high latency, our results indicate a significant speedup (18x) through Knowledge Distillation. In addition, we seamlessly incorporate convolutional layers, fully connected networks, activation functions, and the OpenAI Gym Library into the FHE domain. The use of the Homomorphic Fourier Transform facilitates efficient convolutions, and an approximate comparator enables

the effective mapping of the ReLU activation function. Furthermore, we have devised Tanh approximations, functional mappings from latent feature vectors to action outputs for the Gym Library, and implemented fully connected layers within the FHE domain. In our evaluation of inference, our proposed FHE-based compressed architecture demonstrates lower latency with minimal error across each block in the network, showcasing no discernible accuracy loss when compared to its plaintext counterpart.

## References

1. Aggarwal, V., Kaushik, A.R., Ratha, N.: Enhancing privacy and security of autonomous uav navigation. In: Conference on Artificial Intelligence (2024)
2. Alzahrani, M., Khan, N., Georgieva, L., Bamahdi, A., Abdulkader, O., Alahmadi, A.: Protecting attacks on unmanned aerial vehicles using homomorphic encryption. *Indonesian Journal of Electrical Engineering and Informatics* **11**(1), 88–96 (2023)
3. Ao, W., Boddeti, V.N.: Autofhe: Automated adaption of cnns for efficient evaluation over fhe. *Cryptology ePrint Archive*, Paper (2023)
4. Cheon, J.H., Han, K., Hong, S.M., Kim, H.J., Kim, J., Kim, S., Seo, H., Shim, H., Song, Y.: Toward a secure drone system: Flying with real-time homomorphic authenticated encryption. *IEEE Access* **6**, 24325–24339 (2018)
5. Cheon, J.H., Han, K., Hong, S.M., Kim, H.J., Kim, J., Kim, S., Seo, H., Shim, H., Song, Y.: Toward a secure drone system: Flying with real-time homomorphic authenticated encryption. *IEEE Access* **6**, 24325–24339 (2018)
6. Cheon, J.H., Kim, D., Kim, D.: Efficient homomorphic comparison methods with optimal complexity. *Cryptology ePrint Archive*, Paper (2019)
7. Cheon, J.H., Kim, D., Kim, D., Lee, H.H., Lee, K.: Numerical method for comparison on homomorphically encrypted numbers. *Cryptology ePrint Archive*, Paper (2019)
8. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex fourier series. *Math. Comput.* **19**(90), 297–301 (1965)
9. Gorantala, S., Springer, R., Gipson, B.: Unlocking the potential of fully homomorphic encryption. *Commun. ACM* **66**(5), 72–81 (apr 2023)
10. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. *Int. J. Comput. Vision* **129**(6), 1789–1819 (2021)
11. Guo, R., Wang, B., Weng, J.: Vulnerabilities and attacks of uav cyber physical systems. In: Proceedings of the 2020 International Conference on Computing, Networks and Internet of Things. p. 8–12. CNIOT '20, Association for Computing Machinery, New York, NY, USA (2020)
12. Han, K., Hhan, M., Cheon, J.H.: Improved homomorphic discrete fourier transforms and fhe bootstrapping. *IEEE Access* **7**, 57361–57370 (2019). <https://doi.org/10.1109/ACCESS.2019.2913850>
13. Hassija, V., Chamola, V., Agrawal, A., Goyal, A., Luong, N.C., Niyato, D., Yu, F.R., Guizani, M.: Fast, reliable, and secure drone communication: A comprehensive survey. *IEEE Communications Surveys & Tutorials* **23**(4), 2802–2832 (2021)
14. Hassija, V., Chamola, V., Agrawal, A., Goyal, A., Luong, N.C., Niyato, D., Yu, F.R., Guizani, M.: Fast, reliable, and secure drone communication: A comprehensive survey. *IEEE Communications Surveys & Tutorials* **23**(4), 2802–2832 (2021)









15. He, Y., Xiao, L.: Structured pruning for deep convolutional neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(5), 2900–2919 (2024)
16. Jung, W., Lee, E., Kim, S., Kim, J., Kim, N., Lee, K., Min, C., Cheon, J.H., Ahn, J.H.: Accelerating fully homomorphic encryption through architecture-centric analysis and optimization. *IEEE Access* **9**, 98772–98789 (2021)
17. Khowaja, S.A., Khuwaja, P., Dev, K., Antonopoulos, A.: Spin: Simulated poisoning and inversion network for federated learning-based 6g vehicular networks. In: *ICC - IEEE International Conference on Communications*. pp. 6205–6210 (2023)
18. Krishna, C.G.L., Murphy, R.R.: A review on cybersecurity vulnerabilities for unmanned aerial vehicles. In: *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. pp. 194–199 (2017)
19. Liu, T., Guo, H., Danilov, C., Nahrstedt, K.: A privacy-preserving data collection and processing framework for third-party uav services. In: *IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. pp. 683–690 (2020)
20. Liu, T., Guo, H., Danilov, C., Nahrstedt, K.: A privacy-preserving data collection and processing framework for third-party uav services. In: *IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. pp. 683–690 (2020)
21. Mekdad, Y., Aris, A., Babun, L., Fergougui, A.E., Conti, M., Lazzeretti, R., Uluagac, A.S.: A survey on security and privacy issues of uavs. *Computer networks* **224** (2023-04)
22. Mohsan, S.A.H., Khan, M.A., Noor, F., Ullah, I., Alsharif, M.H.: Towards the unmanned aerial vehicles (uavs): A comprehensive review. *Drones* **6**(6) (2022), <https://www.mdpi.com/2504-446X/6/6/147>
23. Raja, A., Njilla, L., Yuan, J.: Blur the eyes of uav: Effective attacks on uav-based infrastructure inspection. In: *IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. pp. 661–665 (2021)
24. Rezwan, S., Choi, W.: Artificial intelligence approaches for uav navigation: Recent advances and future challenges. *IEEE Access* **10**, 26320–26339 (2022)
25. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. *CoRR* **abs/1707.06347** (2017), <http://dblp.uni-trier.de/db/journals/corr/corr1707.html>
26. Sun, H., Guo, J., Meng, Z., Zhang, T., Fang, J., Lin, Y., Yu, H.: Evd4uav: An altitude-sensitive benchmark to evade vehicle detection in uav (2024)
27. Wang, C., Wang, J., Shen, Y., Zhang, X.: Autonomous navigation of uavs in large-scale complex environments: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **68**(3), 2124–2136 (2019). <https://doi.org/10.1109/TVT.2018.2890773>
28. Wang, C., Wang, J., Wang, J., Zhang, X.: Deep-reinforcement-learning-based autonomous uav navigation with sparse rewards. *IEEE Internet Things J.* **7**(7), 6180–6190 (2020). <https://doi.org/10.1109/JIOT.2020.2973193>
29. Wang, Z., Wang, sB., Zhang, C., Liu, Y., Guo, J.: Defending against poisoning attacks in aerial image semantic segmentation with robust invariant feature enhancement. *Remote Sensing* **15**(12) (2023), <https://www.mdpi.com/2072-4292/15/12/3157>



30. Yang, W., Wang, S., Yin, X., Wang, X., Hu, J.: A review on security issues and solutions of the internet of drones. *IEEE Open Journal of the Computer Society* **3**, 96–110 (2022)
31. Yuncheng Lu, Zhucun Xue, G.S.X., Zhang, L.: A survey on vision-based uav navigation. *Geo-spatial Information Science* **21**(1), 21–32 (2018)
32. Zekri, A.: Enhancing the matrix transpose operation using intel avx instruction set extension. *International Journal of Computer Science & Information Technology* **6**, 67–78 (06 2014). <https://doi.org/10.5121/ijcsit.2014.6305>



# Zero-Shot Object Navigation with Vision-Language Models Reasoning

Congcong Wen<sup>1,2,4</sup> , Yisiyuan Huang<sup>3</sup>, Hao Huang<sup>1,2</sup> , Yanjia Huang<sup>3</sup> ,  
Shuaihang Yuan<sup>1,2</sup> , Yu Hao<sup>1,2</sup> , Hui Lin<sup>4</sup> , Yu-Shen Liu<sup>5</sup> ,  
and Yi Fang<sup>1,2</sup> 

<sup>1</sup> Embodied AI and Robotics (AIR) Lab, New York University Abu Dhabi,  
Abu Dhabi, UAE

hh1811@nyu.edu

<sup>2</sup> Center for Artificial Intelligence and Robotics, New York University Abu Dhabi,  
Abu Dhabi, UAE

<sup>3</sup> Tandon School of Engineering, New York University, New York, USA

<sup>4</sup> University of Science and Technology of China, Anhui,  
People's Republic of China

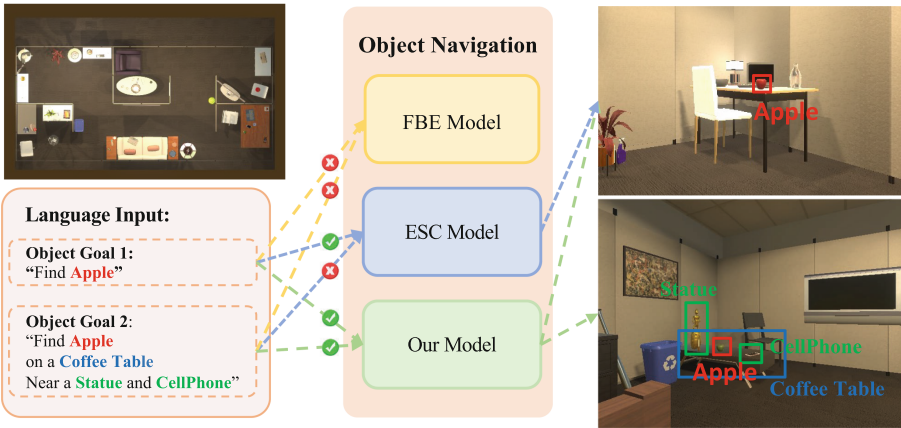
<sup>5</sup> School of Software, Tsinghua University, Beijing, People's Republic of China

**Abstract.** Object navigation is crucial for robots, but traditional methods require substantial training data and cannot be generalized to unknown environments. Zero-shot object navigation (ZSON) aims to address this challenge, allowing robots to interact with unknown objects without specific training data. Language-driven zero-shot object navigation (L-ZSON) is an extension of ZSON that incorporates natural language instructions to guide robot navigation and interaction with objects. In this paper, we propose a novel Vision Language model with a Tree-of-thought Network (VLTNet) for L-ZSON. VLTNet comprises four main modules: vision language model understanding, semantic mapping, tree-of-thought reasoning and exploration, and goal identification. Among these modules, Tree-of-Thought (ToT) reasoning and exploration module serves as a core component, innovatively using the ToT reasoning framework for navigation frontier selection during robot exploration. Compared to conventional frontier selection without reasoning, navigation using ToT reasoning involves multi-path reasoning processes and backtracking when necessary, enabling globally informed decision-making with higher accuracy. Experimental results on PASTURE and RoboTHOR benchmarks demonstrate the outstanding performance of our model in LZSON, particularly in scenarios involving complex natural language as target instructions. Videos are available at <https://vlt-lzson.github.io/>.

**Keywords:** Zero-shot Object Navigation · Vision-Language Model (VLM) · Large Language Model (LLM) · LLM Reasoning

# 1 Introduction

Object navigation, a fundamental task in robotics, is crucial for robots to intelligently explore an environment and interact with objects in the environment. Conventional methods rely on extensive visual training data containing labeled objects from the environment, limiting their ability to generalize to unknown and unstructured environments. To remedy this limitation, recent research [12, 23, 29, 41] explores zero-shot object navigation (ZSON), which allows robots to navigate and interact with unknown objects without the corresponding labeled training data. However, while effective in basic navigation, this method often falls short in scenarios requiring intricate interaction and communication, which are essential for enhanced autonomy and more robust human-robot collaborations.



**Fig. 1.** Comparison of different object navigation methods under two types of language input: 1) word input with only object category, 2) sentence input with detailed spatial descriptions. a) FBE model [36]: cannot accept either word or sentence input. b) ESC model [43]: only accepts word input. c) Our model: accepts both word and sentence as input.

To improve autonomous agents and human-robot interaction which is lacking in the traditional ZSON, there is a growing interest in Language-driven Zero-Shot Object Navigation (L-ZSON). L-ZSON guides agents using natural language instructions to require agents to follow the textual or spoken guidance to reach the specified unseen objects or locations. The pioneering efforts [12, 13, 29] have leveraged Large Language Models (LLMs) for L-ZSON. For instance, Huang et al. [17] introduce VLMaps, a spatial map representation that integrates pre-trained visual-language features with 3D reconstruction of a physical environment. Zhou et al. [43] introduce a novel Exploration with Soft common sense Constraints (ESC) module that utilizes a pre-trained LLM for scene understanding and common sense reasoning. Nonetheless, these approaches can only

handle instructions that explicitly contain object categories, failing to navigate to unknown objects or objects described by spatial or visual attributes in the instructions. To remedy the problem, Gadre et al. [13] build the PASTURE benchmark, which more closely reflects real-world scenarios and provides a more rigorous evaluation of L-ZSON. Therefore, we choose this benchmark to evaluate the performance of our proposed L-ZSON method. Furthermore, it is worth noting that existing works employ standard LLMs for common-sense reasoning or decision-making. However, although LLMs are powerful in many applications, they can still struggle to self-assess their decisions during reasoning processes, thus potentially leading to sub-optimal decisions.

To resolve this critical problem of making more effective decisions in dynamic environments, in this paper, we propose a novel Vision Language Model with Tree-of-thoughts NETWORK, named *VLTNet*, for L-ZSON. VLTNet consists of four core modules: vision language model understanding, semantic mapping, tree-of-thoughts reasoning and exploration, and goal identification. Specifically, we first leverage the vision language model understanding module to perform scene understanding. Then, we use the semantic mapping module to build a semantic navigation map. Next, we utilize the tree-of-thoughts reasoning and exploration module to select frontiers based on common sense reasoning for exploration. Finally, we employ the goal identification module to determine whether the current object being navigated to matches the target object. A significant novelty of our paper is the utilization of the Tree-of-Thoughts (ToT) reasoning framework for frontier selection in robot exploration. As shown in Fig. 1, our model with ToT reasoning can incorporate goal-based instructions of varying complexity to choose the optimal frontier. Unlike conventional LLMs, ToT equips models with the capacity to engage in deliberate, multi-path reasoning processes, enabling them to self-evaluate choices and make informed decisions for the action. This self-evaluation reasoning framework also allows models to anticipate future prediction and backtrack when necessary to make globally informed decisions. Experimental results conducted on two benchmarks, PASTURE [13] and RoboTHOR [9], demonstrate that our model excels in L-ZSON tasks, particularly in complex ZSON tasks that involve natural language as guidance.

## 2 Related Work

*Object Goal Navigation* The primary task of goal-conditioned navigation is to guide robots towards distinct targets based on varying specifications. These specifications can be categorized into position goals, *i.e.*, predefined spatial coordinates [6, 7]; image goals, *i.e.*, locations that match a given image view [25, 44]; and object goals, *i.e.*, locations containing specific objects that the agent needs to find [2, 5, 13, 43]. Our research focus on object goal navigation task, which requires the robot to locate and navigate towards specific objects within an environment.

In order to develop agents capable of navigating previously unseen environments, recent work has shifted focus to Zero-shot Object Navigation (ZSON)

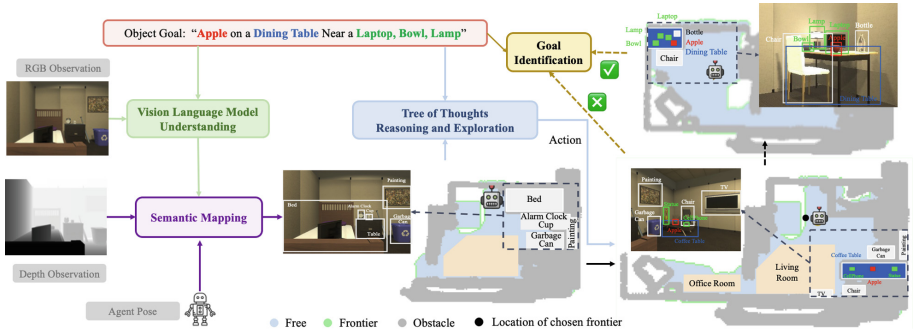
[12, 23, 29, 41]. Nonetheless, most ZSON approaches only take in object names as targets, which can sometimes lead to inefficiency and inaccuracy when navigating through complex environments. Therefore, Language Driven Zero-shot Object Navigation (L-ZSON) were studied as a subset of ZSON, aiming to interpret object goals and descriptive cues from natural language input [12, 13, 29].

*Exploration strategies* Currently, the exploration strategies in object goal navigation can be divided into two main categories: learning-based and frontier-based.

**Learning-based exploration** strategies can be divided into two lines. The first utilizes pre-trained visual encoders [16, 31] to convert egocentric images into descriptive feature vectors, which were fed to train a robot navigation policy by employing imitation learning or reinforcement learning [8, 10, 19, 24, 33, 38]. The second constructs an explicit semantic graph and then the navigation policies are then trained to identify locations of goal objects with the semantic graphs [6, 26, 42]. Learning-based object goal navigation methods rely on training data to fine-tune the navigation policies of agents, often necessitating intricate reward engineering [13]. Furthermore, these methods often face difficulties in generalizing with new objects or unfamiliar environments drastically different from their training data [43].

**Frontier-based exploration** strategies address the limitations of the learning-based approaches. Frontier-based exploration (FBE) [36] is a heuristic algorithm to navigate a robot or an agent in an unseen environment. By reconstructing a depth map of the environment, and marking the boarder between the explored (known) area and unexplored (unknown) area as “frontiers”, FBE iteratively selects the closest frontier to explore. In addition to being used for constructing depth maps [21, 34] and semantic maps [15, 39] in free-exploration tasks, FBE is also employed in real-world object navigation[14]. FBE has also been adapted in ZSON models with different variations. For example, CLIP on Wheel (CoW) [13] generates text-to-image relevance depth maps based on RGB and depth observations, which is then used to determine the region of interest in FBE [36]. ESC [43] and L3MVN [40] employs FBE by using an LLM to assign scores to each potential frontier. However, the methods of assigning numerical scores to each frontier do not account for the complex interrelations between objects and the environment. Therefore, to solve this problem, our VLTNet aims to use an LLM to incorporate more human-like reasoning in navigation.

*LLM Reasoning* Although LLMs have emerged as powerful tools in various domains to understand and generate human-like text [4, 11, 30]. Since the vanilla LLMs are trained with the aim for natural language processing, they tend to perform poorly in tasks such as arithmetic, common sense, and symbolic reasoning [32]. Nonetheless, Wei et al. [35] proposed Chain-of-Thought (CoT) prompting to significantly boost LLM’s reasoning capability, by instructing them to explicitly output the reasoning process. Building up on the linear progression of thoughts in CoT, Tree-of-Thoughts (ToT) [37] proposed a branching reasoning structure that further boosts LLM’s reasoning ability, by instructing LLMs to simulate a discussion among several experts on a given question, until reaching a consensus among these simulated experts. In our study, we seek to employ ToT for



**Fig. 2.** Illustration of our VLTNet framework. During navigation, the *Vision Language Model (VLM) Understanding* module obtains the observed objects by parsing the current RGB observations of an agent. Based on the object locations provided by both the VLM Understanding module and depth observations from the agent, the *Semantic Mapping* module reconstructs a semantic navigation map containing rooms, objects, and frontiers. Conditioned on the navigation instruction and semantic navigation map, the agent then performs common sense reasoning via the *Tree of Thoughts Reasoning and Exploration* module to infer the most probable location of the goal object, and select the corresponding frontier to explore. Upon the VLM Understanding module grounding a candidate object in the same category as the goal object, the *Goal Identification* module further verifies if the candidate object reached by the agent matches the description from the navigation instruction.

decision-making in L-ZSON, which empowers LLMs, *e.g.*, GPT-3.5 [28] to be able to consider complex interrelations between goal objects and their surroundings, so that LLMs have complete analytical and reasoning autonomy during the frontier selection process.

## 3 Methods

### 3.1 Problem Statement

L-ZSON is designed to validate the capability of an intelligent robot or agent system to navigate to the target or goal objects specified by natural language instructions, without any prior knowledge of the target. In this task, the fundamental components include: (1) a natural language instruction  $L$ , which consists of a sequence of words representing the task to be performed by the agent, encompassing descriptions of the target object, location cues, and directional instructions; (2) an environment representation  $S_t$ , denoting the current state or observation of the agent at time  $t$ , typically encapsulating the observed information about the environment; and (3) a collection of objects within the environment, denoted as  $\mathcal{O}$ , where each object  $o_i \in \mathcal{O}$  is assigned a unique identifier and optionally possesses additional attributes, such as position and appearance.

The objective of L-ZSON is to generate a sequence of actions  $\mathcal{A}$  that guides the agent to navigate within the environment and reach the target object  $o^* \in \mathcal{O}$

specified in an instruction  $I$ , which mathematically represented as:

$$\mathcal{A}^* = \arg \max_{\mathcal{A}} P(\mathcal{A} \mid L, S_0, \mathcal{O}) \quad (1)$$

where  $P(\mathcal{A} \mid L, S_0, \mathcal{O})$  represents the probability of a generated action sequence  $\mathcal{A}$ , given the language instruction  $L$ , initial state  $S_0$ , and object set  $\mathcal{O}$ . The central challenge lies in maximizing the likelihood of selecting the optimal action sequence, enabling the agent to navigate to the target object without prior knowledge or tailored training for that object.

### 3.2 VLTNet for L-ZSON

**Overview** We present a novel VLTNet tailored for the L-ZSON task, consisting of four core modules as shown in Fig. 2: *Vision Language Model (VLM) Understanding* module, *Semantic Mapping* module, *Tree of Thoughts Reasoning and Exploration* module, and *Goal Identification* module. At each time  $t$  during navigation, the *VLM Understanding* module leverages a VLM to perform semantic parsing from the observed RGB image  $I_t$ , enhancing the model’s understanding of the environment semantics. Subsequently, the *Semantic Mapping* module integrates the semantically parsed image  $I_t^s$  generated from the VLM Understanding module, depth image  $D_t$  captured by the agent, and the agent pose  $P_t^a$  to construct a more comprehensive semantic map  $M_t$ , defining objects based on the parsed semantic and spatial relationships. Following that, the *Tree of Thoughts Reasoning and Exploration* module strategically selects a frontier to perform a frontier-based exploration, considering the agent position and the target object information. Lastly, the *Goal Identification* module assesses the alignment of the currently reached object with the goal object specified in the instruction  $L$ , ensuring navigation consistency. This framework aims to enhance ZSON through a seamless and intelligent integration of scene understanding, semantic mapping, LLM-based frontier selection, and goal object consistency checking, harnessing the power of LLMs equipped with reasoning ability.

**Vision Language Model Understanding** VLMs excel in semantic understanding, as they have been pre-trained on vast amounts of textual and visual data, which enables them to associate texts with the corresponding visual objects, allowing for a deeper comprehension of the content within images. Specifically, we employ the Grounded Language-Image Pre-training (GLIP) [22] due to its inherent advantages in grounding language description with visual context. Inspired by ESC [43], considering both low-level and high-level scene contexts, we define a set of common objects and rooms in an indoor environment as prompts fed into GLIP. We establish multiple prompts, such as the object prompt ( $p_o$ ) and room prompt ( $p_r$ ), to query the GLIP model in generating detection results. Here,  $p_o$  and  $p_r$  correspond to object and room categories, respectively, as represented in natural language. Specifically, at time  $t$ , we can obtain the detected objects  $\{o_{t,i}\}$ , rooms  $\{r_{t,i}\}$  and bounding boxes  $\{b_{t,i}^o\}$  and

$\{b_{t,i}^r\}$  of the objects and rooms from the currently observed image  $I_t$ :

$$\{o_{t,i}, b_{t,i}^o, r_{t,i}, b_{t,i}^r\} = \text{GLIP}(I_t, p_o, p_r) \in I_t^s \quad (2)$$

where  $I_t^s$  is a semantically parsed image.

**Semantic Mapping** Typically, we need to generate a navigation map that is essential for guiding an agent to make informed decisions during navigation in a complex environment. To achieve this, we utilize the function  $\text{Nav\_M}(\cdot)$  to generate the navigation map. Specifically, at time  $t$ , we utilize depth information obtained from the agent, along with the agent pose  $P_t^a$ , to calculate 3D points from  $D_t$ . These points are then voxelized into 3D voxels. Subsequently, we project these 3D voxels from the top to produce a 2D navigation map  $\mathcal{M}_{nav}$ . We formulate the above process as:

$$\mathcal{M}_{nav} = \text{Nav\_M}(D_t, P_t^a). \quad (3)$$

$\mathcal{M}_{nav}$  provides information about the layouts, obstacles, pathways, landmarks, and other relevant details within a specific area. Furthermore, we also incorporate the semantic understanding of objects and rooms that are obtained by the VLM Understanding module to generate a semantic navigation map  $\mathcal{M}_{sem}$  using  $\text{Sem\_M}(\cdot)$  function:

$$\mathcal{M}_{sem} = \text{Sem\_M}(\mathcal{M}_{nav}, \{o_{t,i}, b_{t,i}^o, r_{t,i}, b_{t,i}^r\}). \quad (4)$$

Semantic information, including the types of objects and rooms associated with detected objects in 3D space, is projected onto a 2D plane to create  $\mathcal{M}_{sem}$ . This semantic navigation map  $M_t := \mathcal{M}_{sem}$  obtained at each time  $t$  enables the agent to navigate through the environment with a deeper understanding of the objects and their arrangements, making it more capable of handling complex and dynamic scenarios.

**Tree-of-Thoughts Reasoning and Exploration** Due to limitations in the field of agent view or the presence of obstacles, target objects often do not appear within the initial view of an agent. Thus, it is necessary to design an efficient algorithm that enables the agent to explore the environment to swiftly locate the target object quickly. Frontier-based exploration aims at autonomously exploring unknown environments. The core idea is to direct an agent towards the boundaries, known as ‘‘frontiers’’, between explored and unexplored areas, ensuring a systematic and efficient exploration. However, traditional frontier-based exploration algorithms [43] usually lead an agent to select the nearest frontier to minimize traversal distance. Given the complexity of certain environments, naively choosing the closest frontier is often not an optimal solution.

To tackle this limitation, we harness the common sense knowledge inherent in LLMs. By analyzing  $M_t$ , our approach identifies unexplored areas that *are likely proximate to the target object*. Unlike previous methods [43] that rely on



Probabilistic Soft Logic (PSL) [3] and craft a bunch of intricate rules to determine the optimal frontier, our approach offers a fresh perspective: *we utilize LLMs to select the frontier that most likely directs to the goal object*. Noting the potential inaccuracies caused by multiple candidate frontiers fed to an LLM in a native way, we integrate the Tree of Thoughts (ToT) mechanism [37] to let the LLM reason about the optimal frontier to select. ToT employs a structured tree-based decision-making process, allowing for organized and systematic exploration, which enhances the model’s ability to make informed decisions in complex environments. Specifically, given a set of frontier candidates  $\{f_n\}_{n=1}^N$  return by [43], we apply the ToT reasoning, as depicted in Algorithm 1 [37], to decide on the optimal frontier for the next move. To instantiate a ToT, we need to implement four components: thought decomposition, thought generation, state evaluation, and tree search. These components are outlined in the comments of Algorithm 1, which are highlighted in blue.

---

**Algorithm 1** ToT reasoning( $x, m, G, k, V, T, b$ )
 

---

**Require:** Input  $x$ , an LLM  $m$ , thought generator  $G$  & size limit  $k$ , states evaluator  $V$ , step limit  $T$ , breadth limit  $b$ .

$S_0 \leftarrow \{x\}$  ▷ Thought decomposition.

**for**  $t = 1, \dots, T$  **do** ▷ Tree search.

$S'_t \leftarrow \{[s, z] \mid s \in S_{t-1}, z_t \in G(m, s, k)\}$  ▷ Thought generation.

$V_t \leftarrow V(m, S'_t)$  ▷ Thought evaluation.

$S_t \leftarrow \arg \max_{S \subset S'_t, |S|=b} \sum_{s \in S} V_t(s)$

**end for**

**return**  $G(m, \arg \max_{s \in S_T} V_T(s), 1)$

---

The input  $x$  consists of prompt decorator and frontier selection query prompt, and Algorithm 1 finally returns the selected frontier. We design a prompt decorator or several prompt decorators for each of the above four components to elicit reasoning in LLMs as below.

- Thought decomposition: *Imagine ten different experts are answering this question. They will brainstorm the answer step by step, reasoning carefully and taking all facts into consideration.*
- Thought generation: *All experts will write down one step of their thinking, then share it with the group. They will each critique their response, and the all the responses of others They will check their answer based on science and the laws of physics. Then all experts will go on to the next step and write down this step of their thinking. They will keep going through steps until they reach their conclusion taking into account the thoughts of the other experts. If at any time they realise that there is a flaw in their logic they will backtrack to where that flaw occurred. If any expert realises they are wrong at any point then they acknowledges this and start another train of thought.*
- Thought evaluation: *Each expert will assign a likelihood of their current assertion being correct.*

- Tree search: *Continue until the experts agree on a single most likely location.*

We append the above ToT prompt decorators with our frontier selection query prompt, *i.e.*, *pick one single location where a laptop is most likely to occur and give a final answer with one single location index*, and the location indices and objects extracted from the semantic navigation map  $M_t$  are formatted as: *location #<i>, located near <room type>, where <{object1, object2, ...}> are also found..* We feed them together into an LLM. The LLM returns a consensus about the most feasible frontier index to the goal object along with a numerical likelihood. The final output from LLM is formatted as: *Conclusion, location #<i> with highest likelihood [%]*. Therefore, our method pinpoints the most promising frontiers, effectively bridging the insights of an LLM with precision in frontier selection, thus enabling more informed and context-aware exploration.

**Goal Identification** This module determines whether the current object approached by an agent matches the target object specified in an instruction  $L$ . Our definition of the target object encompasses more intricate spatial and/or appearance descriptions of the object, rather than just object category as previous work [13, 43], such as: “Alarm clock on a dresser near a desk lamp, bed” or “Small, metallic alarm clock”. Thus, an algorithm that merely checks object category, *e.g.*, if the current object is an “alarm clock”, is insufficient. To make a more informed assessment of whether the scene’s context aligns with the target object description, we initially employ a vision language model to interpret the current scene and convert it into a language-based expression. Subsequently, we use a large language model, specifically GPT-3.5 [28] in our experiments, to analyze the textual descriptions of the target in the instructions  $L$  and the object currently observed in the scene. By integrating both textual and visual semantic information, our model achieves a deep semantic understanding of the environment, enhancing the accuracy of aligning scene context with the target description and thereby improving the results of L-ZSON.

## 4 Experiments

### 4.1 Environments and Datasets

We evaluate the performance of our L-ZSON approach based on ToT reasoning on two benchmarks, *i.e.*, PASTURE [13] and RoboTHOR [9].

*PASTURE* Introduced by Gadre *et al.* in CoW [13], *PASTURE* is characterized by its diverse set of environments, each presenting unique navigation challenges. For example, *PASTURE* introduces categories such as *uncommon* objects, objects with varying *appearance* complexities, objects placed in intricate *spaces*, and also *hidden* objects strategically obscured from plain sight. Designed mainly for L-ZSON tasks, *PASTURE* contains 2,520 validation episodes in 15 validation environments with 12 goal object categories. In the *PASTURE* dataset, agents are tested not only in their navigation skills but also in their adaptability and decision-making ability.

*RoboTHOR* Introduced by Deitke *et al.* [9], offers a platform for ZSON evaluation. Based on real-world indoor settings, RoboTHOR provides precise 3D representations of these environments, creating a more practical and genuine evaluation platform. This benchmark contains a diverse array of objects, set within familiar household and office spaces. It contains 15 validation environments with 12 goal object categories.

## 4.2 Metrics

Following the setting of [13, 43], we employ the Success Rate (SR) and Success Weighted by Path Length (SWPL) as our evaluation metrics. These metrics not only measure the agent’s ability to reach the goal objects, but also consider the efficiency and reliability of navigation. Specifically, the SR quantifies the proportion of episodes in which the agent successfully navigates to the goal object within maximum steps. Represented in percentage, a higher value suggests superior capability. Although SR provides a measure of success, it does not account for the efficiency of the agent’s navigation path. Therefore, the SWPL metric considers both the success of navigation and the optimality of the path taken. It penalizes unnecessary long paths, ensuring that the agent’s navigation is both correct and efficient.

## 4.3 Baselines

Our VLTNet is evaluated against the following state-of-the-art models for both ZSON and L-ZSON tasks.

**CoW** [13]: CoW targets both ZSON and L-ZSON tasks, using CLIP to consistently update a top-down map with image-to-goal relevance. Variants of CoW with different CLIP-like localization modules were also included: **CLIP-Ref** [13], **CLIP-Patch** [13], **CLIP-Grad** [13], **MDETR** [18], **OWL** [27].

**ESC**[43]: ESC utilizes GLIP for object detection to facilitate scene understanding and common sense reasoning. ESC also incorporates soft logic predicates to ensure optimal path and navigation decisions.

## 4.4 Results

Our experiments were designed rigorously to assess the efficacy of our proposed VLTNet for ZSON and L-ZSON tasks. We juxtaposed our method with the state-of-the-art approaches and the results are shown in Table 1.

On the PASTURE dataset, our VLTNet model consistently surpassed competing models across all metrics. Notably, within the *Appearance* category, our VLTNet model achieves a noteworthy success rate of 35.0%. In contrast, the OWL has an SR of 26.9%. Similarly, in the *Spatial* category, the SR of our VLTNet model is 33.3%, outperforming OWL model’s 19.4%. This demonstrates our model’s capability in understanding spatial relationships and interpreting complex object descriptions using the Tree of Thoughts Reasoning and Exploration

**Table 1.** Quantitative results on the PASTURE[13] and RoboTHOR[1] benchmarks are provided, comparing our VLTNet model with six CoW (CLIP on Wheel) variants designed for L-ZSON tasks, while ESC is exclusively used for ZSON tasks. Abbreviations used include Unc. for Uncommon, App. for Appearance, dist. for distract, and Hid. for Hidden. The best results are highlighted in red bold, while the second-best results are highlighted in blue bold.

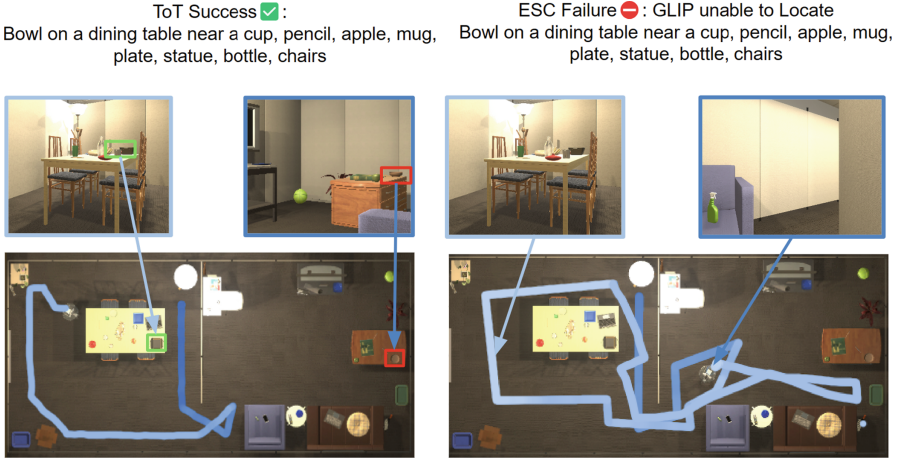
Models	PASTURE									RoboTHOR	
	Unc. App. Space			App. Space		Hid. dist.		Avg.		Avg.	
	SR	SR	SR	SR	SR	SR	SR	SWPL	SR	SWPL	SR
CLIP-Ref.	3.6	2.8	2.8	3.1	3.3	4.7	5.0	1.7	2.5	2.4	2.7
CLIP-Patch	18.1	13.3	13.3	10.8	10.8	17.5	<b>17.8</b>	9.0	14.2	10.6	20.3
CLIP-Grad.	16.1	11.9	11.7	9.7	10.3	14.4	16.1	9.2	12.9	9.7	15.2
MDETR	3.1	7.2	5.0	7.2	4.7	8.1	8.9	5.4	6.3	8.4	9.9
OWL	32.8	<b>26.9</b>	<b>19.4</b>	<b>19.4</b>	<b>16.1</b>	<b>19.2</b>	15.8	<b>12.6</b>	<b>21.1</b>	17.2	27.5
ESC	<b>35.5</b>	-	-	-	-	-	-	-	-	<b>22.2</b>	<b>38.1</b>
VLTNet	<b>36.9</b>	<b>35.0</b>	<b>33.3</b>	<b>21.9</b>	<b>21.7</b>	<b>22.8</b>	<b>26.4</b>	<b>14.0</b>	<b>28.2</b>	17.1	<b>33.2</b>

module. As shown in Fig. 3, our model successfully leverages an LLM to extract the candidate frontier of “bowl” and then the Goal Identification module verifies that the bowl aligns with the spatial cues in an instruction. Conversely, the ESC model is unable to locate the goal object even if the agent was facing the target. Also, it is essential to note that ESC is only designed for ZSON tasks and thus can only accept a single object category instruction and cannot directly handle object descriptions using natural language.

On the RoboTHOR dataset, the ESC model, tailored for RoboTHOR, secures an SR of 38.1%. Our VLTNet continues its commendable performance by achieving an SR of 33.2% and an SWPL of 17.1%, which outperforms CoW that secures an SR of 27.5%. This further proves that our VLTNet navigation model has a competitive performance compared to the state-of-the-art methods.

#### 4.5 Ablation Study

*The effect of ToT Reasoning and Exploration module.* To evaluate the efficacy of Tree of Thoughts Reasoning and Exploration module, we conducted a comparative analysis with two models on the PASTURE LONGTAIL dataset [13], consisting of 12 uncommon object goals. All models employ GPT-3.5, differing only in their input prompts. The first model is guided to directly select a frontier from all the available candidates, devoid of any explicit directive for reasoning. The second model uses ToT input prompts, which requires a deliberation between ten experts to articulate their reasoning and collectively determine



**Fig. 3.** Visualizing egocentric trajectories of VLTNet and ESC navigation process when given a spatial goal instruction. Color indicates trajectory progress, where blue indicating trajectory start and white indicating trajectory end. The goal objects are boxed in green, while distractors are boxed in red.

a frontier to select for exploration. As evidenced by Table 2, the model that uses ToT prompts for frontier selection exhibits a marked superiority over the model without ToT prompts. This underscores the efficacy of ToT prompting in facilitating the selection of frontier that are closer to the goal object.

**Comparison of different models for Goal Identification module.** To prove the robustness of using an LLM in the Goal Identification module, we tested this module using GPT-3.5 along with two other VLM models: ViLT [20] for visual question answering and GLIP [22] for object grounding. All three

**Table 2.** Performance between different prompting in ToT Reasoning and Exploration module on Pasture Uncom. split.

Reasoning Prompt	SWPLSR	
W/o ToT prompts	12.4	29.8
ToT prompts	<b>16.6</b>	<b>36.9</b>

**Table 3.** Performance between different models in Goal Identification module on Pasture Space dist. split.

Module	SWPLSR	
GLIP	5.9	12.6
ViLT	8.7	18.3
GPT-3.5	<b>9.3</b>	<b>21.7</b>

models are evaluated on the PASTURE Space dataset [13], in which target objects are embedded in spatially descriptive prompts. Table 3 illustrates that GLIP faces challenges in grounding objects when presented with intricate spatial cues. When the current frame is isolated and processed through VILT, there is a marginal improvement in object identification based on spatial hints. However, the most effective method for validating the goal object in accordance with a spatial prompt is GPT-3.5, by determining the congruence between objects present in the current scene and the provided spatial cues in an instruction.

## 5 CONCLUSIONS

In this paper, we introduce a VLTNet model, which harnesses both visual language modeling and ToT reasoning for L-ZSON task. We innovatively integrated the Tree of Thoughts reasoning framework, enriching the decision-making process with its nuanced multi-path reasoning capabilities. This empowers the model to make informed decisions during a frontier selection process in language-instructed navigation. The results on the PASTURE and RoboTHOR benchmarks demonstrate that our VLTNet excels in handling complex L-ZSON tasks that demand intricate understanding and interpretation of natural language instructions and environments.

## References

1. AI, A.I.f.: Key features, <https://ai2thor.allenai.org/robothor/>
2. Al-Halah, Z., Ramakrishnan, S.K., Grauman, K.: Zero experience required: Plug & play modular transfer learning for semantic visual navigation (Apr 2022), <https://arxiv.org/abs/2202.02440>
3. Bach, S.H., Broecheler, M., Huang, B., Getoor, L.: Hinge-loss markov random fields and probabilistic soft logic. *J. Mach. Learn. Res.* **18**(109), 1–67 (2017)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners (Jan 2020), [https://proceedings.neurips.cc/paper\\_files/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html)
5. Chang, M., Gupta, A., Gupta, S.: Semantic visual navigation by watching youtube videos (Jan 2020), <https://proceedings.neurips.cc/paper/2020/hash/2cd4e8a2ce081c3d7c32c3cde4312ef7-Abstract.html>
6. Chaplot, D.S., Gandhi, D., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural slam, [https://iclr.cc/virtual\\_2020/poster\\_HklXn1BKDH.html](https://iclr.cc/virtual_2020/poster_HklXn1BKDH.html)
7. Chattopadhyay, P., Hoffman, J., Mottaghi, R., Kembhavi, A.: Robustnav: Towards benchmarking robustness in embodied navigation (Jun 2021), <https://arxiv.org/abs/2106.04531>
8. Chen, P., Ji, D., Lin, K., Zeng, R., Li, T.H., Tan, M., Gan, C.: Weakly-supervised multi-granularity map learning for vision-and-language navigation (Oct 2022), <https://arxiv.org/abs/2210.07506>

9. Deitke, M., Han, W., Herrasti, A., Kembhavi, A., Kolve, E., Mottaghi, R., Salvador, J., Schwenk, D., VanderBilt, E., Wallingford, M., et al.: Robothor: An open simulation-to-real embodied ai platform (Apr 2020), <https://arxiv.org/abs/2004.06799>
10. Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Ehsani, K., Salvador, J., Han, W., Kolve, E., Kembhavi, A., Mottaghi, R.: Proctor: Large-scale embodied ai using procedural generation (Dec 2022)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding, <https://aclanthology.org/N19-1423/>
12. Dorbala, V.S., Sigurdsson, G., Piramuthu, R., Thomason, J., Sukhatme, G.S.: Clipnav: Using clip for zero-shot vision-and-language navigation (Nov 2022), <https://arxiv.org/abs/2211.16649>
13. Gadre, S.Y., Wortsman, M., Ilharco, G., Schmidt, L., Song, S.: Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023). <https://doi.org/10.1109/cvpr52729.2023.02219>
14. Gervet, T., Chintala, S., Batra, D., Malik, J., Chaplot, D.S.: Navigating to objects in the real world (Dec 2022), <https://arxiv.org/abs/2212.00922>
15. Gomez, C., Hernandez, A.C., Barber, R.: Topological frontier-based exploration and map-building using semantic information (Oct 2019), <https://www.mdpi.com/1424-8220/19/20/4595>
16. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. IEEE International Conference on Computer Vision (2017). <https://doi.org/10.1109/iccv.2017.322>
17. Huang, C., Mees, O., Zeng, A., Burgard, W.: Visual language maps for robot navigation. In: IEEE International Conference on Robotics and Automation. pp. 10608–10615. IEEE (2023)
18. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: Mdetr - modulated detection for end-to-end multi-modal understanding. IEEE/CVF International Conference on Computer Vision (2021)
19. Khandelwal, A., Weihs, L., Mottaghi, R., Kembhavi, A.: Simple but effective: Clip embeddings for embodied ai. IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022). <https://doi.org/10.1109/cvpr52688.2022.01441>
20. Kim, W., Son, B., Kim, I.: Vilt: Vision-and-language transformer without convolution or region supervision (Jun 2021), <https://arxiv.org/abs/2102.03334>
21. Leong, K.: Reinforcement learning with frontier-based exploration via autonomous environment (Jul 2023), <https://arxiv.org/abs/2307.07296>
22. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022). <https://doi.org/10.1109/cvpr52688.2022.01069>
23. Majumdar, A., Aggarwal, G., Devnani, B., Hoffman, J., Batra, D.: Zson: Zero-shot object-goal navigation using multimodal goal embeddings (Jun 2022)
24. Maksymets, O., Cartillier, V., Gokaslan, A., Wijmans, E., Galuba, W., Lee, S., Batra, D.: Thda: Treasure hunt data augmentation for semantic navigation. IEEE/CVF International Conference on Computer Vision (2021). <https://doi.org/10.1109/iccv48922.2021.01509>
25. Mezghan, L., Sukhbaatar, S., Lavril, T., Maksymets, O., Batra, D., Bojanowski, P., Alahari, K.: Memory-augmented reinforcement learning for image-goal navigation. IEEE/RSJ International Conference on Intelligent Robots and Systems (2022). <https://doi.org/10.1109/iros47612.2022.9981090>

26. Min, S.Y., Chaplot, D.S., Ravikumar, P.K., Bisk, Y., Salakhutdinov, R.: Film: Following instructions in language with modular methods (Sep 2023), <https://openreview.net/forum?id=qI4542Y2s1D>
27. Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., et al.: Simple open-vocabulary object detection with vision transformers (Jul 2022), <https://arxiv.org/abs/2205.06230>
28. OpenAI: Gpt-3.5 technical report (2023)
29. Park, J., Yoon, T., Hong, J., Yu, Y., Pan, M., Choi, S.: Zero-shot active visual search (zavis): Intelligent object search for robotic assistants. IEEE International Conference on Robotics and Automation (2023). <https://doi.org/10.1109/icra48891.2023.10161345>
30. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations (Mar 2018), <https://arxiv.org/abs/1802.05365>
31. 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 (Feb 2021), <https://arxiv.org/abs/2103.00020>
32. Rae, J.W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al.: Scaling language models: Methods, analysis & insights from training gopher (Jan 2022), <https://arxiv.org/abs/2112.11446>
33. Ramrakhya, R., Undersander, E., Batra, D., Das, A.: Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022). <https://doi.org/10.1109/cvpr52688.2022.00511>
34. Verbiest, K., Berrabah, S.A., Colon, E.: Autonomous frontier based exploration for mobile robots (Jan 2015), [https://link.springer.com/chapter/10.1007/978-3-319-22873-0\\_1](https://link.springer.com/chapter/10.1007/978-3-319-22873-0_1)
35. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models (Jan 2023), <https://arxiv.org/abs/2201.11903>
36. Yamauchi, B.: A frontier-based approach for autonomous exploration. Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation. <https://doi.org/10.1109/cira.1997.613851>
37. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models (May 2023), <https://arxiv.org/abs/2305.10601>
38. Ye, J., Batra, D., Das, A., Wijmans, E.: Auxiliary tasks and exploration enable objectgoal navigation. IEEE/CVF International Conference on Computer Vision (2021). <https://doi.org/10.1109/iccv48922.2021.01581>
39. Yu, B., Kasaei, H., Cao, M.: Frontier semantic exploration for visual target navigation. IEEE International Conference on Robotics and Automation (2023). <https://doi.org/10.1109/icra48891.2023.10161059>
40. Yu, B., Kasaei, H., Cao, M.: L3mvm: Leveraging large language models for visual target navigation (Apr 2023), <https://arxiv.org/abs/2304.05501>
41. Zhao, Q., Zhang, L., He, B., Qiao, H., Liu, Z.: Zero-shot object goal visual navigation. IEEE International Conference on Robotics and Automation (2023). <https://doi.org/10.1109/icra48891.2023.10161289>



42. Zheng, K., Zhou, K., Gu, J., Fan, Y., Wang, J., Di, Z., He, X., Wang, X.E.: Jarvis: A neuro-symbolic commonsense reasoning framework for conversational embodied agents (Sep 2022), <https://arxiv.org/abs/2208.13266>
43. Zhou, K., Zheng, K., Pryor, C., Shen, Y., Jin, H., Getoor, L., Wang, X.E.: Esc: Exploration with soft commonsense constraints for zero-shot object navigation (Jul 2023), <https://arxiv.org/abs/2301.13166>
44. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven visual navigation in indoor scenes using deep reinforcement learning. IEEE International Conference on Robotics and Automation (2017). <https://doi.org/10.1109/icra.2017.7989381>



# Few-Shot Deep Structure-Based Camera Localization with Pose Augmentation

Cheng-Yu Tsai and Shang-Hong Lai<sup>(✉)</sup>

National Tsing Hua University, Hsinchu, Taiwan  
lai@cs.nthu.edu.tw

**Abstract.** Camera localization predicts the camera pose from a query image. There are two types of deep learning-based camera localization methods: image-based and structure-based. Previous works have shown that data augmentation can improve the performance of image-based methods, but there are no research studies on the structure-based method with data augmentation technique. In this paper, we propose a new pose augmentation procedure that can further improve the performance of the deep structure-based camera localization method, especially under few-shot settings. We investigate different inpainting and rendering strategies and compare their performance with pose augmentation. In addition, we propose a confidence-based sampling scheme that drastically reduces the computation time while maintaining high pose estimation accuracy.

**Keywords:** Camera localization · deep learning · pose augmentation.

## 1 Introduction

Camera localization is to estimate the 6-DoF camera pose, including 3D position and orientation, from an image in a known environment. Traditional methods use feature descriptors [1–3] to establish 2D-3D correspondences between the key points on the 2D image and the 3D model generated by a SfM system [4, 5]. These correspondences can then be used to compute the camera pose of the query image. However, these methods are computationally expensive and suffer from textureless scenes, repetitive patterns, duplicated objects, and highly symmetrical indoor scenes.

Various CNN-based methods have been proposed to take advantage of the strong learning capability of CNN in recent years. They can be divided into two categories: image-based and structure-based. Image-based methods [10, 15] regress poses from images. Structure-based methods [20–22, 24, 25] establish 2D-3D correspondences by using CNN, and then compute the 6-DoF camera pose by solving the Perspective-n-Point (PnP) [20, 27] problem. Structure-based methods typically outperform image-based methods.

To make the most of the training data, previous works [6–8] proposed different methods to augment camera poses. They [6–8] proved that augmented image-pose pairs can improve the performance of camera localization models. However, these augmentation methods have only been applied to image-based models. In

this paper, we aim to extend the data augmentation strategy to improve the structure-based camera localization.

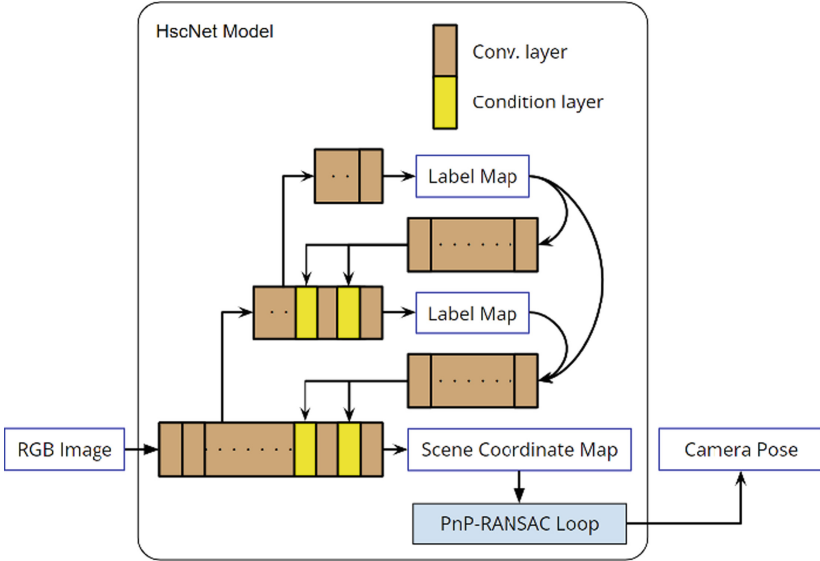
Most camera localization models, either image-based or structure-based, need to be trained on specific datasets for each scene of application interest. Previous models were usually trained on thousands of image-pose data samples. In the real world, collecting fine, dense datasets for model training may be hard. Thus, few-shot learning is a more practical setting for real-world applications of camera localization. Accordingly, data augmentation can make a limited number of data samples cover more scene information, helping models learn better to predict more accurate camera poses from query images.

In this work, we focus on applying data augmentation to improve structure-based camera localization in few-shot situations. In this work, we propose a new data augmentation method to handle invalid pixels and combine our data augmentation method with a state-of-the-art structure-based model: Hierarchical Scene Coordinate Classification and Regression (HscNet) [24]. The improved method can achieve improvements in camera localization accuracy that is superior to the state-of-the-art models. In addition to the image-based methods, we prove that additional image-pose pairs generated by synthesizing images from spatially-augmented camera poses can still improve the performance of the structure-based localization method (HscNet [24]). We also prove that our augmentation method performs satisfactorily under few-shot situations for the camera localization problem. We also propose a confidence-based sampling scheme to improve the quality of the candidate point set for solving the PnP problem that computes the final query camera pose. This improvement speeds up the prediction while maintaining low prediction error and high accuracy for camera localization.

## 2 Related Work

Deep learning based camera localization methods can roughly be divided into image-based and structure-based localization approaches. Image-based methods [10, 15] feed the query images into CNN models, and models output regressed camera poses.

Instead of regressing the camera pose, structure-based methods [19–26] regress each 2D pixel on the query image into 3D scene coordinates to obtain 2D-3D correspondences. Then the camera pose prediction task becomes a PnP [27] problem and can be solved by the above 2D-3D correspondences. After predicting scene coordinates, DSAC [19] samples minimal sets of four scene coordinates to create a pool of hypotheses. It uses another CNN model to score the reprojection errors and selects the best hypothesis as the final predicted pose. DSAC++ [20] improves the scoring model in DSAC [19], and applies a PnP-RANSAC algorithm that puts the PnP solver into a RANSAC [28, 29] loop. The RANSAC process filters outliers in the predicted coordinates to reduce the noise of incorrectly predicted coordinates. The recent structure-based methods [20, 22–26] compute the camera pose of the query image from the predicted scene coordinates by the RANSAC-PnP step. NeuMap [26] decomposes the scene information



**Fig. 1.** HscNet [24] network architecture and the pose estimation procedure.

into scene-agnostic key points and scene-specific latent code. The scene-agnostic auto-transdecoder regresses the sparse 3D scene coordinates for the PnP algorithm by the cross-attention between the robust features and the scene-specific latent codes.

The training objective of deep models in structure-based methods is the regression of 3D scene coordinates. Therefore, it requires depth information to establish the ground truth of the 3D scene coordinates. Due to the precise 3D information during training, structure-based methods perform better than image-based methods. Thus we choose the structure-based approach as our research focus.

HscNet [24] is a state-of-the-art structure-based model for camera localization. Its model architecture is depicted in Figure 1. HscNet [24] clusters all 3D points of training scenes into hierarchical classes with hierarchical k-means clustering and sets clusters as class labels during pre-processing. The model hierarchically classifies labels for each 2D pixel and predicts the closest cluster center to each 2D pixel. Finally, HscNet [24] computes the predicted camera pose by the PnP-RANSAC algorithm.

Dong *et al.* [30] first propose the few-shot problem setting for the visual localization task. Few-shot learning, or low-shot learning, is a learning problem where only a small amount of data is available for learning. Dong *et al.* [30] samples the training data of camera localization datasets by fixed steps to form the new few-shot sets for training. We follow this uniform sampling strategy to create few-shot training sets but still use complete test sets for evaluation.

### 3 Proposed Method

#### 3.1 Structure-Based Camera Localization Model

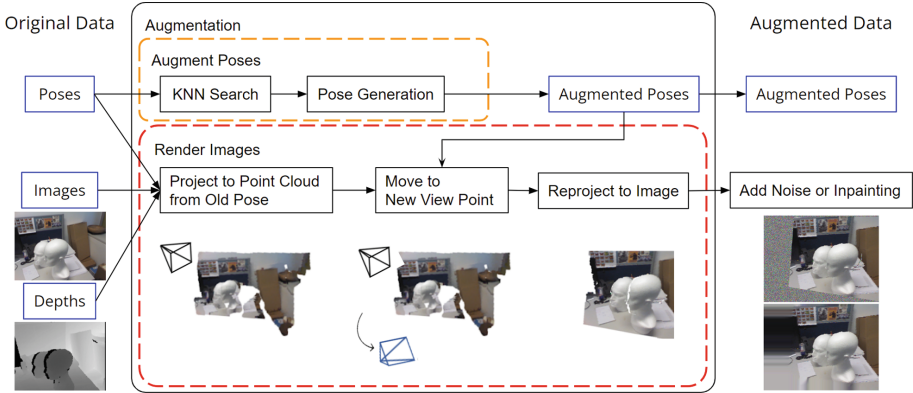
In this work, we propose to apply data augmentation for structure-based camera localization so that the CNN model can benefit from augmented training data with larger pose variations for better generalization. Especially in the few-shot situation, the spatially augmented camera poses and synthesized images can fill in more scene information to support the training process. We use the inpainting technique to deal with the blank area caused by the invalid depth and pose changes. In addition to the new RGB images, the rendering pipeline simultaneously generates new depth maps. This allows us to use the extended 3D information to improve structure-based methods. Since our spatially-augmented data do not fit the original label maps provided by HscNet [24], we use Nearest Neighbor search to assign the nearest cluster label to the 3D point of each pixel, and generate the new label maps for the augmented images as the ground truth.

The new camera poses, synthesized RGB images & depth maps, and the new label maps form additional augmented image-pose pairs, which is  $m$  times the number of original data. After the pre-processing and data preparation, we train and evaluate HscNet [24] with the augmented data to verify whether the structure-based localization can benefit from the data augmentation method that increases the number of image-pose pairs.

HscNet[24] provides point correspondences for the PnP-RANSAC algorithm to estimate the 3D camera localization. The purpose of the pose augmentation is to improve the accuracy of point correspondences generated by HscNet, so that the final pose estimation by using PnP-RANSAC is more accurate. In this work, we use the traditional RANSAC for estimating the 3D camera pose estimation. Our main contribution is focused on improving the training of HscNet for generating more accurate point correspondences to be used as input to PnP-RANSAC [28, 29] for more accurate 3D camera localization. It can also be combined with a more robust RANSAC algorithm to achieve more accurate pose estimation.

#### 3.2 Data Augmentation

We use the following augmentation pipeline to generate the augmented data. It starts by generating new camera poses. It first applies the K-Nearest-Neighbor search to find the  $k$  nearest camera poses, and dynamically decides the sampling ranges for camera poses in the three axes and three Euler angle directions according to the maximum and minimum values for each pose element from the  $k$  neighbors. After finding the bounds in the six dimensions, it randomly adjusts the camera poses by independently sampling each component with uniform distribution within the bounds to create additional poses. It augments an image for  $m$  times, so augmented images is  $m$  times the number of original images. The rendering pipeline projects the pixels of the 2D images onto the 3D point cloud and reprojects them back to 2D images according to new camera poses. We set



**Fig. 2.** Flowchart of the proposed data augmentation pipeline.

$k$  to 50 in the K-Nearest-Neighbors Search, but we lower it to fit the smaller and sparser data in the few-shot situation (Fig. 2).

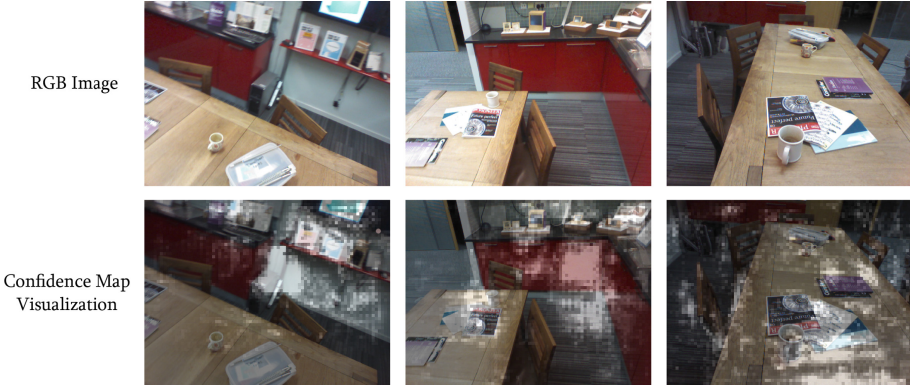
We observe that different point size settings of the point cloud in the rendering pipeline affect the effect of augmentation. The point size setting affects how large the 3D points are projected onto 2D images. When the point size is larger, the rendered images look coarser and less accurate, but there are fewer invalid pixels because the larger point sizes cover more space. On the other hand, the smaller point size produces more accurate and precise images, but they suffer from more blank areas. Even though smaller point sizes cause the blank area problem, we choose a smaller point size for rendering and use the RGB inpainting technique to cover these blank areas and make the images rendered under smaller point sizes more photo-realistic than the coarser ones. In this work, we apply the inpainting function with the Navier-Stokes based algorithm from the OpenCV Python toolkit to fill in the blanks on the augmented RGB images.

### 3.3 Confidence-based Sampling

The FPS bottleneck of HscNet [24] is the first part in the PnP-RANSAC algorithm that samples four points to form a valid hypothesis. It spends much time forming a valid hypothesis because it sometimes picks bad points and has to resample. We plan to improve the quality of candidate points so that it does not have to spend too much time resampling.

HscNet [24] uses the one-hot encoding format for the ground truth of two hierarchical label maps. In addition to HscNet [24] using the “*argmax*” function to extract the predicted classes, we also edit the model to output the “*max*” value for each pixel. This raw value indicates how strongly the model considers the pixel the predicted label. We take this raw value as confidence and create a confidence map. We select the top 50% points with higher confidence as the sample range and filter out the other 50% points with lower confidence. The proposed confidence-based sampling scheme samples the points with higher quality

for the PnP-RANSAC algorithm and has a higher probability of successfully forming a valid hypothesis, thus making the PnP-RANSAC algorithm more efficient. Because the second classification network performs poorly in some scenes, we only use the raw value as confidence in the first classification network. Fig. 3 provides visualization of the confidence map.



**Fig. 3.** Visualization of the confidence maps for the corresponding RGB images. In the second row, the lighter areas mean they have higher confidence than other darker areas.

## 4 Experiments And Discussion

### 4.1 Datasets and Experimental Setup

We use two standard visual localization benchmark datasets, 7-Scenes & 12-Scenes, to evaluate the proposed method. 7-Scenes [32] dataset is an indoor RGB-D dataset with seven scenes. It is widely used for visual localization and SLAM research. It uses Microsoft Kinect V1 to capture RGB images and depth maps, then applies the KinectFusion system to obtain the ground truth camera pose. 12-Scenes [33] is also an indoor RGB-D dataset. It contains four bigger scenes, in total, twelve smaller sub-scenes. It uses an iPad color camera with a Structure.io depth sensor to capture RGB-D images and applies a global-bundle adjustment algorithm to obtain the ground truth camera pose. Fig. 4 depicts some sample images from these two datasets.

Under the few-shot condition, we compare our experimental results with our baseline HscNet [24], and Dong et al. [30] on 7-Scenes. Since Dong et al. [30] did not report experimental results on 12-Scenes, we only compare our method with the baseline HscNet [24] on 12-Scenes.



**Fig. 4.** Some sample images selected from different scenes in the 7-Scenes and 12-Scenes datasets are depicted in the upper and lower rows, respectively.

**Table 1.** Comparison of the experimental results on 7-Scenes with 100% data with different methods. DSM[25] has two prediction modes, Single (frame) and Video (frames), while other methods predict from a single image. SANet [22] only report the total average of accuracy. The numbers are median translation (in meters) and rotation errors (in degrees). Here we set  $k$  to 50 and  $m$  to 32.

7-Scenes 100%	DSAC++[20]		SANet[22]		DSM(Single)[25]		DSM(Video)[25]		HscNet [24]		Ours	
	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.
Chess	97.1	<b>0.02, 0.50</b>	-	0.03, 0.88	94.5	<b>0.02, 0.71</b>	96.1	<b>0.02, 0.68</b>	97.5	<b>0.02, 0.70</b>	<b>97.7</b>	<b>0.02, 0.59</b>
Fire	89.6	<b>0.02, 0.90</b>	-	0.03, 1.08	93.8	<b>0.02, 0.86</b>	94.5	<b>0.02, 0.80</b>	<b>96.7</b>	<b>0.02, 0.90</b>	96.2	<b>0.02, 0.84</b>
Heads	92.4	<b>0.01, 0.80</b>	-	0.02, 1.48	96.4	<b>0.01, 0.85</b>	99.5	<b>0.01, 0.80</b>	<b>100</b>	<b>0.01, 0.90</b>	<b>100</b>	<b>0.01, 0.82</b>
Office	86.6	0.03, <b>0.70</b>	-	0.03, 1.00	82.3	0.03, 0.84	84.2	0.03, 0.78	86.5	0.03, 0.80	<b>87.2</b>	<b>0.02, 0.72</b>
Pumpkin	59.0	<b>0.04, 1.10</b>	-	0.05, 1.32	57.0	<b>0.04, 1.16</b>	57.2	<b>0.04, 1.11</b>	59.9	<b>0.04, 1.00</b>	<b>60.3</b>	<b>0.04, 1.03</b>
Kitchen	66.6	0.04, <b>1.10</b>	-	0.04, 1.40	68.7	0.04, 1.17	<b>69.2</b>	<b>0.03, 1.12</b>	65.5	0.04, 1.20	63.2	0.04, 1.17
Stairs	29.3	0.09, 2.60	-	0.16, 4.59	53.9	0.05, 1.36	69.9	0.04, 1.16	87.5	<b>0.03, 0.80</b>	<b>87.7</b>	<b>0.03, 0.73</b>
Average	76.1	0.04, 1.10	68.2	0.05, 1.68	78.1	<b>0.03, 0.99</b>	81.6	<b>0.03, 0.92</b>	<b>84.8</b>	<b>0.03, 0.90</b>	84.3	<b>0.03, 0.84</b>

**Table 2.** Experimental results on 7-Scenes dataset under few-shot settings.

7-Scenes Few-shot	HLoc [34,35]		DSAC* [21]		Dong et al [30]		HscNet [24]		Ours	
	Median err.	Median err.	Median err.	Median err.	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.
Chess (0.5%)	0.04, 1.42	<b>0.03, 1.16</b>	0.04, 1.23	77.9	<b>0.03, 1.13</b>	77.0	<b>0.03, 0.99</b>			
Fire (0.5%)	<b>0.04, 1.72</b>	0.05, 1.89	<b>0.04, 1.52</b>	56.9	<b>0.04, 1.50</b>	62.2	<b>0.04, 1.30</b>			
Heads (1%)	0.04, 1.59	0.04, 2.71	<b>0.02, 1.56</b>	63.9	0.04, 2.16	74.6	<b>0.02, 1.58</b>			
Office (0.5%)	<b>0.05, 1.47</b>	0.09, 2.21	<b>0.05, 1.47</b>	40.0	0.06, 1.61	49.9	<b>0.05, 1.28</b>			
Pumpkin (0.5%)	0.08, 1.70	0.07, 1.68	0.07, 1.75	30.3	0.07, 1.65	33.5	<b>0.06, 1.56</b>			
Kitchen (0.5%)	0.07, 1.89	0.07, 2.02	<b>0.06, 1.93</b>	27.1	0.07, 2.09	39.4	<b>0.06, 1.81</b>			
Stairs (1%)	0.10, 2.21	0.18, 4.80	<b>0.05, 1.47</b>	26.6	0.10, 2.76	39.3	0.06, 1.61			
Average	0.06, 1.71	0.08, 2.35	<b>0.05, 1.56</b>	46.1	0.06, 1.84	53.7	<b>0.05, 1.45</b>			



## 4.2 Experimental Comparison

Table 1 summarizes the results on 7-Scenes with 100% data. All methods are state-of-the-art structure-based methods. HscNet [24] is our baseline that does not use augmentation data. The rightmost column is HscNet [24] combined with our data augmentation method. It shows that the additional augmented image-pose pairs can also improve the performance of the structure-based method (HscNet [24]). It makes HscNet [24] perform better and achieve the best results.

Table 2 demonstrates the experimental results on 7-Scenes under few-shot conditions. We obtain competitive results on median error and accuracy rate. Table 3 shows the experimental comparison on 12-Scenes, under the 1% and 0.5% few-shot conditions. The proposed method can achieve over 50% improvement in median translation and rotation errors and over 30% improvement in accuracy rate. The few-shot conditions on 7-Scenes follow the setting from Dong et al. [30]. The numbers are median translation and rotation errors (m, °), and the percentages of test images accurately predicted (error < 0.05 m, 5°). The results of HLoc [34,35], DSAC\* [21], and Dong et al. [30] are copied from [30].

## 4.3 RGB Inpainting and Noise

Table 4 shows the comparison of RGB noise and RGB inpainting for filling empty areas on augmented images. Both RGB noise and RGB inpainting can improve the performance of HscNet [24], but RGB inpainting is better than RGB noise.

**Table 3.** Experimental results on 12-Scenes datasets under few-shot settings.

12-Scenes	Few-shot 1%				Few-shot 0.5%			
	HscNet [24]		Ours		HscNet [24]		Ours	
	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.
Average	55.3	0.060, 2.4	<b>76.4</b>	<b>0.028, 1.2</b>	39.4	0.259, 16.8	<b>51.3</b>	<b>0.180, 15.5</b>

**Table 4.** Experimental results on 7-Scenes and 12-Scenes datasets under few-shot settings. Here we set  $k$  to 4 and  $m$  to 64 on 7-Scenes, and  $k$  to 2 and  $m$  to 32 on 12-Scenes.

Few-ShotAverage	HscNet [24]		w/ aug + Noise		w/ aug + Inpainting	
	Acc.	Media err.	Acc.	Media err.	Acc.	Media err.
7S	46.1	0.060, 1.8	51.2	0.050, 1.6	<b>53.7</b>	<b>0.047, 1.5</b>
12S 1%	55.3	0.060, 2.4	75.0	0.029, <b>1.2</b>	<b>76.4</b>	<b>0.028, 1.2</b>
12S 0.5%	39.4	0.259, 16.8	50.3	0.182, 15.6	<b>51.3</b>	<b>0.180, 15.5</b>

#### 4.4 Rendering Quality

Although the images with smaller point sizes are more accurate and precise, the larger blank regions negatively impact model training. Fortunately, the RGB inpainting technique can fill in the blanks and improve the benefits of augmented data. Adding RGB inpainting to the blank pixels can improve the training results, especially for images with smaller point size.

Table 5 compares the results of different point size settings and invalid pixel fixing strategies (RGB inpainting/RGB noise) when rendering augmented images. The augmented images with rendering quality “point size = 2.0” perform best. So we set the point size as 2.0. Table 5 also shows that adding RGB inpainting is better than adding RGB noise to augmented images with different rendering qualities (point size: 2.0/3.0).

**Table 5.** Ablation study of different rendering quality (point size settings) and different invalid pixel inpainting strategies on 7-Scenes under few-shot conditions. Here we set  $k$  as 2 and  $m$  as 32. The top table is augmented images with RGB inpainting, and the bottom is augmented images with RGB noise.

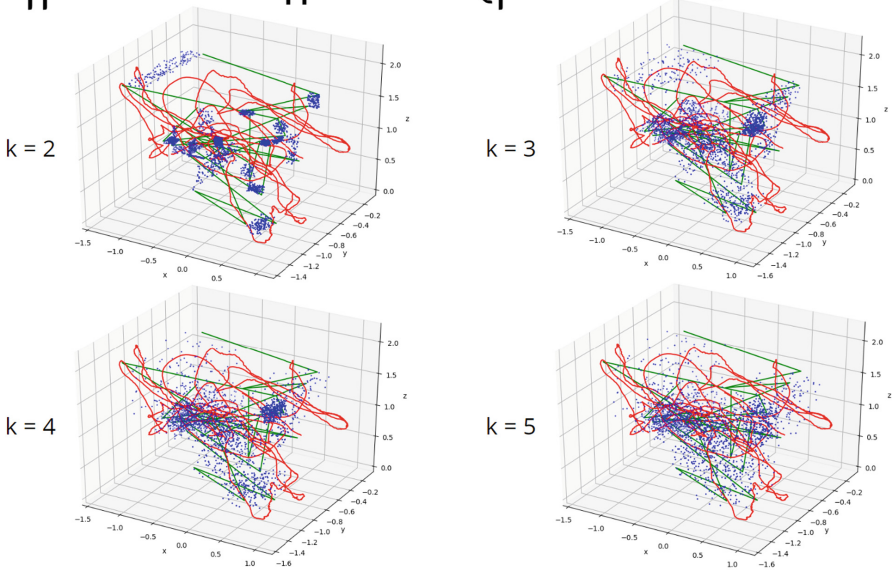
7-Scenes Few-shot Average	Point Size = 3.0		Point Size = 2.0		Point Size = 1.5	
	w/ RGB Inpainting					
	Acc.	Media err.	Acc.	Media err.	Acc.	Media err.
	48.7	0.06, 1.66	<b>49.5</b>	<b>0.05, 1.62</b>	47.9	0.06, 1.67
w/ RGB Noise						
	Acc.	Media err.	Acc.	Media err.	Acc.	Media err.
	44.8	0.06, 1.81	44.6	0.07, 1.84	44.1	0.07, 1.83

#### 4.5 Discussion on $k$ and $m$

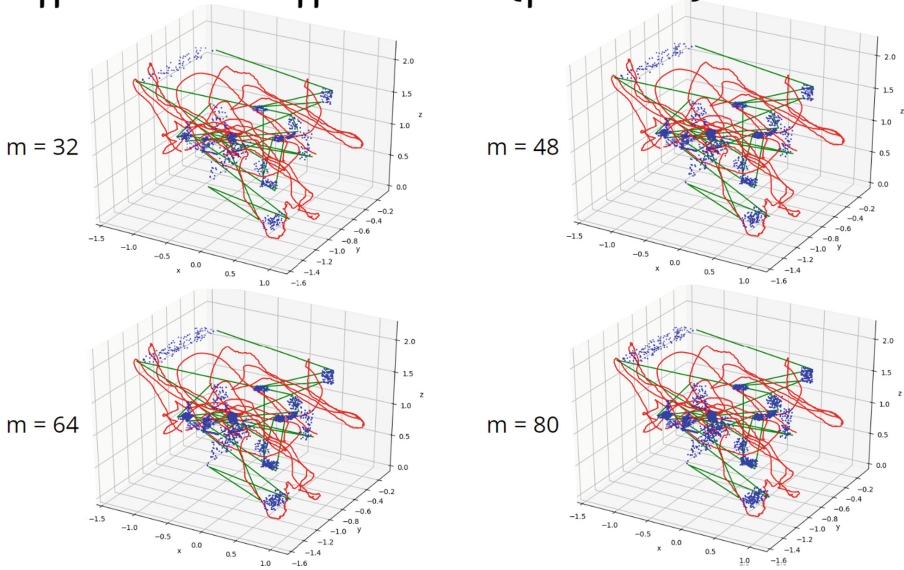
We analyze the influence of different  $k$  and  $m$  settings. The parameter  $k$  for K-Nearest-Neighbors Search affects the distribution of augmented data. The parameter  $m$  for augmentation multiple affects the density of augmented data. We test different  $k$  and  $m$  to find the best parameters for data augmentation. We first adjust the parameter  $m$  and find that  $m = 64$  is the best. And we adjust the parameter  $k$  with fixed  $m = 64$ , and find that  $k = 4$  is the best, as shown in Table 6. We use this fixed  $k/m$  for the few-shot experiments on 7-Scenes dataset.

**$k$  for K-Nearest-Neighbors Search:** The parameter  $k$  for K-Nearest-Neighbors Search affects the distribution of augmented data. Suppose  $k$  is too large or too small. In that case, the augmented camera pose can not benefit sufficiently from the KNN process, which dynamically decides the range of random new poses according to the dataset’s attribute. At the top of Fig. 5, the figure shows the distribution of augmented data (blue dots) with different values of

# Office with Different $k$ (fix $m = 64$ )



# Office with Different $m$ (fix $k = 2$ )



**Fig. 5.** These figures show the distribution and density of augmented data in the 7-Scenes dataset. The blue dots are augmented camera poses, the red lines are testing sequences, and the green lines are few-shot training sequences without augmentation. At the top, the figure shows the distribution with different values of  $k$ . At the bottom, the figure shows the density of augmented data with different values of  $m$ .

$k$ . Two endpoints of green lines restrict a group of blue dots in the upper left corner while  $k$  is 2 or 3. After  $k$  becomes larger, the KNN process can find the appropriate random range by the larger scale with more neighbors, and the new augmented poses spread wider.

**$m$  for Multiple of Augmentation:** The parameter  $m$  for augmentation multiple affects the amount and density of augmented data. The bottom of Fig. 5 shows the density of augmented data (blue dots).

**Table 6.** Ablation study of different  $m$  and different  $k$  values on 7-Scenes under few-shot conditions.

7-ScenesFew-shot	$m = 48, k = 2$		$m = 64, k = 2$		$m = 80, k = 2$	
	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.
Average	49.6	<b>0.05</b> , 1.62	<b>51.1</b>	<b>0.05</b> , <b>1.58</b>	50.6	<b>0.05</b> , 1.62
7-ScenesFew-shot	$m = 64, k = 2$		$m = 64, k = 3$		$m = 64, k = 4$	
	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.
Average	51.1	<b>0.05</b> , 1.58	<b>53.8</b>	<b>0.05</b> , 1.46	53.7	<b>0.05</b> , <b>1.45</b>

We compare our results of the previous fixed  $k/m$  and the new dynamic  $k/m$  with Dong et al [30]. Since the dynamic selection of  $k$  and  $m$  is not yet mature enough to be automatically selected by some mechanisms, we show the results of dynamic  $k/m$  only in this section. For the few-shot problem settings on 7-Scenes, our results with dynamic  $k/m$  achieve the best. An important future work is how to decide the values of  $k$  and  $m$  automatically and dynamically according to the datasets' attributes (Table 7).

#### 4.6 Confidence-based Sampling

In this section, we show the benefits of applying the proposed confidence-based sampling to the model of HscNet [24]. We evaluate the reference time and FPS performance on NVIDIA RTX 3090 GPU and Intel i9-12900KS CPU.

Table 8 shows that after applying the proposed confidence-based sampling, the model's FPS performance is improved by about 55% on the 7-Scenes dataset and 60% on the 12-Scenes dataset under the few-shot condition. At the same time, the accuracy (recall) and the median translation and rotation errors are about the same.

The confidence-based sampling significantly reduces the reference time while maintaining high accuracy (recall) and low translation and rotation error. The points with higher confidence bring a higher probability of successfully solving the coarse camera poses during the first part in the PnP-RANSAC algorithm. After reducing the number of failed hypothesis generation cases, the FPS for our method is increased by 55% ~60%.

**Table 7.** This table shows the results on the 7-Scenes dataset under the few-shot conditions following the setting from Dong *et al.* [30]. The numbers are median translation and rotation errors (m,  $\circ$ ), and the percentages of test images accurately predicted (error  $< 0.05$  m,  $5^\circ$ ). The third column, labeled “Fixed  $k/m$ ”, means that we use the fixed  $k = 4$  and  $m = 64$ , which is the result reported in this section. The fourth column labeled “Dynamic  $k/m$ ” means that we use the optimal  $k$  and  $m$  from the tables above, which are shown in this subsection. For consistency, we use the results with fixed  $k/m$  to compare with other research results. We compare our results with dynamic  $k/m$  with other research results in this table only.

7-Scenes Few-shot	Dong <i>et al.</i> [30]	HscNet [24]		Ours: Fixed $k/m$		Ours: Dynamic $k/m$	
	Median err.	Acc.	Median err.	Acc.	Median err.	Acc.	Median err.
Chess	0.04, 1.23	77.9	<b>0.03</b> , 1.13	77.0	<b>0.03</b> , 0.99	<b>78.6</b>	<b>0.03</b> , <b>0.92</b>
Fire	<b>0.04</b> , 1.52	56.9	<b>0.04</b> , 1.50	<b>62.2</b>	<b>0.04</b> , 1.30	61.2	<b>0.04</b> , <b>1.29</b>
Heads	<b>0.02</b> , <b>1.56</b>	63.9	0.04, 2.16	<b>74.6</b>	<b>0.02</b> , <b>1.58</b>	<b>74.6</b>	<b>0.02</b> , 1.58
Office	<b>0.05</b> , 1.47	40.0	0.06, 1.61	49.9	<b>0.05</b> , 1.28	<b>50.5</b>	<b>0.05</b> , <b>1.24</b>
Pumpkin	0.07, 1.75	30.3	0.07, 1.65	<b>33.5</b>	<b>0.06</b> , 1.56	33.2	<b>0.06</b> , <b>1.51</b>
Kitchen	<b>0.06</b> , 1.93	27.1	0.07, 2.09	<b>39.4</b>	<b>0.06</b> , <b>1.81</b>	<b>39.4</b>	<b>0.06</b> , <b>1.81</b>
Stairs	<b>0.05</b> , 1.47	26.6	0.10, 2.76	39.3	0.06, 1.61	<b>44.0</b>	0.06, <b>1.46</b>
Average	<b>0.05</b> , 1.56	46.1	0.06, 1.84	53.7	<b>0.05</b> , 1.45	<b>54.5</b>	<b>0.05</b> , <b>1.40</b>

**Table 8.** Comparison of camera localization experiment results with and without using the proposed confidence-based sampling on 7-Scenes and 12-Scenes datasets under few-shot conditions. On 12-Scenes, as we consider the results of Kitchen-2 and Living-2 under the 0.5% condition to be outliers, we exclude them and calculate a new average result (Avg.\*) for the 0.5% condition.

7-Scenes Few-shot		Acc.	Median err.	Time(s)	FPS
Avg.	w/ aug	53.7	0.047, 1.45	0.25634	3.90
	w/ aug + cfd.	54.1	0.047, 1.44	0.16568	6.04
12-Scenes Few-shot		Acc.	Median err.	Time(s)	FPS
Avg.(1%)	w/ aug	76.4	0.028, 1.15	0.26413	3.79
	w/ aug + cfd.	75.9	0.029, 1.17	0.16409	6.09
Avg.(0.5%)	w/ aug	51.3	0.180, 15.50	0.42770	2.34
	w/ aug + cfd.	51.8	0.170, 14.43	0.26566	3.76
Avg.*(0.5%)	w/ aug	54.9	0.055, 2.23	0.40210	2.49
	w/ aug + cfd.	55.4	0.053, 2.23	0.24459	4.09

## 5 Conclusion

In this paper, we presented the pose augmentation strategy for the structure-based camera localization method. We combined the proposed data augmentation method with the state-of-the-art structure-based model HscNet [24], and

prove that the augmented image-pose pairs can further improve the performance of the structure-based model. Furthermore, our augmentation method can provide reasonable model training results under few-shot settings for the structure-based camera localization model. In addition, we propose a confidence-based sampling scheme for the structure-based camera localization model, which brings about 40% reduction in the inference time. Meanwhile, it maintains high accuracy in the camera localization results.

The proposed data augmentation method works well for indoor scenes, but it may not work well for outdoor scenes. Our augmentation method relies on high-quality depth information to reproject the pixels for image rendering at different camera poses. Indoor datasets usually contain deep maps with good quality for the novel view generation, such as 7-Scenes and 12-Scenes. In contrast, Outdoor datasets usually have sparse depth data in large space, making it difficult to generate novel views for scenes with large depth variations. Extension of the proposed data augmentation method to outdoor scenes is a topic worthy of further research.

## References

1. D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 1150-1157 vol.2, <https://doi.org/10.1109/ICCV.1999.790410>.
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). *Computer Vision & Image Understanding* **110**(3), 346–359 (2008)
3. E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," In International Conference on Computer Vision, 2012
4. J. L. Schönberger and J. -M. Frahm, "Structure-from-Motion Revisited," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 4104-4113, <https://doi.org/10.1109/CVPR.2016.445>.
5. J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," In ECCV, 2016
6. T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-DoF global localization in outdoor environments," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 1525-1530, <https://doi.org/10.1109/IROS.2017.8205957>.
7. J. Wu, L. Ma and X. Hu, "Delving deeper into convolutional neural networks for camera relocalization," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 5644-5651, <https://doi.org/10.1109/ICRA.2017.7989663>.
8. F. Y. Shih, "Improving the accuracy of deep localization models by spatially-augmented camera poses," M.S. thesis, Dept. Computer Science, National Tsing Hua University, 2020
9. A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in 2015 IEEE International Conference on Computer Vision (ICCV), (Los Alamitos, CA, USA), pp. 2938-2946, IEEE Computer Society, dec 2015

10. S. Brahmbhatt, J. Gu, K. Kim, J. Hays and J. Kautz, "Geometry-Aware Learning of Maps for Camera Localization," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 2616-2625, <https://doi.org/10.1109/CVPR.2018.00277>.
11. B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "AtLoc: Attention Guided Camera Localization," Proceedings of the AAAI Conference on Artificial Intelligence, Apr. 2020, <https://doi.org/10.1609/aaai.v34i06.6608>.
12. F. Xue, X. Wu, S. Cai and J. Wang, "Learning Multi-View Camera Relocalization With Graph Neural Networks," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 11372-11381, <https://doi.org/10.1109/CVPR42600.2020.01139>.
13. A. Valada, N. Radwan and W. Burgard, "Deep Auxiliary Learning for Visual Localization and Odometry," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 2018, pp. 6939-6946, doi: <https://doi.org/10.1109/ICRA.2018.8462979>.
14. Radwan, N., Valada, A., Burgard, W.: VLocNet++: Deep Multitask Learning for Semantic Visual Localization and Odometry. IEEE Robotics and Automation Letters **3**(4), 4407-4414 (2018). <https://doi.org/10.1109/LRA.2018.2869640>
15. Z. Laskar, I. Melekhov, S. Kalia and J. Kannala, "Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network," 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 2017, pp. 920-929, <https://doi.org/10.1109/ICCVW.2017.113>.
16. V. Balntas, S. Li, and V. A. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," in European Conference on Computer Vision, 2018
17. M. Ding, Z. Wang, J. Sun, J. Shi and P. Luo, "CamNet: Coarse-to-Fine Retrieval for Camera Re-Localization," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 2871-2880, <https://doi.org/10.1109/ICCV.2019.00296>.
18. H. Taira et al., "InLoc: Indoor Visual Localization with Dense Matching and View Synthesis," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7199-7209, <https://doi.org/10.1109/CVPR.2018.00752>.
19. E. Brachmann et al., "DSAC - Differentiable RANSAC for Camera Localization," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 2492-2500, <https://doi.org/10.1109/CVPR.2017.267>.
20. E. Brachmann and C. Rother, "Learning Less is More - 6D Camera Localization via 3D Surface Regression," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 4654-4662, <https://doi.org/10.1109/CVPR.2018.00489>.
21. E. Brachmann and C. Rother, "Visual camera re-localization from RGB and RGB-D images using DSAC," TPAMI, 2021
22. L. Yang, Z. Bai, C. Tang, H. Li, Y. Furukawa and P. Tan, "SANet: Scene Agnostic Network for Camera Localization," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 42-51, <https://doi.org/10.1109/ICCV.2019.00013>.
23. L. Zhou et al., "KFNet: Learning Temporal Camera Relocalization Using Kalman Filtering," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 4918-4927, <https://doi.org/10.1109/CVPR42600.2020.00497>.

24. X. Li, S. Wang, Y. Zhao, J. Verbeek, and J. Kannala, "Hierarchical scene coordinate classification and regression for visual localization," in CVPR, 2020
25. S. Tang, C. Tang, R. Huang, S. Zhu and P. Tan, "Learning Camera Localization via Dense Scene Matching," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 1831-1841, <https://doi.org/10.1109/CVPR46437.2021.00187>.
26. S. Tang, S. Tang, A. Tagliasacchi, P. Tan, and Y. Furukawa, "Neumap: Neural coordinate mapping by auto-transdecoder for camera localization," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023
27. L. Kneip, D. Scaramuzza and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," CVPR 2011, Colorado Springs, CO, USA, 2011, pp. 2969-2976, doi: <https://doi.org/10.1109/CVPR.2011.5995464>.
28. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981)
29. Chum, O., Matas, J.: Optimal Randomized RANSAC. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(8), 1472–1482 (2008). <https://doi.org/10.1109/TPAMI.2007.70787>
30. S. Dong, S. Wang, Y. Zhuang, J. Kannala, M. Pollefeys, and B. Chen, "Visual localization via few-shot scene region classification," in 2022 International Conference on 3D Vision (3DV), (Los Alamitos, CA, USA), pp. 393-402, IEEE Computer Society, sep 2022
31. M. Bertalmio, A. L. Bertozzi and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, <https://doi.org/10.1109/CVPR.2001.990497>.
32. J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi and A. Fitzgibbon, "Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images," 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 2930-2937, <https://doi.org/10.1109/CVPR.2013.377>.
33. J. Valentin et al., "Learning to Navigate the Energy Landscape," 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 2016, pp. 323-332, <https://doi.org/10.1109/3DV.2016.41>.
34. P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
35. P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020





# Multimodal Point Cloud Completion via Residual Attention Feature Fusion

Junkang Wan , Hang Wu, and Yubin Miao  

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China  
{wanjunkang123,ybmiao}@sjtu.edu.cn

**Abstract.** Point cloud completion aims to fill in incomplete or partially missing point cloud data to restore its complete shape information. Currently, some methods attempt to incorporate image modality information to achieve high-quality point cloud completion. However, they fail to fully integrate complementary features within the multimodal context. In this paper, we propose a novel Residual Multimodal Fusion network for point cloud completion, significantly improving the quality of shape completion. Specifically, we introduce a cross-modal residual feature fusion module to capture local shape features. It uses cross-modal attention mechanisms, while employing a residual structure to mitigate the process of globalizing features, thus effectively enhancing feature diversity. The decoder adopts an innovative attention-based multi-branch structure to reconstruct the complete point cloud by regions. Additionally, the point cloud refinement module is divided into local refinement units and view-assisted units, which can simultaneously capture global shape structures and local details, reducing outliers in the predicted point cloud. Experiments show that our network achieves competitive performance on synthetic and real-world datasets, outperforming existing methods.

**Keywords:** Point cloud completion · Multi-modality · Feature fusion · Transformer.

## 1 Introduction

Nowadays, the popularity of depth cameras and LiDAR has made it easier to capture color images and 3D point clouds, leading to wide applications of point clouds in fields like autonomous driving [10], shape understanding [26], and robotics [14]. However, due to constraints such as perspective and occlusion, 3D point clouds collected from real-world scenes are often sparse and incomplete, resulting in the loss of geometric and semantic information. Completing incomplete 3D point clouds to obtain a full representation is a challenging issue in current point cloud applications.

Most previous methods [24] [20] [21] [13] [29] only used partial point clouds as input. For example, by combining a PointNet-based [15] encoder with a folding-based decoder, PCN [24] represented the first dedicated network for completing

point clouds, marking an important milestone in the field. SnowflakeNet [20] modeled the generation of complete point clouds as the snowflake-like growth of points, generating detail-rich complete point clouds. However, these methods extracted global features from partial point clouds, failing to fully utilize the details carried by local features. Additionally, self-occlusion of objects made it difficult to determine their missing parts. Recently, some methods [27] [1] [30] have utilized 2D images to assist in completing point clouds, aiming to introduce image modal information into point cloud completion and leverage the complementary information between images and point clouds to achieve high-quality completion. For example, ViPC [27] fused information by estimating rough point clouds from images using single-view reconstruction techniques. However, directly predicting complete point clouds from images was difficult and inaccurate. Despite considerable efforts in using images to assist in completing point clouds, fusing data from cross-sensors remains challenging. It is difficult to make images a true complement to high-quality point cloud complementation tasks.

In this paper, we aim to devise a module that seamlessly blends the complementary information/features from 2D images and 3D point clouds. Additionally, we seek to develop a framework for merging these two modalities, thereby significantly enhancing the quality of completed point cloud shapes. Inspired by previous work, our point cloud encoder adopts the three-layer set abstraction in PointNet++ [16] to aggregate local features, which utilizes a hierarchical architecture to extract local features of the point cloud layer by layer. In addition, we employ Point transformer [28] to merge local shape contexts, aiming to learn the correspondence between points in local neighborhoods. Our image encoder, on the other hand, leverages multiple 2D convolutional layers to extract features from 2D images representing local shapes. Specifically, we introduce a cross-modal residual fusion module that capitalizes on cross-modal attention mechanisms to fully integrate features from both modalities. The residual structure [9] within this module mitigates the process of feature globalization, thereby effectively enhancing feature diversity. Our decoder is designed to convert local features into complete point clouds. It innovatively adopts a multi-branch decoder based on attention to regionally reconstruct complete point clouds. To comprehensively integrate image information in a coarse-to-fine manner for shape completion, we have devised a point cloud refinement module. This module is segmented into local refinement units and view-assisted units, enabling it to simultaneously capture global shape structures and local details. Additionally, it predicts the offset of each point and calibrates it into the final result.

In summary, the main contributions of this paper are as follows:

1. We design a novel cross-modal residual fusion module that combines the attention mechanism and residual connection with a gated variable weighting. The module achieves capturing complementary features in different modal spaces and fusing multimodal contextual features to fully integrate point cloud and image features.

2. We propose a novel Residual Multimodal FusionNet (RMF-Net) for point cloud completion. Its decoder uses an attention-based multi-branching structure to reconstruct the complete point cloud by region. The refinement module uses view features to constrain the point cloud. Our approach generates convincing complete point clouds from coarse to fine.
3. Our experiments show that our network has competitive performance compared to existing methods on both synthetic and real-world datasets.

We experiment on ShapeNet [2] dataset with rendered images and KITTI [7] dataset to test our approach. The quantitative and qualitative evaluations from the experiments indicate that our proposed method (RMF-Net) achieves competitive performance compared to recent representative methods.

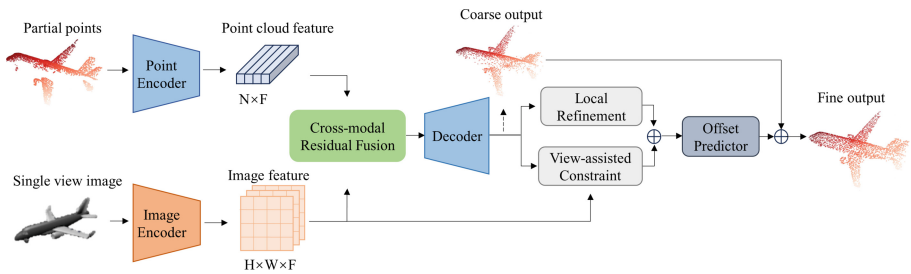
## 2 Related Work

**Unimodal Point Cloud Completion** Due to PointNet [15], learning and extracting features from unordered point sets became feasible, leading to the rapid development of completion methods based on deep learning. PCN [24] was the pioneer in introducing a point cloud completion network structure based on PointNet [15], employing an encoder-decoder design. The encoder extracts global features from incomplete point clouds and then folds 2D grids [22] to reconstruct intricate complete shapes. This encoder-decoder architecture has been widely adopted by most methods, with popular point cloud encoders including PointNet [15], PointNet++ [16], and DGCNN [18]. GRNet [21] consolidates unordered and irregular point clouds into regular 3D grids while preserving the spatial layout of the point set, enabling CNN utilization without sacrificing structural information. Moreover, VRCNet [13] introduces a dual-path architecture for probabilistic modeling and a relationship-enhancement module based on VAE, thereby refining local shape details. PointTr [23] adapts transformer blocks to leverage the inductive bias of 3D geometry, creating a geometry-aware block to simulate local geometric relationships for point cloud completion. Recently, SeedFormer [29] introduced a point cloud representation called Patch Seeds based on key point features. Unlike prior methods relying on global feature vectors, it not only captures the general structure obtained from partial inputs but also retains regional information regarding local patterns. AnchorFormer [3] innovatively employs pattern-aware discriminative nodes, termed anchors, to dynamically capture the regional information of objects. It models region discrimination by learning a set of anchors based on input local observations of point features.

**Image-Assisted Point Cloud Completion** Recently, the multimodal fusion of point clouds and images has proven to be effective in many tasks. ME-PCN [8] completes missing parts in point clouds based on blank information in occluded regions. However, its judgment of occluded areas is not accurate enough, leading to some unnatural results in the recovery. ViPC [27] first introduced the use of images to assist point cloud completion, explicitly reconstructing rough

point clouds from a single image, which is itself a challenging inverse problem. XMFNet [1] attempted to effectively combine features extracted from both modalities into a local latent space, proposing a multimodal feature fusion network. Additionally, CSDN [30] proposed a cross-modal shape transfer dual refinement network, allowing auxiliary images to participate in the coarse-to-fine completion pipeline, but the extracted global features may lose geometric details. In fact, how to fully utilize multimodal information remains an unresolved issue.

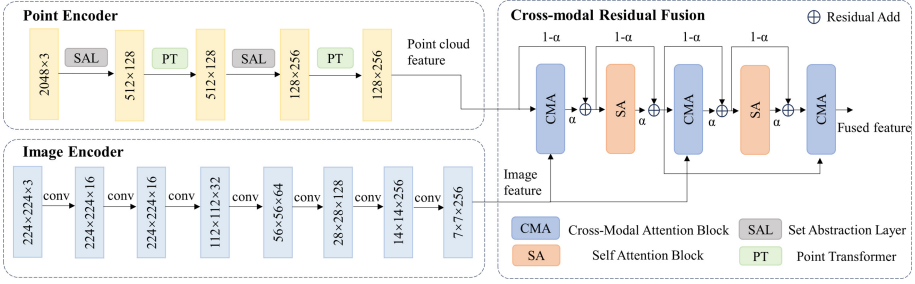
**Single-View Reconstruction** Predicting the invisible parts of an object from a single image has always been a major challenge in 3D reconstruction. Current methods mostly use deep neural networks for prediction. In the work [5], an attempt was made to use a GAN network model to convert RGB images into 3D point clouds. However, this method only focuses on reconstruction and does not involve model completion. PSGN [6] uses a point set generation network to generate 3D point clouds from a single image. This method consists of a 2D encoder and a 3D decoder for predicting complete point clouds, but its output point cloud is sparse and lacks detail. DensePCR [11] proposes a deep pyramid network to generate high-resolution 3D point clouds. It continuously predicts higher-resolution 3D point clouds in a layered manner. However, the issue of edge point pseudo-shadow still needs further resolution.



**Fig. 1.** The overall architecture of residual multimodal fusion network (RMF-Net). It consists of four parts: point cloud and image encoder, cross-modal residual fusion, decoder and point cloud refinement.

### 3 Methodology

In this section, we will provide a detailed overview of our network architecture. Firstly, we introduce the point cloud and image encoders, which serve as the primary feature extractors. Subsequently, we introduce a novel modality fusion module: the cross-modal residual fusion. Following this, we outline the multi-branch decoder and point cloud refinement module, designed to generate and refine the output point cloud. Fig. 1 shows the overall architecture of our network.



**Fig. 2.** Architecture of point cloud, image encoder and cross-modal residual fusion module.

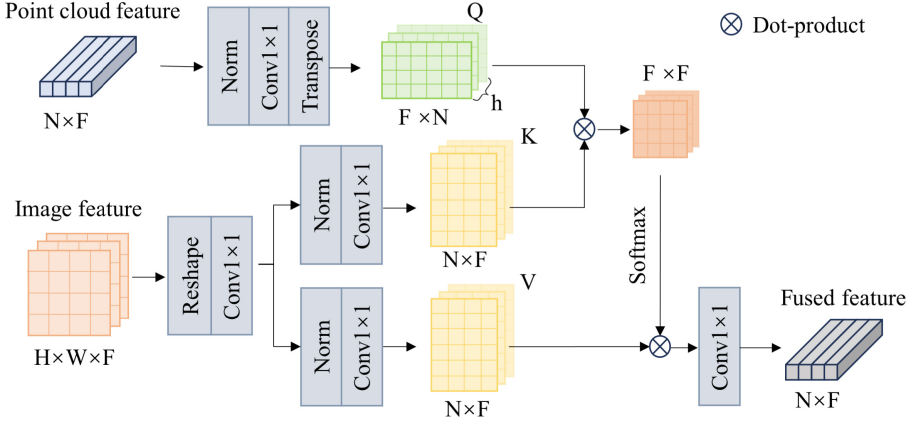
### 3.1 Point and Image Encoder

To fully extract the features of partial point clouds and images, their feature extraction processes should avoid interfering with each other. Therefore, we adopt two independent feature extractors: a point cloud encoder and an image encoder (see Fig. 2). The task of a point cloud encoder is to extract features from local shapes, rather than simply embedding the entire point cloud. This is because most models need to deal with local information, requiring association with only a small number of points, without considering the entire point cloud. However, at the same time, it is also crucial to have a sufficiently large receptive field to infer some global information about the entire object. To address this issue, we add a Point Transformer [28] layer between the Set Abstraction Layers of PointNet++ [16], aiming to capture the correlation between local features of point clouds. The Set Abstraction Layers aggregate and abstract points at different levels of point cloud to obtain higher-level feature representations. The Point Transformer [28] layer computes attention weights between points, allowing the network to adjust feature representations based on the similarity and importance between points.

On the other hand, the image encoder extracts 2D image features representing local shapes from view images. It utilizes a subnet of 7 convolutional layers to extract a  $7 \times 7 \times C$  feature map, and then obtains its local features through average pooling layers.

### 3.2 Cross-modal Residual Fusion

When processing point cloud and image data, the localized information we obtain may differ in domain but contains complementary features. Therefore, we need to effectively combine these two types of information. To integrate features from both modalities and capture local shape information, we introduce a cross-modal residual fusion module, which integrates low-level features from shallow layers into deep network layers, avoiding the attention mechanism losing focus on local information. Fig. 2 and 3 show the architecture of cross-modal attention module.



**Fig. 3.** Architecture of cross-modal attention module.

Specifically, we adopt the multi-head attention mechanism of the Transformer [17] to find features in the point cloud corresponding to features in the image. Through this layer, point cloud and image features are projected into tensors Query  $Q$ , Key  $K$ , and Value  $V$ , and then different features of different image regions are aggregated based on their associated weights. At the same time, we use skip connections to allow attention information to propagate between consecutive layers, which can integrate low-level features while maintaining the original ability of the Transformer to extract context. The self-attention layer performs permutation-invariant transformations on point features with a global receptive field, allowing better integration of information not correctly integrated in the image.

However, the network may accumulate an excessive amount of attention information related to low-level features, thereby hindering the network from learning higher-level representations. To address this issue, we introduce a learnable gating variable  $\alpha$ , allowing the network to autonomously determine how much attention to propagate between layers. The cross-modal residual fusion module is represented as follows:

$$K_l = \begin{cases} F_l W_l^K, & l = 2n - 1, n \in \mathbb{N}^* \\ X_{l-1} W_l^K, & l = 2n, n \in \mathbb{N}^* \end{cases}, V_l = \begin{cases} F_l W_l^V, & l = 2n - 1, n \in \mathbb{N}^* \\ X_{l-1} W_l^V, & l = 2n, n \in \mathbb{N}^* \end{cases} \quad (1)$$

$$Q_l = X_{l-1} W_l^Q \quad (2)$$

$$A(Q_l, K_l, V_l) = \text{softmax} \left( \frac{Q_l K_l^T}{\sqrt{F}} \right) V_l \quad (3)$$

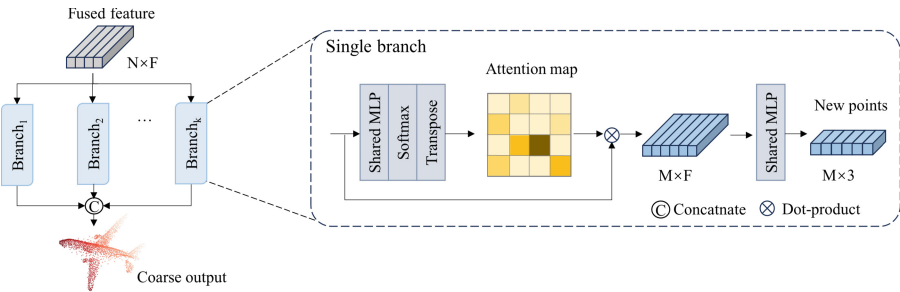
$$X_{l+1} = \begin{cases} \text{LN}[A(Q_l, K_l, V_l)], & \text{if } l = 0 \\ \text{LN}[\alpha_l A(Q_l, K_l, V_l) + (1 - \alpha_l) X_l], & \text{otherwise} \end{cases} \quad (4)$$

Where  $l$  denotes the  $l$ -th attention layer,  $W_l^Q \in \mathbb{R}^{F_P \times F}$ ,  $Q_l, K_l, V_l \in \mathbb{R}^{N_p \times F}$ ,  $\alpha \in [0, 1]$ , the fused feature  $X_l \in \mathbb{R}^{N_p \times F}$ ,  $A$  represents the multi-head attention

mechanism of the Transformer, LN represents layer normalization. For cases where  $l$  is odd,  $W_l^K, W_l^V \in \mathbb{R}^{F_I \times F}$ , and the attention layer is a cross-modal attention layer; for cases where  $l$  is even,  $W_l^K, W_l^V \in \mathbb{R}^{F_P \times F}$ , the attention layer is a self-attention layer.  $F_p$  represents the feature dimension of the point cloud,  $F_I$  represents the feature dimension of the image, and  $N_p$  represents the number of feature points. It is worth noting that the cross-attention layer at the end of this module takes low-level features as input, merging information from the end and middle of the module together.

### 3.3 Attention-based Decoder

The point cloud decoder aims to reconstruct the complete shape from fused features, while incorporating farthest point sampling [12] (FPS) to retain parts of the input point cloud. Inspired by [25], our approach involves a process from local to global: employing different branches to predict multiple point clusters, each corresponding to different parts of the point cloud, which are then merged into a global point cloud. In the specific implementation,  $N$  temporary points in the fused features are mapped to an  $N \times M$  matrix through a shared multi-layer perceptron (MLP). Next, a softmax activation function is applied along the  $N$  dimension of the matrix, followed by transposing the matrix, generating an  $M \times N$  attention map. Finally, based on the attention map, the original  $N$  points are aggregated to generate  $M$  new points. These new points represent a cluster of points. The architecture of the decoder is shown in Fig. 4.



**Fig. 4.** Architecture of point cloud decoder. It combines point clusters generated by multiple branches into a global point cloud.

This method utilizes attention mechanism to generate each temporary point, and the convex combination is reformed by weighted operations of the attention map. Specifically, if  $N$  temporary points exist in set  $S$ , forming the convex hull  $\text{conv}(S)$ , then the  $M$  points output by this branch are located inside  $\text{conv}(S)$ . This cohesive constraint ensures that the distribution of generated points is concentrated rather than dispersed. Therefore, when our architecture has multiple

branches, the points generated by each branch will automatically gather together to form clusters of points:

$$D_j = \text{MLP}_j^1(X), j = 1, 2 \dots K \quad (5)$$

$$\hat{Y}_j = \left( \text{softmax}(\text{MLP}_j^2(D_j))^T D_j \right) W_j \quad (6)$$

$$\hat{Y} = \text{concat} \left[ \hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_K, \text{FPS}(P_0) \right] \quad (7)$$

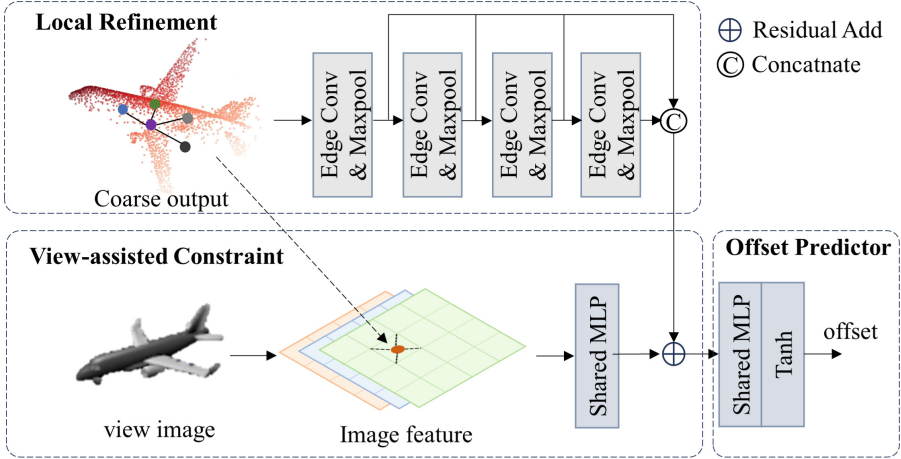
Where the multilayer perceptron  $\text{MLP}_j^1 : \mathbb{R}^F \rightarrow \mathbb{R}^{F'}$ ,  $\text{MLP}_j^2 : \mathbb{R}^{F'} \rightarrow \mathbb{R}^{N_1/K}$ ,  $S_j \in \mathbb{R}^{M \times N_1/K}$  represents the attention map of branch  $j$ ,  $\hat{Y}_j$  denotes the point cluster generated by branch  $j$ ,  $W_j \in \mathbb{R}^{F' \times 3}$  is the projection matrix in 3D space,  $P_0$  is the input point cloud, "concat" refers to concatenation operation, FPS stands for farthest point sampling. The equation 7 combines the output point clouds generated by all branches and the point clouds sampled by FPS to produce a rough point cloud.

### 3.4 Point cloud refinement and offset prediction

To further adjust the positions of calibration points [19], we designed a coordinate refinement module aimed at generating a set of coordinate offsets for each point. This module consists of two units, namely the local refinement unit and the view-assisted constraint unit (see Fig 5). The local refinement unit employs DGCNN [18] to learn the offset features  $F_P^{off}$  for each point in the calibration shape. Specifically, we utilize four layers of EdgeConv [18], which effectively models the features of the local neighborhood of point clouds. EdgeConv [18] is based on edge features relative to its neighboring points to obtain the global feature of that local neighborhood. The point-wise features outputted by each EdgeConv [18] module are concatenated to obtain a fusion of global and local features, serving as the offset feature  $F_P^{off}$  for each point.

However, due to reasons such as missing data, these partial point clouds may not provide complete local shape information, especially in specific areas of ground truth. Meanwhile, RGB images typically contain information about the underlying 3D perceptual shape attributes, such as boundaries, textures, and local connections. To address this issue, we adopt an approach called the view-assisted constraint unit to repair the missing information in the partial point cloud, which is obtained by learning image features. This constraint unit projects 3D points through camera parameters onto the last four feature maps (derived from the 2D encoder), and then merges them using bilinear interpolation from nearby pixels. Subsequently, through a residual MLP module (i.e., Offset Predictor), these obtained per-point features are processed to calibrate the final offset, thereby repairing the missing information of the partial point cloud.





**Fig. 5.** Architecture of point cloud refinement and offset prediction module. It reduces outliers in the predicted point cloud.

### 3.5 Loss Function

The loss function measures the difference between point cloud  $P_{\text{fine}}$  and ground truth  $P_{\text{gt}}$ . Chamfer Distance (CD) is a commonly used loss function in 3D point cloud completion, so we choose Chamfer Distance (CD) as the loss function. The completion process is divided into two steps, namely generating coarse point clouds and refining point clouds. The loss function 8 we set consists of two terms with hyperparameters  $\alpha$  weighted.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{CD}(P_{\text{coarse}}, P_{\text{gt}}) + \alpha \mathcal{L}_{CD}(P_{\text{fine}}, P_{\text{gt}}) \tag{8}$$

where  $\mathcal{L}_{CD}$  is defined as:

$$\mathcal{L}_{CD}(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2^2 \tag{9}$$

In the experiment, we will perform the first 50k iterations by  $\alpha$  increase from 0.01 to 2 because  $P_{\text{coarse}}$  is more important at the beginning of training.

## 4 Experiments

### 4.1 Datasets

We collect point clouds and render images on the ShapeNet [2] dataset. It contains 8 categories and 28974 objects covering airplanes, benches, cabinets, and cars, following the training and test set partitioning of PCN [24]. The point clouds of this dataset are presented in two forms: complete ground truth point

clouds and partially occluded point clouds. The incomplete point cloud comprises 2048 points, generated from the respective viewpoint (taking into account occlusion). The complete ground truth point clouds consist of 2048 points uniformly sampled from the mesh surfaces of ShapeNet. For rendering the images, we rendered RGB images from ShapeNet’s CAD model by Blender. The image data follows the 3D-R2N2 [4] rendering of 24 viewpoints. During the training process, we randomly select an image viewpoint, and align the point cloud with the chosen image for each training data pair.

In addition, we evaluate our RMF-Net on a real-world dataset (i.e., the KITTI [7] dataset). According to the input settings, we extract point clouds on the surface of the real scanned object, and then downsample them into a point cloud of 2048 points.

## 4.2 Implementation Details

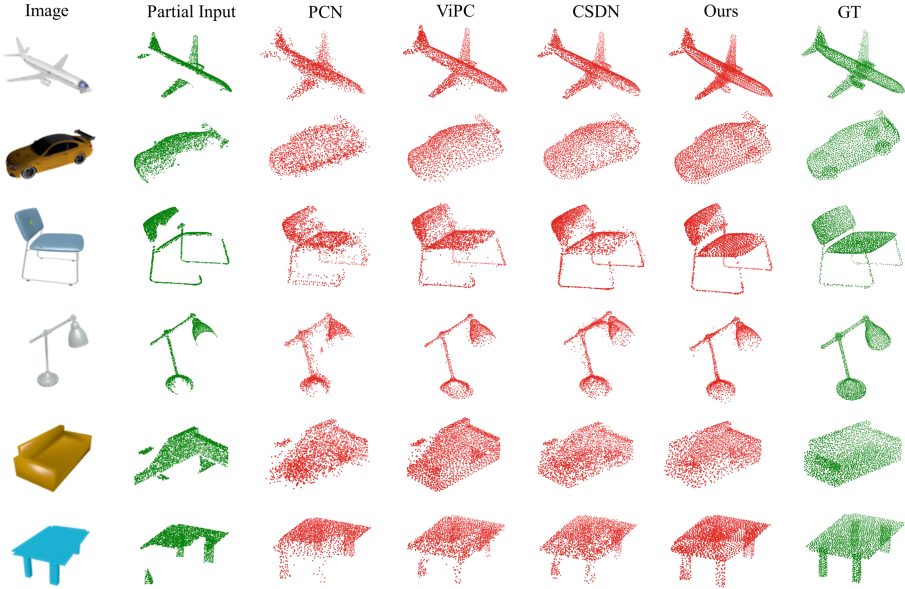
The input image size is  $224 \times 224 \times 3$ , and the input number of points is 2048. The point cloud encoder outputs point cloud features with a dimensionality of 256. The architecture of the image encoder is shown in Fig. 2, with its output feature map being  $14 \times 14 \times 256$ . The multi-head attention in the feature fusion module consists of 4 attention heads, with an embedding size set to 256. The decoder has  $k = 8$  branches, each producing  $M = 128$  points. The generated coarse point cloud comprises 2048 points. We implemented our model using PyTorch, training it on an Ubuntu 18.04 system equipped with a single NVIDIA RTX A6000 GPU. The entire network was trained end-to-end using the Adam optimizer for approximately 40 epochs, with a batch size of 32. The learning rate was initialized to  $5 \times 10^{-5}$  and decayed by a factor of 0.1 every 15 epochs.

## 4.3 Quantitative Comparison with Baselines

We conducted experiments on the ShapeNet dataset with rendered images. Table 1 and 2 present quantitative results comparing our method with other state-of-the-art approaches, including several representative methods for completing point clouds: PCN [24], GRNet [21], PointTr [23], ViPC [27], Seedformer [29], and CSDN [30]. PCN employs a point-based approach, while PointTr and Seedformer utilize Transformer-based techniques, and ViPC and CSDN employ multimodal methods. From the data in Table 1 and 2, it is evident that our approach outperforms previous multimodal point cloud completion methods. Specifically, compared to the best-performing previous multimodal method, CSDN, our method reduces the average CD across all categories by 0.393. Notably, in the categories of "sofa" and "lamp", our method achieves a CD reduction of over 0.6 compared to the optimal method.

## 4.4 Qualitative Results

Fig. 6 shows the qualitative results of our method on the ShapeNet dataset with rendered images. compared to state-of-the-art approaches. From Fig. 6, it



**Fig. 6.** Qualitative results on the ShapeNet dataset with rendered images. Our RMF-Net result has more local details.

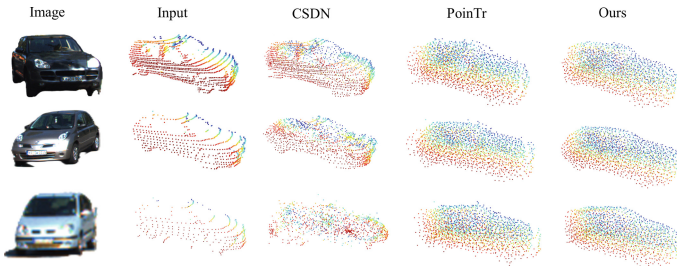
**Table 1.** Quantitative comparison on the ShapeNet dataset with rendered images. Mean Chamfer Distance per point  $\times 10^{-3}$  (lower is better).

Methods	Avg	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
PCN [24]	5.619	4.246	6.409	4.840	7.441	6.331	5.668	6.508	3.510
GRNet [21]	3.171	1.916	4.468	3.915	3.402	3.034	3.872	3.071	2.160
PointTr [23]	2.851	1.686	4.001	3.203	3.111	2.928	3.507	2.845	1.737
ViPC [27]	3.308	1.760	4.558	3.138	2.476	2.867	4.481	4.990	2.197
Seedformer [29]	2.902	1.716	4.049	3.392	3.151	3.226	3.603	2.803	1.679
CSDN [30]	2.507	1.251	3.670	2.977	2.835	2.554	3.240	2.575	1.742
Ours	<b>2.114</b>	<b>1.008</b>	<b>3.027</b>	<b>2.714</b>	<b>2.285</b>	<b>1.673</b>	<b>2.613</b>	<b>2.219</b>	<b>1.375</b>

can be observed that most methods effectively recover the missing parts while retaining some input. Compared to PCN, ViPC, and CSDN, our method restores a clearer overall shape and finer local details, with a neater arrangement of points. Particularly, it performs well in recovering certain details such as airplane engines and table legs, while other methods can only predict rough shapes. In cases of large missing areas, such as chairs, our method generates complete chair seats and legs, while other methods produce noisy points.

**Table 2.** Quantitative comparison on the ShapeNet dataset with rendered images. F-Score@0.001 (higher is better).

Methods	Avg	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
PCN [24]	0.407	0.578	0.270	0.331	0.323	0.456	0.293	0.431	0.577
GRNet [21]	0.601	0.767	0.426	0.446	0.575	0.694	0.450	0.639	0.704
PointTr [23]	0.683	0.842	0.516	0.545	0.662	0.742	0.547	0.723	0.780
ViPC [27]	0.591	0.803	0.451	0.512	0.529	0.706	0.434	0.594	0.730
Seedformer [29]	0.688	0.835	0.551	0.544	0.668	0.777	0.555	0.716	0.786
CSDN [30]	0.695	0.862	0.548	0.560	0.669	0.761	0.557	0.729	0.782
Ours	<b>0.720</b>	<b>0.897</b>	<b>0.585</b>	<b>0.574</b>	<b>0.712</b>	<b>0.816</b>	<b>0.608</b>	<b>0.752</b>	<b>0.814</b>

**Fig. 7.** Visualization results on the KITTI dataset. Our RMF-Net can predict reasonable shapes.

#### 4.5 Results on Real-world Scans

To evaluate the performance in real-world scans, we experiment on the KITTI dataset, which includes raw point clouds and RGB images. We extract car objects in KITTI, where part of the point cloud is highly sparse and we remove objects with less than 100 points. Fig. 7 shows some visualisation results of our method and the CSDN and PoinTr methods, where the output is selected from 2048 points by farthest point sampling (FPS). Compared to CSDN and PoinTr, our network predicts reasonable shapes and recovers clearer unobserved parts, while CSDN has difficulty in dealing with the domain gap problem, and the complementary results produce noise. In contrast, our method is able to recover reliable results in real-world scans.

#### 4.6 Ablation Study

To evaluate the effectiveness of the model design, we conducted ablation studies on key modules. Specifically, we systematically removed or modified modules, including the image feature branch, cross-modal residual fusion, and point cloud refinement, to analyze the contributions of these three key modules to the model.

**Table 3.** Ablation study for our method.

Methods	CD $\times 10^{-3}$ F-Score	
w/o Image	3.218	0.610
w/o Cross-Modal Fusion	2.849	0.667
w/o Residual Connections	2.372	0.693
w/o Point Cloud Refinement	2.535	0.684
Ours	<b>2.114</b>	<b>0.720</b>

**Effect of Image Input** This ablation study aimed to explore the impact of image inputs on completion effectiveness. Specifically, we converted the network into a single-modal version to analyze the contribution of image input modality. We conducted ablation on single-view images, where the image encoder and view auxiliary units were removed, leaving only the self-attention block in the feature fusion module. Table 3 shows the results of this ablation study. Without input images, i.e., only using point clouds for reconstruction compared to the multi-modal approach, the average CD value increased by 1.1, while the F-score decreased by 0.1. The comparison indicates that images can provide complementary information about point cloud shapes to aid in point cloud completion.

**Effect of Cross-Modal Residual Fusion** To demonstrate the contribution of the cross-modal residual module to completion performance, we conducted ablation experiments on this module. Specifically, we implemented two approaches: (1) replacing the cross-modal attention block with a self-attention block; (2) removing the residual connections within the module. As shown in Table 3, without using cross-modal attention, the performance of our framework slightly decreased. This demonstrates that our cross-modal residual fusion module integrates complementary information from both modalities, rather than solely reconstructing shapes from images.

**Effect of Point Cloud Refinement** The point cloud refinement module aims to correct outliers in the point cloud. We conducted an ablation of this module to assess its impact on completion effectiveness. Without the point cloud refinement module for point cloud calibration, the average CD value increased by 0.4, and the F-score decreased by 0.04. This implies that the point cloud refinement module can predict the coordinates of calibration points more accurately.

## 5 Conclusion

This paper aims to design a module that fully integrates 2D image and 3D point cloud information. It proposes a framework for blending both modalities to significantly improve the quality of completing the point cloud shape. By introducing a cross-modal residual fusion module, it utilizes cross-modal attention mechanisms

to fully integrate the features of both modalities while using a residual structure to ease the process of feature globalizing, effectively enhancing feature diversity. The decoder innovatively adopts an attention-based multi-branch decoder to reconstruct the complete point cloud by region. Additionally, the point cloud refinement module is divided into local refinement units and view-assisted units, which can simultaneously capture global shape structures and local details, predicting the offset of each point and calibrating them to achieve the final result. Experimental results demonstrate that on synthetic and real-world datasets, this network performs competitively, surpassing existing methods.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China under Grant 51975361.

## References

1. Aiello, E., Valsesia, D., Magli, E.: Cross-modal learning for image-guided point cloud shape completion. *Adv. Neural. Inf. Process. Syst.* **35**, 37349–37362 (2022)
2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint [arXiv:1512.03012](https://arxiv.org/abs/1512.03012) (2015)
3. Chen, Z., Long, F., Qiu, Z., Yao, T., Zhou, W., Luo, J., Mei, T.: Anchorformer: Point cloud completion from discriminative nodes. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 13581–13590 (2023)
4. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: *European Conference on Computer Vision*. pp. 628–644 (2016)
5. Chu, P.M., Sung, Y., Cho, K.: Generative adversarial network-based method for transforming single rgb image into 3d point cloud. *IEEE Access* **7**, 1021–1029 (2018)
6. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*. pp. 605–613 (2017)
7. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* **32**(11), 1231–1237 (2013)
8. Gong, B., Nie, Y., Lin, Y., Han, X., Yu, Y.: Me-pcn: Point completion conditioned on mask emptiness. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 12488–12497 (2021)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 770–778 (2016)
10. Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monroy, A., Ando, T., Fujii, Y., Azumi, T.: Autoware on board: Enabling autonomous vehicles with embedded systems. In: *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. pp. 287–296 (2018)
11. Mandikal, P., Radhakrishnan, V.B.: Dense 3d point cloud reconstruction using a deep pyramid network. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. pp. 1052–1060 (2019)

12. Moenning, C., Dodgson, N.A.: Fast marching farthest point sampling. University of Cambridge, Computer Laboratory, Tech. rep. (2003)
13. Pan, L., Chen, X., Cai, Z., Zhang, J., Zhao, H., Yi, S., Liu, Z.: Variational relational point completion network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition. pp. 8524–8533 (2021)
14. Pomerleau, F., Colas, F., Siegwart, R., et al.: A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics* **4**(1), 1–104 (2015)
15. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
16. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural. Inf. Process. Syst.* **30**, 5105–5114 (2017)
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł, Polosukhin, I.: Attention is all you need. *Adv. Neural. Inf. Process. Syst.* **30**, 6000–6010 (2017)
18. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)* **38**(5), 1–12 (2019)
19. Wu, L., Zhang, Q., Hou, J., Xu, Y.: Leveraging single-view images for unsupervised 3d point cloud completion. *IEEE Transactions on Multimedia* pp. 1–14 (2023)
20. Xiang, P., Wen, X., Liu, Y.S., Cao, Y.P., Wan, P., Zheng, W., Han, Z.: Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5499–5509 (2021)
21. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: Grnet: Gridding residual network for dense point cloud completion. In: European Conference on Computer Vision. pp. 365–381 (2020)
22. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 206–215 (2018)
23. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointnet: Diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12498–12507 (2021)
24. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 International Conference on 3D Vision (3DV). pp. 728–737 (2018)
25. Zhang, K., Yang, X., Wu, Y., Jin, C.: Attention-based transformation from latent features to point clouds. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 3291–3299 (2022)
26. Zhang, Q., Hou, J., Qian, Y.: Pointmcd: Boosting deep point cloud encoders via multi-view cross-modal distillation for 3d shape recognition. *IEEE Transactions on Multimedia* (2023)
27. Zhang, X., Feng, Y., Li, S., Zou, C., Wan, H., Zhao, X., Guo, Y., Gao, Y.: View-guided point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15890–15899 (2021)
28. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16259–16268 (2021)

29. Zhou, H., Cao, Y., Chu, W., Zhu, J., Lu, T., Tai, Y., Wang, C.: Seedformer: Patch seeds based point cloud completion with upsample transformer. In: European Conference on Computer Vision. pp. 416–432 (2022)
30. Zhu, Z., Nan, L., Xie, H., Chen, H., Wang, J., Wei, M., Qin, J.: Csdn: Cross-modal shape-transfer dual-refinement network for point cloud completion. *IEEE Transactions on Visualization and Computer Graphics* pp. 3545–3563 (2023)





# GP-PCS: One-Shot Feature-Preserving Point Cloud Simplification with Gaussian Processes on Riemannian Manifolds

Stuti Pathak<sup>1</sup>(✉) , Thomas Baldwin-McDonald<sup>2</sup> , Seppe Sels<sup>1</sup> ,  
and Rudi Penne<sup>1</sup> 

<sup>1</sup> University of Antwerp, Prinsstraat 13, 2000 Antwerp, Belgium  
stuti.pathak@uantwerpen.be

<sup>2</sup> University of Manchester, Oxford Road, Manchester, M13 9PL, UK

**Abstract.** The processing, storage and transmission of large-scale point clouds is an ongoing challenge in the computer vision community which hinders progress in the application of 3D models to real-world settings, such as autonomous driving, virtual reality and remote sensing. We propose a novel, one-shot point cloud simplification method which preserves both the salient structural features and the overall shape of a point cloud without any prior surface reconstruction step. Our method employs Gaussian processes suitable for functions defined on Riemannian manifolds, allowing us to model the surface variation function across any given point cloud. A simplified version of the original cloud is obtained by sequentially selecting points using a greedy sparsification scheme. The selection criterion used for this scheme ensures that the simplified cloud best represents the surface variation of the original point cloud. We evaluate our method on several benchmark and self-acquired point clouds, compare it to a range of existing methods, demonstrate its application in downstream tasks of registration and surface reconstruction, and show that our method is competitive both in terms of empirical performance and computational efficiency. The code is available at <https://github.com/stutipathak5/gps-for-point-clouds>.

**Keywords:** Point clouds · Simplification · Gaussian processes · Riemannian manifolds

## 1 Introduction

Recent years have seen a growing need for the conversion of real-world objects to computerized models [9, 35] across several domains, such as digital preservation of cultural heritage [27] and manufacturing of mechanical parts for industry [21]. This need has given rise to a range of modern data acquisition techniques such as

S. Pathak and T. Baldwin-McDonald—Equal contribution.

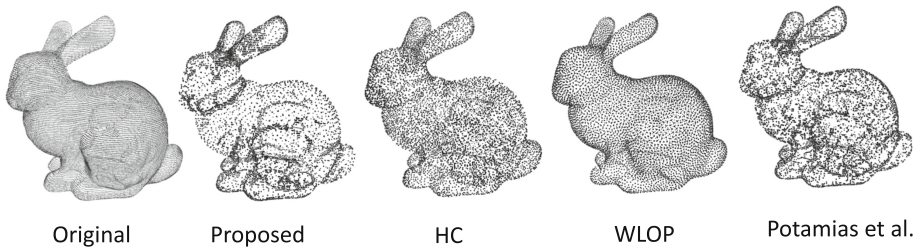
---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78456-9\\_28](https://doi.org/10.1007/978-3-031-78456-9_28).

laser scanning, which densely samples the surface of a 3D object, thereby generating millions of significantly redundant data points. 3D models can be obtained from this *point cloud* by constructing a polygonal mesh using techniques such as the *ball-pivoting algorithm* and *Poisson surface reconstruction* [1, 2, 16]. However, the sheer size of these dense point clouds makes this task computationally expensive in terms of both memory and time. Furthermore, the size of such generated meshes impedes further processing efforts, and necessitates the use of costly mesh simplification strategies [7, 11, 13] for size reduction. This makes efficient simplification of the underlying point cloud, prior to any surface reconstruction, an important and impactful problem which if addressed, has the potential to significantly improve the scalability of several computer vision applications.

The inherent dependency of surface reconstruction methods on surface normals, makes the visual perceptual quality of a point cloud an indirect yet important aspect of any mesh processing pipeline [7]. Although it is difficult to quantify this visual degradation in the case of point cloud simplification methods, one can say that the more enhanced the characteristic features of an object (such as sharp edges and high curvature regions) are in the simplified cloud, the higher is its human perceptual quality [19]. Therefore, an optimal point cloud simplification technique should preserve both the global structural appearance, and the salient features of the point cloud in question. Some of these methods will be discussed in detail in the upcoming section.

Given that the point cloud representing an object exists on a Riemannian manifold in 3D space, Euclidean distance fails to measure the intrinsic distance between any two points on its surface. Recently, techniques which extend existing machine learning methods to model functions defined on manifolds have gained popularity. For instance, *Gaussian processes* (GPs), a widely used class of non-parametric statistical models, which often use Euclidean distance-based covariance functions, have been made compatible for functions whose domains are compact Riemannian manifolds using ideas from harmonic analysis [5].



**Fig. 1.** Point cloud simplification methods typically fail to strike a balance between preserving sharp features and maintaining the overall structure of the original cloud. Our approach circumvents this trade-off by achieving both targets, as is evident from the simplified versions of the Stanford Bunny [20] obtained using the proposed technique and three pre-existing methods; Hierarchical Clustering (HC) [26], Weighted Locally Optimal Projection (WLOP) [14], and Potamias *et al.* [28] simplification.

In this work, we propose a novel, one-shot, feature-preserving simplification method using GPs with kernels defined on Riemannian manifolds. Using a greedy algorithm for GP sparsification, we iteratively construct a simplified representation of a point cloud without the need for any prior surface reconstruction or training on large point cloud datasets. We experiment on several point clouds, compare with several techniques and demonstrate competitive results both empirically and in terms of computational efficiency. Qualitatively, as shown in Fig. 1, our method effectively preserves visual features whilst providing a sufficiently dense coverage of the domain of the original cloud.

**Outline of the Paper:** Section 2 briefly reviews a number of existing point cloud simplification techniques which are relevant to our work. Section 3 provides background details regarding the computation of surface variation, GPs with kernels defined on non-Euclidean domains and a greedy subset-of-data scheme for GP inference. Section 4 outlines the proposed GP-based point cloud simplification algorithm. Section 5, in combination with the supplementary material, includes an empirical evaluation of our method on various benchmark and self-acquired point clouds, with comparisons to competing simplification techniques, along with applications to some downstream tasks and ablation studies. Finally, Sect. 6 summarises our contributions and provides a brief discussion of the scope for future work.

## 2 Related Work

In this section we will introduce a number of existing point cloud simplification techniques, with a particular focus on works which have a feature-preserving element to their approach. Some of the earliest curvature-sensitive simplification techniques were proposed by Pauly *et al.* [26] and Moenning *et al.* [25]. The former method, termed *Hierarchical Clustering* (HC), recursively divides the original point cloud into two sets, until each child set attains a size smaller than a threshold *size parameter*. Moreover, a *variation parameter* plays an important role in sparsifying regions of low curvature by selective splitting. The perceptual quality and the size of the simplified cloud depend entirely on these two parameters, which must be carefully and manually tuned, making HC unsuitable for automated applications. Additionally, the surface reconstructions obtained from HC-simplified point clouds are often poor for clouds with complex surfaces, as will be seen in Sect. 5. This is because it is challenging to tune the parameters of HC in such a way that preservation of sharp features is achieved whilst still ensuring dense coverage of the original cloud.

Another widely-used technique is *Weighted Locally Optimal Projection* (WLOP) proposed by Huang *et al.* [14]. In this work, the authors modified the existing parameterization-free denoising simplification scheme termed *Locally Optimal Projection* (LOP) [22], which is unsuitable for non-uniformly distributed point clouds. WLOP overcomes this limitation by incorporating locally adaptive

density weights into LOP. Although WLOP results in an evenly distributed simplified cloud, it still lacks sensitivity towards salient geometric features which will also become apparent in Sect. 5. Recently, Potamias *et al.* [28] have proposed a graph neural network-based learnable simplification technique which uses a modified variant of Chamfer distance in order to backpropagate errors. Their method can simplify point clouds in real-time but involves a computationally intensive training process using large point cloud datasets such as TOSCA [6]. Moreover, their model’s efficiency is limited to simplifying point clouds which are structurally similar to the learned data, as inherently neural networks struggle to generalize outside of the domain of the training data. Even more recent work from Wu *et al.* [33], named *APEs*, proposes an edge-sampling method which claims to capture the salient points within a point cloud using an attention mechanism. As shown in their paper, this technique generally provides good results for some point cloud tasks. However, as discussed by the authors themselves, the edge-enhancing nature of their method hinders upsampling operations, which can lead to poor reconstruction and segmentation results later.

*Approximate Intrinsic Voxel Structure for Point Cloud Simplification* (AIVS), introduced by Lv *et al.* [24], combines global voxel structure and local farthest point sampling to generate simplification demand-specific clouds which can be either isotropic, curvature-sensitive or have sharp edge preservation. As with HC however, AIVS requires manual tuning of user-specified parameters in order to obtain optimal results. Additionally, even in parallel computation mode, AIVS is quite costly in terms of computational runtime. Potamias *et al.* and Lv *et al.* do not provide open-source implementations of their curvature-sensitive simplification techniques, which poses a challenge for reproducibility and benchmarking. However, we thank the authors of Potamias *et al.* for directly providing some simplified point clouds; their results are included later in this paper. Qi *et al.* [29] introduced *PC-Simp*, a method which aims to produce uniformly-dense and feature-sensitive simplified clouds, leveraging ideas from graph signal processing. This uniformity depends on a *weight parameter* which as with HC and AIVS, is user-specified. Alongside simplification, they also apply their technique to point cloud registration. However, in practice PC-Simp is unreliable for complex-surfaced point clouds as it fails to provide a high degree of feature-preservation, regardless of the weight parameter chosen. Additionally, as discussed later in Sect. 5, the runtime of this technique is considerably longer than any other method tested.

Finally, it has been observed that most of the aforementioned works on feature-preserving point cloud simplification schemes experiment on structurally simple point clouds. Furthermore, surface reconstruction results are rarely presented and discussed. Hence, to underline the efficiency of our method, we experiment on point clouds generated from complex-surfaced objects and provide the corresponding reconstruction results. Also, some of the datasets used by the mentioned techniques are synthetically generated and already have a higher concentration of points around salient features when compared to low curvature regions (for example, the TOSCA dataset). Hence, unlike them, we do not

experiment on point clouds from these datasets as it defeats the purpose of being a feature-sensitive simplification technique.

### 3 Background

#### 3.1 Surface Variation

Consider an unstructured dense point cloud  $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$  of size  $N$  existing in 3D Euclidean space,  $\mathbb{R}^3$ . We can generate the local neighbourhood  $N_{\mathbf{p}_i}$  of each point  $\mathbf{p}_i$  in  $P$  by two different methods. Firstly, we can gather all of the points within a certain Euclidean distance  $r$  from  $\mathbf{p}_i$ ; this approach is referred to as *radius search*. Alternatively, we can gather all of the  $k$ -nearest Euclidean neighbours of  $\mathbf{p}_i$ , which is referred to as *KNN search*. The choice of this scale-factor ( $r$  or  $k$ ) not only depends on the size and density of a point cloud but also on the desired level of detail for a given application. These aspects make the task of automatic estimation of the neighbourhood of a point in a cloud an important, yet challenging one [31]. In this work, we implement the approach taken by the *CloudCompare* software package, where this process is automated by first calculating an approximate surface per point from the bounding box volume. This estimated value, along with a user-defined approximate neighbour number, is used to estimate a radius  $r$ , which is then used to perform radius search for each point. In our method, we have fixed this approximate neighbour number to 25 as it provides good empirical performance across a wide variety of point clouds. However, we provide ablation studies over a range of neighbourhood sizes in the supplementary material.

Several local surface properties [32] of the point cloud at a given query point  $\mathbf{p}_i$  can be estimated by analysing the eigenvalues and eigenvectors of the covariance matrix  $\mathbf{C}_i$  defined by the point's neighbourhood  $N_{\mathbf{p}_i} = \{\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \dots, \mathbf{p}_{i_n}\}$ :

$$\mathbf{C}_i = \begin{bmatrix} \mathbf{p}_{i_1} - \bar{\mathbf{p}}_i \\ \mathbf{p}_{i_2} - \bar{\mathbf{p}}_i \\ \dots \\ \mathbf{p}_{i_n} - \bar{\mathbf{p}}_i \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{p}_{i_1} - \bar{\mathbf{p}}_i \\ \mathbf{p}_{i_2} - \bar{\mathbf{p}}_i \\ \dots \\ \mathbf{p}_{i_n} - \bar{\mathbf{p}}_i \end{bmatrix}, \quad (1)$$

where,  $\bar{\mathbf{p}}_i$  is the centroid of all the points  $\mathbf{p}_{i_j} \in N_{\mathbf{p}_i}$ . By means of *principal component analysis* (PCA), we may now fit a plane tangent to the 3D surface, formed by all of the points within  $N_{\mathbf{p}_i}$ , at  $\bar{\mathbf{p}}_i$ . As  $\mathbf{C}_i$  is a  $3 \times 3$  symmetric and positive semi-definite matrix, all of its eigenvalues ( $\lambda_j, j \in \{0, 1, 2\}$ ) are positive and real, whilst the corresponding eigenvectors ( $\mathbf{v}_j$ ) form an orthogonal frame corresponding to the principal components of  $N_{\mathbf{p}_i}$ . If  $0 \leq \lambda_0 \leq \lambda_1 \leq \lambda_2$ , then  $\mathbf{v}_2$  and  $\mathbf{v}_1$  span the aforementioned tangent plane, whilst  $\mathbf{v}_0$  represents the vector perpendicular to it. Therefore,  $\mathbf{v}_0$  can be considered as an estimate of the surface normal to the point cloud (without actual surface reconstruction) at query point  $\mathbf{p}_i$ . Furthermore, as defined by Pauly *et al.* [26], we can calculate the *surface variation* at the query point as:

$$\sigma_n(\mathbf{p}_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}. \quad (2)$$

This quantity is not only closely related to the surface curvature at  $\mathbf{p}_i$  but also serves as a more suitable criterion for simplification, as discussed in detail by the authors [26].

### 3.2 Gaussian Processes on Riemannian Manifolds

Gaussian processes (GPs) are non-parametric Bayesian models which allow for a rigorous estimation of predictive uncertainty, and have been widely studied and applied by the machine learning community over the last two decades. Consider a scenario where we have a training dataset of  $N$  observations,  $\{\mathbf{x}_i, y_i\}_{i=1}^N$ , where  $\mathbf{x}_i \in \mathbb{R}^P$  and  $y_i \in \mathbb{R}$ . In our application,  $\mathbf{x}_i \in \mathbb{R}^3$  is a Euclidean coordinate, and  $y_i$  is the surface variation associated with said coordinate. We assume access to noisy observations of an underlying latent function, such that  $y_i = f(\mathbf{x}_i) + \epsilon_i$ , where  $\epsilon_i \sim \mathcal{N}(0, \sigma_y^2)$ . A GP defines a distribution over functions which we can use to infer the form of the true latent function which generated our training data. The GP prior can be written as  $f \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ , where,  $\mu(\cdot)$  and  $k(\cdot)$  are the mean and kernel functions respectively, which completely describe our process [30]. As is common, we assume a zero-mean prior throughout this work, using the kernel as the primary means of modeling the variation in our function over its domain. A popular choice for GP kernels is the Matérn class of covariance function, which takes the form,  $k_\nu(\mathbf{x}, \mathbf{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{r\sqrt{2\nu}}{\kappa}\right)^\nu K_\nu\left(\frac{r\sqrt{2\nu}}{\kappa}\right)$ , where  $r = \|\mathbf{x} - \mathbf{x}'\|$  and  $K_\nu$  is a modified Bessel function. We define  $\boldsymbol{\theta} = \{\sigma^2, \kappa, \nu\}$  to be the set of kernel hyperparameters;  $\sigma^2$  controls the variance of the GP,  $\kappa$  the lengthscale of its variation and  $\nu$  its degree of differentiability.

**Inference:** Using Bayes' Rule, we can condition our GP on the training data and derive closed form expressions for the posterior mean and covariance:

$$\boldsymbol{\mu}_{\text{post}} = \mathbf{K}_* (\mathbf{K} + \sigma_y^2 \mathbf{I})^{-1} \mathbf{y}, \quad (3)$$

$$\boldsymbol{\Sigma}_{\text{post}} = \mathbf{K}_{**} - \mathbf{K}_* (\mathbf{K} + \sigma_y^2 \mathbf{I})^{-1} \mathbf{K}_*^\top. \quad (4)$$

Generally, the noise variance  $\sigma_y^2$  and the kernel function hyperparameters  $\boldsymbol{\theta}$  are optimised via maximisation of the log-marginal likelihood, which can also be derived analytically. Where  $\mathbf{X} \in \mathbb{R}^{N \times P}$  and  $\mathbf{y} \in \mathbb{R}^N$  are matrix and vectorial representations of our training inputs and targets respectively, the log-marginal likelihood takes the form [30],

$$\begin{aligned} \log p(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}, \sigma_y^2) &= -\frac{1}{2} \mathbf{y}^\top (\mathbf{K} + \sigma_y^2 \mathbf{I})^{-1} \mathbf{y} \\ &\quad - \frac{1}{2} \log |\mathbf{K} + \sigma_y^2 \mathbf{I}| - \frac{N}{2} \log(2\pi). \end{aligned} \quad (5)$$

**Kernels on Manifolds:** Many different kernel functions for GPs exist, and choosing a kernel is in itself a model selection problem as some kernels are more suited to modeling certain types of data. However, one characteristic which many

kernels share is that they are defined using Euclidean distance. This presents an issue should we wish to use a GP to model variation in a quantity over a non-Euclidean space. Borovitskiy *et al.* [5] proposed a solution to this problem in the form of an extension to the Matérn kernel, which allows for modeling of functions whose domains are compact Riemannian manifolds. The approach proposed by the authors involves two stages. Firstly, numerical estimation of the eigenvalues  $\lambda_n$  and eigenfunctions  $f_n$  corresponding to the Laplace-Beltrami operator of the given manifold is performed. Secondly, for a manifold of dimensionality  $d$ , the kernel is approximated using a finite truncation of:

$$k_\nu(\mathbf{x}, \mathbf{x}') = \frac{\sigma^2}{C_\nu} \sum_{n=0}^{\infty} \left( \frac{2\nu}{\kappa^2} + \lambda_n \right)^{-\nu - \frac{d}{2}} f_n(\mathbf{x}) f_n(\mathbf{x}'), \quad (6)$$

where,  $C_\nu$  is a normalizing constant. The hyperparameters  $\sigma^2$ ,  $\kappa$  and  $\nu$  have similar interpretations to those introduced for the conventional Euclidean Matérn kernel.

### 3.3 Greedy Subset-of-Data Algorithm

A major challenge which arises when working with GPs in practice is the  $\mathcal{O}(N^3)$  complexity associated with performing exact inference, which arises due to the matrix inversions in Eqs. (3) and (4). To circumvent this issue, numerous formulations of *sparse GPs* have been proposed, many of which are based on approximate inference techniques and concepts such as *inducing points* [23]. In this work however, we consider the *subset-of-data* (SoD) approach. As explained in Sect. 8.3.3 of [30], it is a conceptually simple form of sparse approximation which allows for exact Bayesian inference. In this setting, rather than modifying the formulation of the GP itself, we simply perform exact inference using a carefully selected subset of  $M (\ll N)$  observations. Specifically, for our case we modify the greedy SoD approach of [18], which uses a selection criterion to sequentially construct a subset of size  $M$  which is representative of our full training set of  $N$  observations. We use this technique for GP sparsification in order to construct a set of inducing points for a point cloud which are best capable of representing the changes in surface variation over the cloud; this set of points forms our simplified point cloud. The original method involves randomly selecting one initial inducing point and then adding one point to the set at each iteration, however we have employed *farthest point sampling* (FPS) for selecting a set of initial inducing points instead of one, and we add several points to our set of inducing points at each iteration. Our approach is explained in further detail in Sect. 4.

Our method forms a simplified point cloud which is a subset of the original, thus the optimization problem is a discrete one. There has been recent work on inducing point optimization on discrete domains [10], however such methods only obtain comparable performance to methods based on greedy selection of the inducing points from the input domain, which are considerably conceptually simpler. The main disadvantage of a greedy approach is that the training set does not necessarily span the whole input domain, however in our setting this is

indeed the case, making our application especially well-suited to a greedy approach. Additionally, our proposed method allows us to obtain competitive results for clouds containing millions of points, whilst still employing exact Bayesian inference rather than approximate variational schemes, which can often underestimate the variance of the posterior distribution [4].

## 4 Point Cloud Simplification with Riemannian Gaussian Processes

In this section, we outline our GP-based approach with the help of a concise algorithm. We can represent a point cloud of size  $N$  as a set of 3D Euclidean coordinates  $P = \{\mathbf{x}_i\}_{i=1}^N$ , where  $\mathbf{x}_i \in \mathbb{R}^3$ . The surface variation  $y_i \in \mathbb{R}$  at each point in  $P$  can be computed using Equation (2). Using this data we formulate a regression problem, whereby we employ a GP with a Matérn kernel defined on a Riemannian manifold (as described in Sect. 3.2) to predict the surface variation from the coordinates of each point. We then employ the greedy subset-of-data scheme discussed in Sect. 3.3 in order to obtain a simplified set of  $M (\ll N)$  3D coordinates,  $P_{\text{simp}} = \{\mathbf{x}_j\}_{j=1}^M$ , where  $P_{\text{simp}} \subset P$ .

We formally outline our proposed approach in Algorithm 1.  $\text{FPS}(P, k_{\text{init}})$  denotes a function which selects  $k_{\text{init}}$  initial points from  $P$  using FPS; we use this to initialise our active set  $P_{\text{simp}}$  with an initial set of points from across the point cloud.  $\text{MAX}(\mathbf{s}, R, k_{\text{add}})$  selects the points from the remainder set  $R$  which are associated with the  $k_{\text{add}}$  largest values in our selection criterion vector  $\mathbf{s}$ . The notation  $\mathbf{y}(R)$  denotes a vector containing the target surface variation values associated with each of the points contained within the set  $R$ . At each step  $t$  of the algorithm, we update the posterior mean  $\boldsymbol{\mu}_t$  and covariance  $\boldsymbol{\Sigma}_t$  using Eqs. (3) and (4) respectively, where the active set  $P_{\text{simp}}$  is used as training data, whilst the remainder set  $R$  is unseen test data.

---

### Algorithm 1 GP-based simplification algorithm

---

**Data:**  $P, \mathbf{y}, M, k_{\text{init}}, k_{\text{add}}, k_{\text{opt}}$ , GP prior  $\mathcal{GP}(0, k(\cdot, \cdot))$ , where  $k$  is defined in Eq. (6)

**Result:**  $P_{\text{simp}}$

$P_{\text{opt}} \leftarrow$  random subset of  $k_{\text{opt}}$  points from  $P$ ;

Optimise GP hyperparameters using Eq. (5),  $P_{\text{opt}}$  and  $\mathbf{y}(P_{\text{opt}})$ ;

Active set  $P_{\text{simp}} \leftarrow \text{FPS}(P, k_{\text{init}})$ ;

Remainder set  $R \leftarrow P - P_{\text{simp}}$ ;

**while**  $|P_{\text{simp}}| < M$  **do**

Compute  $\boldsymbol{\mu}_t$  and  $\boldsymbol{\Sigma}_t$  using Eq. (3) and (4);

$\mathbf{s} \leftarrow \sqrt{\text{diag}(\boldsymbol{\Sigma}_t) + |\boldsymbol{\mu}_t - \mathbf{y}(R)|}$ ;

$P_{\text{simp}} \leftarrow P_{\text{simp}} + \text{MAX}(\mathbf{s}, R, k_{\text{add}})$ ;

$R \leftarrow R - \text{MAX}(\mathbf{s}, R, k_{\text{add}})$ ;

**end while**

---

To clarify, we predict the surface variation and the uncertainty values for  $R$  based on  $P_{\text{simp}}$  at each iteration of our algorithm. The selection criterion which



we use favours selection of points within the original cloud which lie in regions of high predictive uncertainty and/or error. By selecting a set of points using this criterion, we form a simplified cloud which implicitly favours selection of points surrounding finer details within the cloud, where the error and uncertainty is likely to be high if we have not yet selected a sufficient number of points around said location.

As  $P_{\text{simp}}$  grows with each iteration to be gradually more representative of our input data, the uncertainty and predicted surface variation values for points in  $R$  also change. For example, consider two neighbouring points on the tip of one of the Stanford bunny’s ears, and assume that neither of them are currently in  $P_{\text{simp}}$ . If one of these points is added to  $P_{\text{simp}}$ , the elements of the uncertainty  $\sqrt{\text{diag}(\Sigma_t)}$  and error  $|\mu_t - \mathbf{y}(R)|$  associated with the second point will decrease, and in subsequent iterations it may no longer be one of the top-ranked points based on the selection metric  $\mathbf{s}$ .

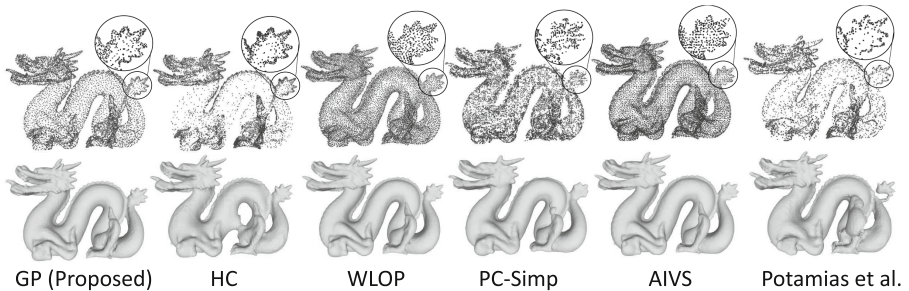
## 5 Empirical Evaluation

In this section, we extensively evaluate the proposed simplification method using various point cloud datasets and processing techniques. First, we compare our simplification technique both quantitatively and qualitatively using benchmark object level point clouds as given in Subsect. 5.1. Second, in Subsect. 5.2 we extend the use-case of our algorithm as a time and memory efficient pre-processing step for the downstream task of point cloud registration. Moreover, we provide some experiments on scene level and self-acquired point clouds along with ablation studies in the supplementary material (Sect. 2).

### 5.1 Benchmark Object Level Point Clouds

**Evaluation Criteria:** In order to evaluate the performance of our method in comparison to other simplification techniques, we firstly use each simplified point cloud obtained from three object level point clouds to form simplified meshes, using *screened Poisson surface reconstruction* [17]. We can then compute the reconstruction errors between the original meshes, and the reconstructed meshes formed from our simplified clouds. Specifically, we choose to evaluate the mean and maximum *Hausdorff distance* [8]. Evaluating the error associated with mesh reconstruction is effective at quantifying the ability of each method to preserve features from the original cloud, as accurate reconstruction of a mesh from a simplified point cloud requires that a high density of points be placed in the vicinity of finer details within the cloud. The *MeshLab* software was used to reconstruct all surfaces and compute the Hausdorff distances. Also, given that one of our primary aims is to preserve sharp features within each point cloud, we also report the *average surface variation* over each simplified point cloud. The surface variation at each point is computed using the approach described in Sect. 3.1.

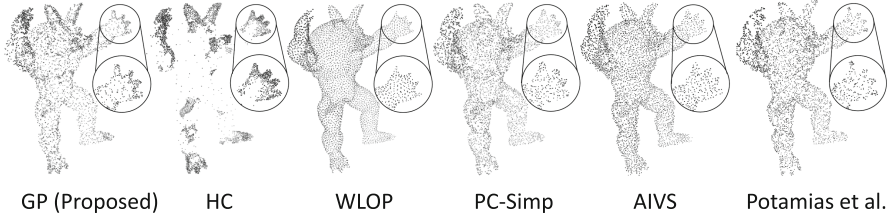
**Baselines:** We use the aforementioned evaluation procedure to compare our method (denoted *GP*) empirically to a number of competing simplification techniques discussed in Sect. 2. We compare our approach to *PC-Simp*, *AIVS*, *Potamias et al.*, *HC* and *WLOP*, with the latter two approaches implemented using the *CGAL* library. Additionally, we provide a visual comparison of our simplification method with *APES*. For the *HC* method, the size and variation parameters discussed in Sect. 2 were manually tuned to obtain approximate desired simplified sizes. Also, as noted in Sect. 2, we use the non-curvature aware version of the *AIVS* algorithm, as there is no available open-source implementation of the curvature-aware variant.



**Fig. 2.** Simplified representations of the Dragon point cloud for simplification ratio  $\alpha = 0.03$  (top row) and associated reconstructed meshes (bottom row) for all evaluated simplification techniques.

**Experimental Details:** We evaluate our proposed method and the aforementioned baselines on three complex object-level point clouds from the Stanford 3D Scanning Repository [20], namely *Armadillo* ( $N = 1,72,974$ ), *Dragon* ( $N = 4,37,645$ ) and *Lucy* ( $N = 1,40,27,872$ ). Let the *simplification ratio* be defined as  $\alpha = M/N$ . In this work we focus on the challenging regime where we wish to significantly reduce the size of the cloud, such that  $\alpha \ll 1$ . It is in this regime that feature-preserving techniques such as ours become particularly important, as we do not have a large number of points to select, thus we must efficiently select points which allow us to capture the salient features of the original cloud. We chose  $\alpha$  for each cloud by finding the minimum  $\alpha$  at which all evaluated techniques were capable of forming simplified clouds from which meshes visually comparable to the original meshes could be generated [20]. This value varies depending on the surface complexity of each cloud, thus for *Armadillo*, *Dragon* and *Lucy* we chose  $\alpha = 0.05$ ,  $0.03$  and  $0.002$  respectively. Additionally, we also visually evaluate the point cloud simplification results of all aforementioned techniques on a noisy *Armadillo* from the PCPNet dataset [12], with  $\alpha = 0.05$ . This corresponds to the original *Armadillo* model surface sampled  $10^5$  times ( $N = 1,00,000$ ), with Gaussian noise (of standard deviation

$\sigma = 2.5 \times 10^{-3} \times d$ , where  $d$  is bounding box diagonal length) added to every point position. We also perform the same evaluation for three objects, an airplane, a glass and a toilet ( $N = 2,048$  for all three) from ModelNet40 dataset [34] to compare our method with APES.

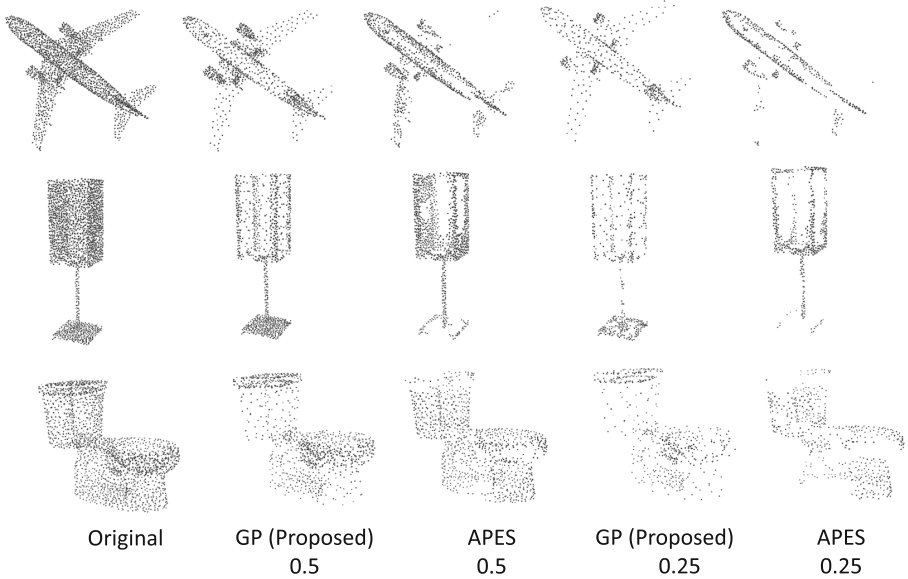


**Fig. 3.** Simplification results of a noisy Armadillo with Gaussian noise added to every point position (of standard deviation  $\sigma = 2.5 \times 10^{-3} \times d$ , where  $d$  is the bounding box diagonal length) for simplification ratio  $\alpha = 0.05$  for all evaluated simplification techniques.

**Discussion:** From the results presented in Table 1, it is clear that our proposed method is capable of comparable empirical performance to many of the existing methods for simplifying point clouds. The GP-based approach outperforms the AIVS baseline across all experiments and metrics, and outperforms the PC-Simp baseline on all but the mean Hausdorff distance for the Armadillo experiment. Moreover, our algorithm also runs considerably faster than both of these approaches. Note that due to the scale of *Lucy*, we were unable to evaluate PC-Simp on this cloud as it was taking more than two hours to run.

**Table 1.** Empirical results and total runtimes (time taken by surface variation computation and simplification) for all tested simplification methods and point clouds. We report the maximum and mean Hausdorff distances between the original meshes, and the meshes reconstructed from the simplified point clouds. Also reported is the average surface variation over each simplified point cloud. Best, second-best and third-best results are in red, green and blue respectively. It is worth mentioning that as per the evaluated metrics, our algorithm mostly stays within the top three methods.

	Mean Hausdorff Distance (↓)			Max. Hausdorff Distance (↓)			Mean Surface Variation (↑)			Total Time (s) (↓)		
	Armadillo	Dragon	Lucy	Armadillo	Dragon	Lucy	Armadillo	Dragon	Lucy	Armadillo	Dragon	Lucy
GP (ours)	0.246	0.000246	1.11	3.26	0.00457	195.78	0.0728	0.0546	0.0724	0.8	1.4	12.9
HC	0.374	0.000758	1.14	3.26	0.0141	195.41	0.0803	0.0686	0.0762	0.1	1.1	10.0
WLOP	0.197	0.000188	1.29	4.14	0.00417	195.52	0.0557	0.0413	0.0631	3.5	6.5	84.2
PC-Simp	0.241	0.000487	-	5.48	0.00802	-	0.0364	0.0433	-	132.6	245.5	-
AIVS	0.715	0.000638	8.75	4.11	0.00539	196.45	0.0513	0.0441	0.0666	17.2	44.6	1983.5
Potamias et al.	0.215	0.000599	4.28	3.47	0.00933	190.00	0.0478	0.0650	0.0511	0.00060	0.00070	0.00212



**Fig. 4.** Simplification results of an airplane, a glass and a toilet point cloud for simplification ratios  $\alpha = 0.25$  and  $0.5$  using APES and GP-based simplification.

HC and Potamias *et al.* are the only baselines with shorter runtimes than our method, and obtain maximum Hausdorff distances comparable to those obtained by our approach. However, as discussed in Sect. 2, tuning the user-specified HC parameters make striking a balance between feature preservation and retaining a sufficient density of points across the cloud relatively challenging. Moreover, there is no control over the size of the simplified cloud, as discussed by the authors [26] and in subsequent work [24]. We tuned this baseline to attempt to balance this trade-off, and whilst the HC-simplified clouds shown in Figs. 2 and 3 here, and Fig. 3 of the supplementary material, do have clearly preserved features (an observation supported by the high mean surface variation across all clouds), the density of points away from these areas is very low. This leads to inferior mesh reconstructions compared to our approach, as evidenced by the fact that we obtain superior mean Hausdorff distance compared to HC across all three clouds.

Since results and inference times for the Potamias *et al.* approach were provided by the author of the paper, we do not have knowledge of the exact details of their experimental setup, especially the time required in hours to train the model. As mentioned in Sect. 2, their learning-based approach demands huge datasets to train on, which not only increases the computational requirements but also limits their method’s generalizability. When compared to our method quantitatively, our method generally gives superior results, except for two of the nine error metric values. This is supported by the quality of their simplified point clouds and the corresponding reconstructions shown in Figs. 2 and 3 here, and in

Figs. 2, 3 and 5 of the supplementary material. Although their method performs best in the case of Lucy’s maximum Hausdorff distance, in reality their simplified cloud gives arguably the poorest qualitative reconstruction result amongst all of the other baselines. As expected, the inference time of their approach is the lowest of all the baselines, because of their neural network-based approach, which involves pre-training.

The WLOP baseline does not efficiently preserve the features and favours uniformly covering the domain of the original cloud. Therefore, the mean surface variation of the WLOP simplified clouds is lower, but overall the Hausdorff distances obtained from the reconstructed meshes are superior to those obtained by our method. However, it is noteworthy that on the largest and unarguably the most challenging point cloud, *Lucy*, our method achieves a superior mean Hausdorff distance as compared to all of the other techniques evaluated, including WLOP. Additionally, WLOP is significantly slower than our approach, as shown in Table 1. Our surface variation computation is currently performed on a CPU, therefore further improvements to the runtimes of our method shown could be achieved by re-implementing this in a GPU-compatible framework.

Overall, these results show that our approach provides a computationally efficient option for performing point cloud simplification in settings where the user wishes to strike a balance between preserving high fidelity around sharp features in the cloud, and ensuring that the simplified cloud covers the manifold defined by the original cloud with a sufficient density of points. This is important for generating reconstructions which resemble the original meshes, as is evident from visual inspection of the reconstruction results in Fig. 2 here and Fig. 3 of the supplementary material. In terms of surface reconstruction, our method clearly outperforms all of the other techniques for the *Dragon* (compare the tail, teeth, horns and the face detailing for all methods and additionally the curved body for HC) and the *Armadillo* (compare the ears, hands and feet across all the methods) and gives competitive results for *Lucy*, shown in Fig. 2 of the supplementary material. We highlight once again the poor surface reconstructions resulting from the Potamias *et al.* simplified clouds, compared to those obtained using all of the other baselines. Again, visual inspection of the simplification results for the noisy Armadillo in Fig. 3 demonstrates the balanced feature-sensitivity of our method in comparison to others. We experiment with more noise levels in the supplementary material (Sect. 2). Finally, from Fig. 4 we can see how the edge-sampling-based APES simplified clouds have several missing portions including object edges, whereas our method enhances the salient features and captures the overall object structure simultaneously. We do not provide corresponding surface reconstructions and hence quantitative results for this baseline because their low simplified point cloud sizes ( $N = 1,024$  and  $512$ ) and aforementioned missing areas will always result in open meshes.

The  $\mathcal{O}(M^3)$  and  $\mathcal{O}(M^2N)$  complexities associated with training and prediction respectively in the greedy inference scheme described in Sect. 3.3 allow for increased scalability compared to typical GP regression, in which inference has  $\mathcal{O}(N^3)$  complexity. The scalability of our approach is limited by the fact that,

as in a conventional exact GP, we have a storage demand associated with  $\mathbf{K}$  matrix which scales according to  $\mathcal{O}(N^2)$ . However, we can circumvent this issue when  $N$  is very large by simply using Algorithm 1 with a randomly selected subset of  $P$ . For *Armadillo* and *Dragon* we obtain the above results with just 25,000 randomly selected points. For a large point cloud such as *Lucy*, we obtain competitive results using a subset of just 40,000 points to run our simplification algorithm.

## 5.2 Point Cloud Registration

As discussed earlier, PC simplification has benefits for many downstream tasks, not solely surface reconstruction. In Table 2 we present registration results on some simplified clouds. We firstly translate and rotate the original, HC and GP-simplified clouds in the same fashion, before performing global and ICP point-to-point registration [3] with the *Open3D* package [36]; visualisations are available in the supplementary material (Fig. 4). Our GP-simplified cloud allows for quicker registration and leads to superior inlier RMSE.

**Table 2.** Inlier RMSE and time taken for global and ICP registration. Best results are in red, whilst second-best are in green.

	Inlier RMSE ( $\downarrow$ )		Time (s) ( $\downarrow$ )		
	Global ( $10^{-3}$ )	ICP ( $10^{-7}$ )	Global	ICP	Total
Original	4.76	4.08	0.017	1.448	1.465
HC	5.41	4.08	0.018	0.046	0.064
GP (ours)	3.91	4.08	0.017	0.040	0.057

## 6 Conclusion

In this work we have presented a novel, one-shot point cloud simplification algorithm capable of preserving both the salient features and the overall structure of the original point cloud. We reduce the cloud size by up to three orders of magnitude without the need for computationally intensive training on huge datasets. This is achieved via a greedy algorithm which iteratively selects points based on a selection criterion determined by modeling the surface variation over the original point cloud using Gaussian processes with kernels which operate on Riemannian manifolds. We show that our technique achieves competitive results and runtimes when compared to a number of relevant methods, outperforming all baselines tested in terms of mean Hausdorff distance on *Lucy*, the largest and

most complex point cloud we consider, consisting of approximately 14 million points. Our method can also be used to improve the computational efficiency of downstream tasks such as point cloud registration with no negative effects on the empirical performance.

**Future Work:** Whilst Hausdorff distance is a useful metric, it is not the ideal candidate for assessing the feature sensitivity of a simplification algorithm, as it tends to return lower errors for more evenly distributed clouds. Whilst out of the scope of this work, there is a clear need for a well-defined and widely adopted error metric for curvature-sensitive simplification. Currently, the best way to evaluate this is a qualitative visual inspection of the resulting point cloud (or reconstructed mesh). This view is supported by the fact that some recent works employ user studies to evaluate their feature-preserving approaches [28].

In this work we study the setting where we enforce the restriction that the simplified cloud be a subset of the original; as discussed in Sect. 3.3, a greedy inference scheme is appropriate in this setting. However, this assumption could be relaxed and sparse GPs can be used to perform continuous optimization of the inducing points across the point cloud [15]. This would allow occluded as well as outlier-ridden extremely noisy point clouds, where the original observations do not necessarily lie on the true surface of the manifold, to be denoised and/or simplified.

## References

1. Berger, M., et al.: State of the art in surface reconstruction from point clouds. In: Eurographics 2014-State of the Art Reports, vol. 1, no. 1, pp. 161–185 (2014)
2. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Visual Comput. Graphics* **5**(4), 349–359 (1999)
3. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes. In: *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611, pp. 586–606. SPIE (1992)
4. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* **112**(518), 859–877 (2017)
5. Borovitskiy, V., Terenin, A., Mostowsky, P., et al.: Matérn Gaussian processes on Riemannian manifolds. *Adv. Neural. Inf. Process. Syst.* **33**, 12426–12437 (2020)
6. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: *Numerical Geometry of Non-rigid Shapes*. Springer (2008)
7. Cignoni, P., Montani, C., Scopigno, R.: A comparison of mesh simplification algorithms. *Comput. Graphics* **22**(1), 37–54 (1998)
8. Cignoni, P., Rocchini, C., Scopigno, R.: METRO: measuring error on simplified surfaces. In: *Computer Graphics Forum*, vol. 17, pp. 167–174. Blackwell Publishers (1998)
9. Fernandes, D., et al.: Point-cloud based 3D object detection and classification methods for self-driving applications: a survey and taxonomy. *Inf. Fusion* **68**, 161–191 (2021)
10. Fortuin, V., Dresdner, G., Strathmann, H., Rätsch, G.: Sparse Gaussian processes on discrete domains. *IEEE Access* **9**, 76750–76758 (2021)

11. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 209–216 (1997)
12. Guerrero, P., Kleiman, Y., Ovsjanikov, M., Mitra, N.J.: PCPNet learning local shape properties from raw point clouds. In: Computer Graphics Forum, vol. 37, pp. 75–85. Wiley Online Library (2018)
13. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Mesh optimization. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 19–26 (1993)
14. Huang, H., Li, D., Zhang, H., Ascher, U., Cohen-Or, D.: Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graphics (TOG)* **28**(5), 1–7 (2009)
15. Hutchinson, M., Terenin, A., Borovitskiy, V., Takao, S., Teh, Y., Deisenroth, M.: Vector-valued Gaussian processes on Riemannian manifolds via gauge independent projected kernels. *Adv. Neural. Inf. Process. Syst.* **34**, 17160–17169 (2021)
16. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing, vol. 7 (2006)
17. Kazhdan, M., Hoppe, H.: Screened Poisson surface reconstruction. *ACM Trans. Graphics (ToG)* **32**(3), 1–13 (2013)
18. Lalchand, V., Faul, A.: A fast and greedy subset-of-data (SoD) scheme for sparsification in Gaussian processes. arXiv preprint [arXiv:1811.07199](https://arxiv.org/abs/1811.07199) (2018)
19. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. In: ACM SIGGRAPH 2005 Papers, pp. 659–666 (2005)
20. Levoy, M., Gerth, J., Curless, B., Pull, K.: The Stanford 3D scanning repository, vol. 5, no. 10 (2005). <https://graphics.stanford.edu/data/3Dscanrep/>
21. Li, L., Schemenauer, N., Peng, X., Zeng, Y., Gu, P.: A reverse engineering system for rapid manufacturing of complex objects. *Robot. Comput.-Integr. Manuf.* **18**(1), 53–67 (2002)
22. Lipman, Y., Cohen-Or, D., Levin, D., Tal-Ezer, H.: Parameterization-free projection for geometry reconstruction. *ACM Trans. Graphics (TOG)* **26**(3), 22–es (2007)
23. Liu, H., Ong, Y.-S., Shen, X., Cai, J.: When Gaussian process meets big data: a review of scalable GPs. *IEEE Trans. Neural Netw. Learn. Syst.* **31**(11), 4405–4423 (2020)
24. Lv, C., Lin, W., Zhao, B.: Approximate intrinsic voxel structure for point cloud simplification. *IEEE Trans. Image Process.* **30**, 7241–7255 (2021)
25. Moenning, C., Dodgson, N.A.: A new point cloud simplification algorithm. In: Proceedings International Conference on Visualization, Imaging and Image Processing, pp. 1027–1033 (2003)
26. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient simplification of point-sampled surfaces. In: 2002 IEEE Visualization, VIS 2002, pp. 163–170. IEEE (2002)
27. Pieraccini, M., Guidi, G., Atzeni, C.: 3D digitizing of cultural heritage. *J. Cult. Herit.* **2**(1), 63–70 (2001)
28. Potamias, R.A., Bouritsas, G., Zafeiriou, S.: Revisiting point cloud simplification: a learnable feature preserving approach. In: ECCV 2022, Part II, pp. 586–603. Springer (2022)
29. Qi, J., Hu, W., Guo, Z.: Feature preserving and uniformity-controllable point cloud simplification on graph. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 284–289. IEEE (2019)
30. Rasmussen, C., Williams, C.: Gaussian processes for machine learning. In: Gaussian Processes for Machine Learning (2006)



31. Rusu, R.B.: Semantic 3D object maps for everyday manipulation in human living environments. *KI-Künstliche Intell.* **24**, 345–348 (2010)
32. Thomas, H., Goulette, F., Deschaud, J.-E., Marcotegui, B., LeGall, Y.: Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In: 2018 International conference on 3D vision (3DV), pp. 390–398. IEEE (2018)
33. Wu, C., Zheng, J., Pfrommer, J., Beyerer, J.: Attention-based point cloud edge sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5333–5343 (2023)
34. Wu, Z., et al.: 3D shapenets: a deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920 (2015)
35. Xiao, D., et al.: Estimating reference bony shape models for orthognathic surgical planning using 3D point-cloud deep learning. *IEEE J. Biomed. Health Inform.* **25**(8), 2958–2966 (2021)
36. Zhou, Q.-Y., Park, J., Koltun, V.: Open3D: a modern library for 3D data processing. [arXiv:1801.09847](https://arxiv.org/abs/1801.09847) (2018)



# GSTran: Joint Geometric and Semantic Coherence for Point Cloud Segmentation

Abiao Li<sup>1</sup>, Chenlei Lv<sup>2</sup>, Guofeng Mei<sup>3</sup>, Yifan Zuo<sup>1</sup>, Jian Zhang<sup>4</sup>,  
and Yuming Fang<sup>1</sup>(✉)

<sup>1</sup> Jiangxi University of Finance and Economics, Nanchang 330013, China  
leo.fangyuming@foxmail.com

<sup>2</sup> Shenzhen University, Shenzhen 518060, China  
chenleilv@mail.bnu.edu.cn

<sup>3</sup> Fondazione Bruno Kessler, 38123 Trento, Italy

<sup>4</sup> University of Technology Sydney, Sydney 2007, Australia  
jian.zhang@uts.edu.au

**Abstract.** Learning meaningful local and global information remains a challenge in point cloud segmentation tasks. When utilizing local information, prior studies indiscriminately aggregates neighbor information from different classes to update query points, potentially compromising the distinctive feature of query points. In parallel, inaccurate modeling of long-distance contextual dependencies when utilizing global information can also impact model performance. To address these issues, we propose GSTran, a novel transformer network tailored for the segmentation task. The proposed network mainly consists of two principal components: a local geometric transformer and a global semantic transformer. In the local geometric transformer module, we explicitly calculate the geometric disparity within the local region. This enables amplifying the affinity with geometrically similar neighbor points while suppressing the association with other neighbors. In the global semantic transformer module, we design a multi-head voting strategy. This strategy evaluates semantic similarity across the entire spatial range, facilitating the precise capture of contextual dependencies. Experiments on ShapeNetPart and S3DIS benchmarks demonstrate the effectiveness of the proposed method, showing its superiority over other algorithms. The code is available at <https://github.com/LAB123-tech/GSTran>.

**Keywords:** Point cloud segmentation · Local geometric transformer · Global semantic transformer

## 1 Introduction

3D point cloud segmentation is a crucial topic in computer vision and graphics, with widespread applications in autonomous driving, simultaneous localization

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78456-9\\_29](https://doi.org/10.1007/978-3-031-78456-9_29).

and mapping (SLAM), augmented reality, and virtual reality. Limited by the weak discriminative capability of handcraft features, traditional solutions fail to achieve satisfactory segmentation performance. Fortunately, benefited from the development of deep learning, the performance of segmentation has been significantly improved to support high-level semantic analysis [14].

Recent efforts [8, 16] have shown promising results in point cloud processing by integrating multi-scale neighbor features to enhance the capability of feature analysis. However, repeating similar contextual features at various scales seems redundant and computationally expensive, especially for hierarchical architectures. Subsequent studies [17, 22, 24], advanced the refinement of point features by encoding the detailed geometric description information in the local region of the point cloud. Nonetheless, these algorithms simply stack the geometric and coordinate information of the point cloud and take the stacked result as input to the network. These approaches fail to fully exploit geometric properties, as geometric information is likely to be lost during learning. Large language model (LLM) based methods [10, 32] have also demonstrated their effectiveness in point clouds. The main idea of these approaches is to render the 3D point cloud as a set of multi-view 2D images for semantic parsing. But, LLM, being a type of generalized model, lacks the ability to perceive the internal structure of point clouds [13]. For accurate geometric structure determination of point clouds, the LLM still have limitations.

With the breakthrough of transformer in the fields of natural language processing and computer vision, some algorithms [27, 28] consider incorporating geometric information of the point cloud into the self-attention mechanism to execute segmentation. PointTr [27] introduces a geometric-aware module that models the local geometric relationships of point clouds by constructing neighborhood graph structures. PointGT [28] decouples the local neighborhoods and utilizes a bi-directional cross-attention mechanism to merge the edge and inside components of the local features. Nonetheless, the weights assigned to neighbor points rely solely on learning. The model struggles to prioritize neighbor points belonging to the same category as the query point. As a result, the descriptive power of point features is compromised, potentially leading to erroneous segmentation results at the boundary.

Although deploying transformer in the local regions of point clouds plays a positive role in capturing geometric information, its receptive field remains limited. In point clouds, points that are semantically related to query points may be located far apart. Therefore, certain research studies [11, 15, 31] employ transformer to compute global similarity for each point. Such approaches treat all points as neighbor points of the query point. They employ self-attention to capture long-range contextual dependencies among points of the same class. This contributes to the enhancement of semantic understanding. However, the strong similarity between points computed with self-attention does not guarantee that these points belong to the same category.

In light of the above analysis, fully exploiting local geometric features and accurately capturing long-range dependencies hold significant importance for

transformer to understand point clouds. Based on this, we propose a novel transformer architecture for point cloud segmentation, named GSTran. It consists of two crucial transformer modules: (i) a local geometric transformer and (ii) a global semantic transformer. In the local geometric transformer, we thoroughly investigate the geometric disparity in the local region to quantify the significance of each neighbor point. Specially, we explicitly compute the distance from the query point to the tangent plane of its corresponding neighbor points. In general, the query point tends to be closer to the tangent plane of its geometrically similar neighbors. As a result, greater significance is attributed to these neighbor points, leading to enhanced segmentation outcomes for boundaries within the point cloud.

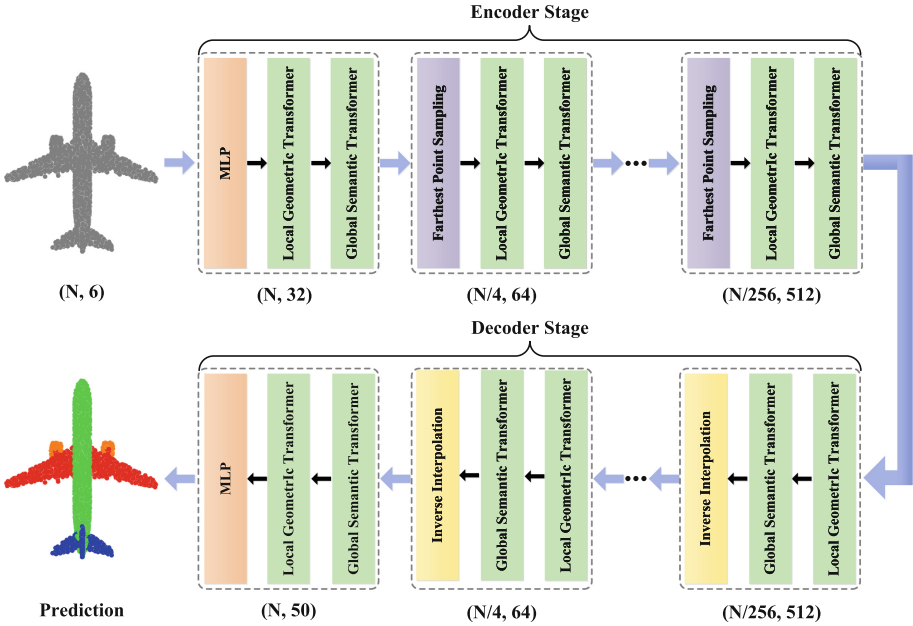
In the global semantic transformer, a multi-head voting strategy is introduced to facilitate the preservation of meaningful long-range dependencies. Essentially, this involves employing a multi-head attention mechanism to compute multiple global similarities for each point. However, unlike multi-head attention mechanisms, we further extract the shared information from multiple global similarities to generate a global mask. By leveraging the global mask, we can more accurately compute the long-distance similarity between points in the point cloud. Equipped with both transformer modules, GSTran effectively infuses features with rich local geometric structures and comprehensive global semantic context. The main contributions of our paper are summarized as follows:

- We introduce a local geometric transformer module that leverages the geometric disparity within the local point cloud. This module assigns higher weights to neighbor points that exhibit similar geometric structure to the query point. This enhancement improves the model’s ability to distinguish target boundaries.
- We design a multi-head voting-based global semantic transformer module to capture semantically aligned contextual dependencies beyond local regions. By leveraging this module, we can enhance the accuracy of computing long-distance similarity between points within the point cloud.
- We present the performance evaluation of GSTran on both ShapeNetPart and S3DIS benchmarks to demonstrate the efficacy of our approach in addressing segmentation tasks.

## 2 Methodology

### 2.1 Overview

The architecture of the proposed model, as shown in Fig. 1, features a hierarchical framework consisting of symmetric encoder and decoder stages. There are five stages in the encoder. The point cloud sampling rate between two adjacent stages is  $1/4$ , and the channel expansion rate is 2. Within the first stage of the encoder, the MLP layer projects the point cloud data with coordinate and normal vector information into higher dimensional feature. Subsequently,

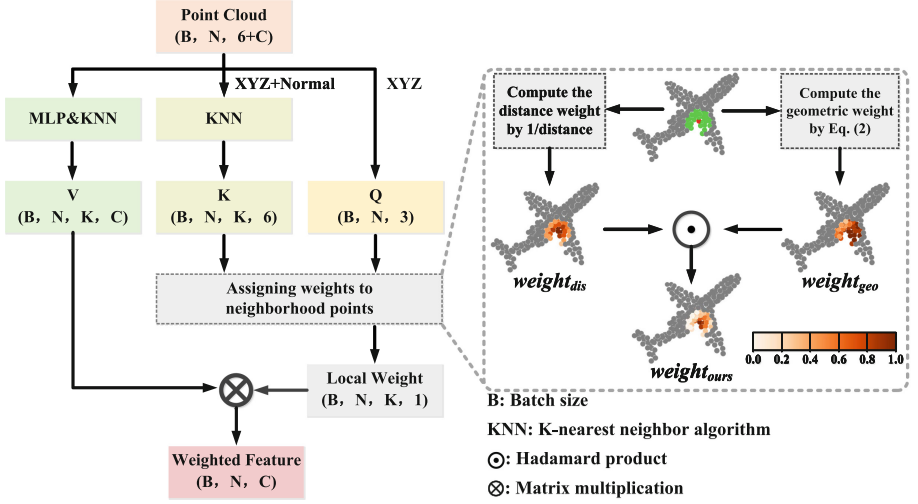


**Fig. 1.** Overview of the proposed model. In the encoder and decoder stages, the transformer structure serves as the primary feature aggregator throughout the network. MLP: Multi-layer perception. N: The number of points in the point cloud.

two sequential modules - local geometric and global semantic transformer - are sequentially applied for feature extraction at different scales. The features are progressively downsampled with channel expanding in the following four stages. This process continues until the sampling rate reaches  $1/256$  and the channel expands to 512. The decoder stage follows a similar structure but utilizes inverse interpolation [16] for progressive upsampling. Details about the local geometric and global semantic transformer modules are given in Sect. 2.2 and Sect. 2.3, respectively.

### 2.2 Local Geometric Transformer Module

The local geometric transformer module is designed to achieve discriminative feature extraction. It achieves this by investigating the geometric disparity within the local region. As depicted in Fig. 2, given input point cloud  $\chi = \{x_i | i = 1, 2, \dots, N\} \in \mathcal{R}^{N \times (6+C)}$ , each point  $x_i$  is defined by its position coordinate  $p_i \in \mathcal{R}^3$ , normal vector  $n_i \in \mathcal{R}^3$  and feature  $f_i \in \mathcal{R}^C$ . We separate the coordinate information as the  $Q$  vector. At the same time, corresponding neighbor sets are constructed for each point in both Euclidean space and feature space using the KNN algorithm, represented by  $K$  and  $V$ , respectively. Subsequently, corresponding weights are assigned to the neighbor points to quantify their importance with respect to the query points.



**Fig. 2.** Structure of the local geometric transformer module. We visualize the local weight on an airplane, with a red query point located on the wing. In  $weight_{ours}$ , the weights of neighbor points in the wing decay slowly as the distance increases. However, the weights of other neighbor points decay rapidly.

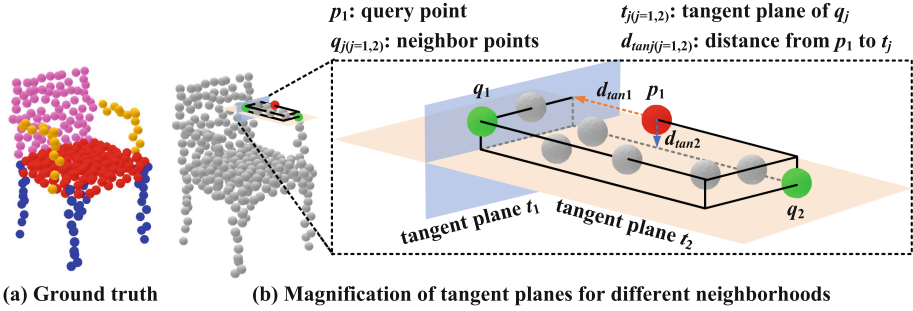
As shown in the dashed box of Fig. 2, given a red query point  $p_i$  (located at the wing) and its green neighbor points  $h = \{q_1, \dots, q_k\} \in \mathcal{R}^{k \times 3}$ , we compute the reciprocal of the Euclidean distance to obtain the distance weights denoted as  $weight_{dis}$ . Clearly, smaller weights are assigned to distant neighbor points, but the weights remain consistent for neighbor points of different classes, such as fuselage and wing. With only distance weights, the query point may indiscriminately aggregate information from neighbor points of various classes. This may compromise the descriptive power of the query point.

Therefore, we use the following formula [21] to characterize the geometric importance of neighbor points with respect to the query point.

$$d_{tan} = (p_i - q_j)n_j, \forall j \in G_i, \quad (1)$$

where  $p_i$  is the coordinate of  $i$ -th query point,  $q_j$  denotes the coordinate of  $j$ -th neighbor point corresponding to  $p_i$ .  $G_i$  is the index set of the local group centered on  $p_i$ .  $n_j$  is the normal vector corresponding to  $q_j$ . Essentially,  $d_{tan}$  signifies the distance from the query point to the tangent plane of the neighbor points, as shown in Fig. 3. (Note: to clearly explain  $d_{tan}$ , we choose a chair target for illustration.) The tangent plane of neighbor points belonging to the same class as the query point is close to the query point. This proximity implies that the importance attached to this neighbor point should be large. Based on  $d_{tan}$ , we quantify the weight in the geometry using an exponential function as follows.

$$weight_{geo} = \exp(-d_{tan}) \quad (2)$$



**Fig. 3.** Illustration of the distance  $d_{tan1}$  and  $d_{tan2}$  from  $p_1$  to the tangent plane of  $q_1$  and  $q_2$ , respectively. Although the Euclidean distance from  $p_1$  to both  $q_1$  and  $q_2$  remain the same,  $p_1$  is closer to  $t_2$ , signifying that  $q_2$  holds greater significance than  $q_1$ .

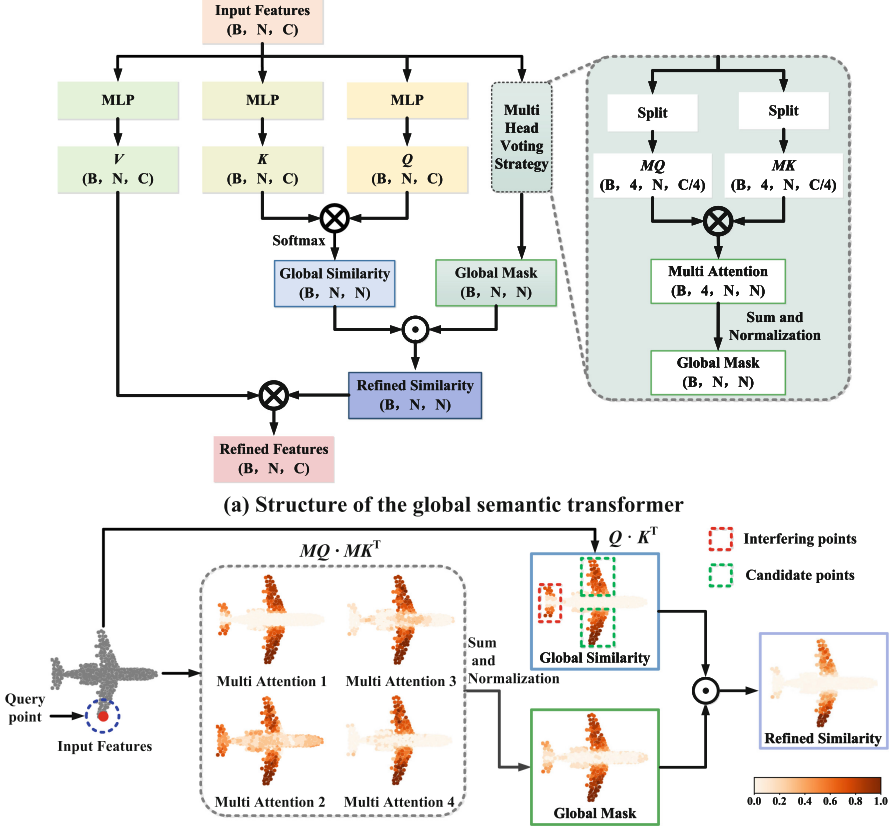
Eq. (2) shows that the weight is inversely proportional to the value of  $d_{tan}$ . It can be seen from Fig. 2 that the  $weight_{geo}$  assigns higher weights to points on the wings and lower weights to points on the fuselage. However, the weights assigned to the neighbor points from the wings are nearly equal in geometric weight. Following the principle that larger distances correspond to fewer dependencies between points, we consider combining the distance and geometric weights through a Hadamard product to obtain the local weight of ours. It can be observed that the weights of neighbor points on the wing in  $weight_{ours}$  exhibit a slow decay rate with increasing distance. This is in contrast to points located on the fuselage, where the weights decrease more rapidly. High response weight values are predominantly distributed in the wings. Finally, the weighted features are obtained by multiplying the local weight with the  $V$  vector.

### 2.3 Global Semantic Transformer Module

The output of the local geometric transformer module exhibits powerful discriminative capabilities for local features. However, the relations captured by this module are restricted to local structures within the point cloud. Certain approaches [6, 11, 31] employ transformer to calculate the global similarity between each pair of points in the entire point cloud. Then, the global similarity, along with the point cloud features, is multiplied to perform feature aggregation. Mathematically, the formula is as follows:

$$y_i = \sum_{f_j \in \chi} \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V, \quad Q = \varphi(f_i), K = \psi(f_j), V = \phi(f_j) \quad (3)$$

where  $f_i$  denotes the feature of the  $i$ -th query point, and  $f_j$  represents the feature of the  $j$ -th point within point cloud  $\chi$ . The symbols  $\varphi$ ,  $\psi$ , and  $\phi$  represent three MLP operations conducted on the feature of the point clouds to obtain the query  $Q$ , key  $K$ , and value  $V$  vectors, respectively.  $d_k$  is the dimension of the feature.



**Fig. 4.** Overview of the global semantic transformer module. We visualize the refined similarity on an airplane, with a red query point located at the wing. In the refined similarity generated by our method, high response weights are exclusively assigned to points belonging to the wing section. (Color figure online)

A drawback of using  $QK^T$  in Eq. (3) to compute global similarity is that the obtained global similarity may not be accurate. To be specific, points that show strong similarity to query points may belong to different classes. An example of the global similarity is shown in Fig. 4(b). Given a query point within the wing section (marked in red), we visualize the global similarity corresponding to this query point. It is evident that the global similarity exhibits high response weights for points in both the wing and tail sections. However, the weights of the points in the wing section are the only ones truly relevant. We designate points in the tail as interfering points and points in the wings as candidate points.

To accurately compute the long-distance similarity between points within the point cloud, we devise a multi-head voting strategy that is incorporated into the global semantic transformer module. The corresponding structure is illustrated in Fig. 4. Given the features of the input point cloud, three MLP operations are



performed to obtain the  $Q$ ,  $K$ , and  $V$  vectors, respectively. Then, the matrix multiplication between  $Q$  and the transposition of  $K$  is performed, followed by the softmax function to generate the global similarity for each point. In order to mitigate the effect of interfering points, we further refine the global similarity by generating a global mask using a multi-head voting strategy. The details are depicted within the dashed box in Fig. 4(a).

Specifically, we split the channels of the point cloud to obtain multi-head representations of the feature, denoted as  $MQ$ . Similarly, we do the same to obtain  $MK$ . We proceed by performing matrix multiplication between  $MQ$  and the transpose of  $MK$  to produce multiple attentions. Corresponding results are shown in the dashed box in Fig. 4(b). It can be observed that the weights of candidate points consistently exhibit strong response in multi-attention 1 to 4. However, the weights of the interfering points exhibit randomness. For instance, in multi-attention 1 and 2, the weights of the interfering points are large, whereas in multi-attention 3 and 4, they are small. This inconsistency leads to uncertainty in selecting the optimal attention. To mitigate the influence of randomness, we propose to summarize multiple attentions and apply normalization to obtain a global mask. It can be observed that in the global mask, the weights of the interfering points are reduced to some extent. To further reduce the weights of interfering points, we consider subjecting the global similarity and global mask to Hadamard product operations to generate refined similarity. In the refined similarity, the weights of the interfering points are significantly reduced, while the weights of the candidate points still exhibit strong responses. Finally, the refined similarity is multiplied with the vector  $V$  to obtain the refined features, which fulfill the aggregation of global information for each point.

## 3 Experiments

### 3.1 Experimental Setting

We demonstrate the effectiveness of the proposed model in various point cloud segmentation tasks. Specifically, we utilize the S3DIS dataset for 3D semantic segmentation and the ShapeNetPart dataset for 3D part segmentation. Experiments were conducted on an Ubuntu system equipped with two NVIDIA GTX 2080Ti GPUs. We employed the Adam optimizer with momentum and weight decays set to 0.9 and 0.0001, respectively. For the S3DIS dataset, we trained for 60,000 iterations, starting from an initial learning rate of 0.5. This rate drops by a factor of 10 at steps of 30,000 and 50,000. For the ShapeNetPart dataset, we trained for 200 epochs. The initial learning rate is set to 0.05 and reduced by a factor of 10 at epochs 100 and 150.

### 3.2 3D Semantic Segmentation and Part Segmentation

**Semantic Segmentation.** The S3DIS dataset comprises 271 scenes from 6 indoor areas, and each point labeled among 13 categories. Since the S3DIS

**Table 1.** Semantic segmentation results on S3DIS dataset.

Method	mIoU	mAcc	OA	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointWeb [29]	66.7	76.2	87.3	93.5	94.2	80.8	52.4	41.3	64.9	68.1	71.4	67.0	50.3	62.7	62.2	58.5
PointAttn. [5]	66.3	77.3	88.9	94.3	97.0	76.0	64.7	53.7	59.2	58.8	72.4	69.2	42.6	60.8	54.1	59.0
SCFNet [4]	71.6	82.7	88.4	93.3	96.4	80.9	64.9	47.4	64.5	70.1	81.6	71.4	64.4	67.2	<b>67.5</b>	60.9
CBL [23]	73.1	79.4	89.6	94.1	94.2	85.5	50.4	<b>58.8</b>	70.3	<b>78.3</b>	75.7	75.0	71.8	74.0	60.0	62.4
PointTrans. [30]	73.5	81.9	90.2	94.3	97.5	84.7	55.6	58.1	66.1	78.2	77.6	74.1	67.3	71.2	65.7	<b>64.8</b>
BAAFNet [18]	72.2	83.1	88.9	93.3	96.8	81.6	61.9	49.5	65.4	73.3	72.0	83.7	67.5	64.3	67.0	62.4
DPFA [2]	61.7	61.6	89.2	<b>94.6</b>	<b>98.0</b>	79.2	40.7	36.6	52.2	70.8	65.9	74.7	27.7	49.8	51.6	60.6
RepSurf [19]	74.1	82.6	90.8	93.8	96.3	<b>85.6</b>	62.5	52.5	67.4	75.3	73.9	82.1	71.5	73.3	65.1	61.8
Ours	<b>74.9</b>	<b>83.5</b>	<b>91.3</b>	93.2	96.1	85.1	<b>65.1</b>	50.7	<b>71.2</b>	73.3	<b>79.1</b>	<b>84.2</b>	<b>71.9</b>	<b>73.9</b>	<b>67.5</b>	62.4

Note: bold font indicates best result.

**Table 2.** Part segmentation results on ShapeNetPart dataset.

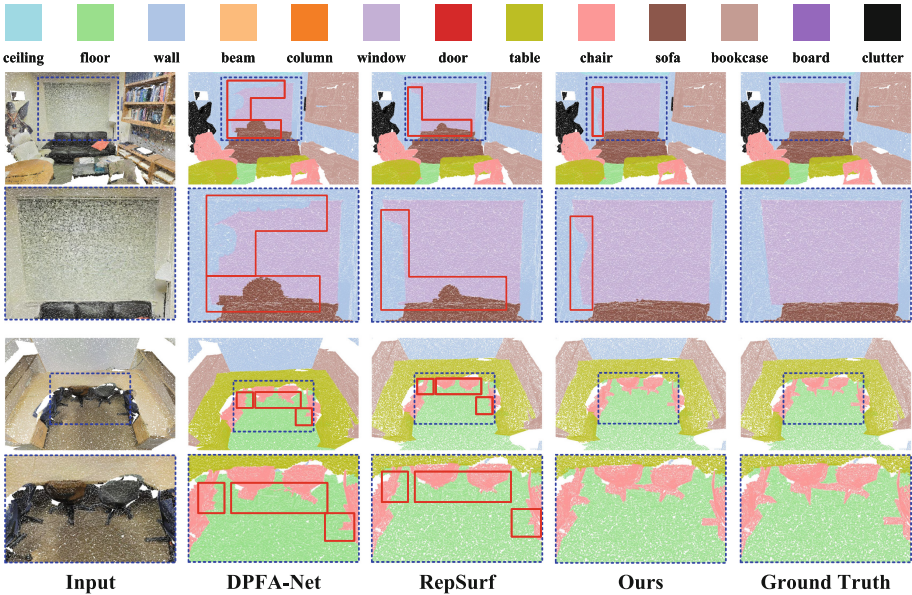
Methods	Ins.mIoU	Cat.mIoU	air.	bag	cap	car	cha.	ear.	gui.	kni.	lam.	lap.	mot.	mug	pis.	roc.	ska.	tab.
PointAttn. [5]	85.9	84.1	83.3	86.1	85.7	80.3	90.5	82.7	91.5	88.1	85.5	95.9	77.9	95.1	84.0	64.3	77.6	82.8
PointASNL [26]	86.1	83.4	84.1	84.7	87.9	79.7	92.2	73.7	91.0	87.2	84.2	95.8	74.4	95.2	81.0	63.0	76.3	83.2
PointMLP [12]	86.1	84.6	83.5	83.4	87.5	80.5	90.3	78.2	92.2	88.1	82.6	96.2	77.5	<b>95.8</b>	85.4	<b>64.6</b>	<b>83.3</b>	84.3
APES [25]	85.8	83.9	85.3	85.8	88.1	<b>81.2</b>	90.6	74.0	90.4	<b>88.7</b>	85.1	95.8	76.1	94.2	83.1	61.1	79.3	84.2
PointTran. [30]	86.5	83.7	85.8	85.3	86.8	77.2	90.5	82.0	90.8	87.5	85.2	96.3	75.4	93.5	83.8	59.7	77.5	82.5
PointGT [9]	85.8	84.2	84.3	84.5	88.3	80.9	91.4	78.1	92.1	88.5	85.3	95.9	77.1	95.1	84.7	63.3	75.6	81.4
PCT [7]	86.4	83.1	85.0	82.4	89.0	<b>81.2</b>	91.9	71.5	91.3	88.1	86.3	95.8	64.6	95.8	83.6	62.2	77.6	83.7
LGGCM [3]	86.7	84.8	85.1	85.9	<b>90.3</b>	80.8	91.6	75.4	<b>92.7</b>	88.1	86.5	96.1	77.0	94.2	84.5	63.6	80.2	84.3
Ours	<b>87.5</b>	<b>85.6</b>	<b>86.1</b>	<b>87.2</b>	88.1	79.4	<b>92.4</b>	<b>82.3</b>	92.0	88.4	<b>86.9</b>	<b>96.7</b>	<b>78.7</b>	95.6	<b>85.8</b>	63.8	79.3	<b>85.3</b>

Note: air.: airplane. cha.: chair. ear.: ear-phone. gui.: guitar. kni.: knife. lam.: lamp. lap.: laptop. mot.: motorbike. pis.: pistol. roc.: rocket. ska.: skateboard. tab.: table.

dataset does not provide normal vector information, it becomes necessary for us to compute the normal vector before training. In this step, we refer to the method outlined in [1]. Here, we extract the eigenvector corresponding to the minimum eigenvalue of the point cloud. This is achieved by performing singular value decomposition on the covariance matrix of the point cloud. At the same time, we can precompute the normal vector for each point before training, storing them in the dataset to reduce running time.

During the training process, the computational complexity of the global semantic transformer module being  $O(n^2)$ , where  $n$  represents the number of points processed. However, in the entire algorithm process, only the first global transformer module in the encoding stage and the last one in the decoder stage handle a relatively large number of points. The other transformer modules, due to the UNet [20] structure used in the algorithm, process a smaller number of points. Moreover, we use block-wise training strategy [16] to ensure that  $n$  is not excessively large, thereby improving running efficiency. Specially, each room is divided into  $2\text{ m} \times 2\text{ m}$  blocks, from which 4096 points are randomly sampled for training. During testing, we employ six-fold cross-validation for evaluation. This approach involves using all points in the scene for testing purposes. For evalu-

ation metrics, we use mean instance IoU (mIoU), mean class accuracy (mAcc), and overall pointwise accuracy (OA).

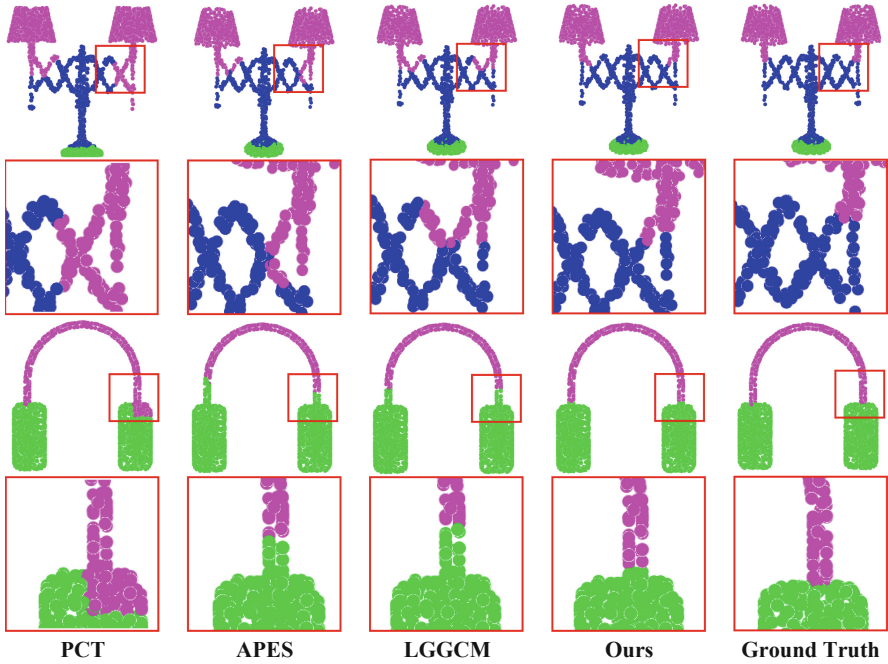


**Fig. 5.** Visualization of segmentation results for different methods on S3DIS. The red box indicates the region where the segmentation error occurs. (Color figure online)

As shown in Table 1, compared with other algorithms, the proposed algorithm achieves satisfied results in mIoU, mAcc, and OA. The superior performance over other transformer architecture [30] (+1.4% mIoU, 1.6% mAcc, 1.1% OA) proves the importance of incorporating the geometric relationships in semantic segmentation. Meanwhile, we compare the segmentation results of our algorithm with several algorithms in various scenarios on the S3DIS dataset. The corresponding results are shown in Fig. 5. In scene one (top row), compared to other algorithms, our proposed algorithm does not show significant errors in the sofa segmentation results. Due to the fact that windows are nearly embedded within walls, both our method and other algorithms suffer from certain shortcomings in the segmentation results of windows. But, the segmentation results of our method for the window are the closest to the ground truth. In scenario two (third row), our algorithm effectively improves the segmentation results of chairs, especially at the junction areas with the floor.

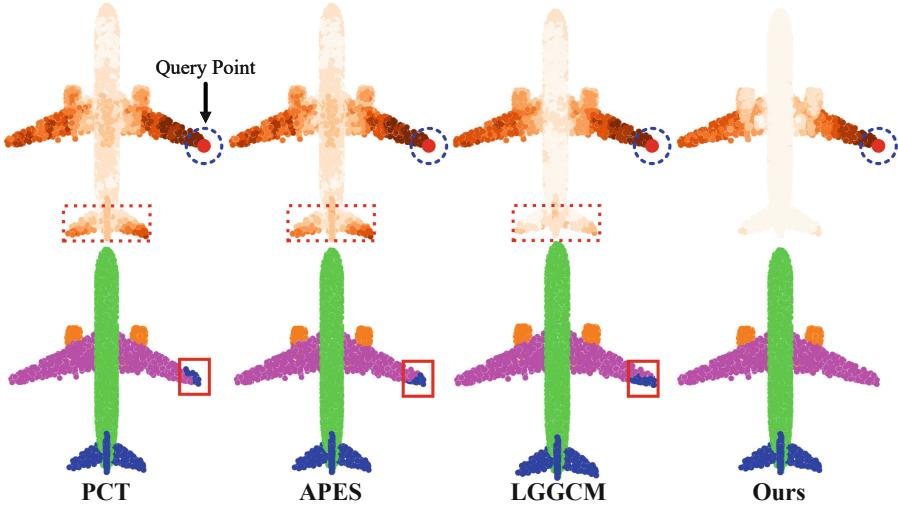
**Part Segmentation.** The ShapeNetPart dataset consists of 16, 880 3D models categorized into 16 shape categories with 50 different parts. For each point cloud object, 2048 points are uniformly sampled from its surface for both the training and testing phases. The data augmentation strategy we applied involves random

scaling of the object within a range of 0.8 to 1.25, coupled with arbitrary rotation along any coordinate axis. We evaluate the performance of our method using Intersection over Union (IoU) for each category. Additionally, we calculate the average IoU of all instances (Ins.mIoU) and the average IoU of all categories (Cat.mIoU) for the entire dataset.



**Fig. 6.** Visualization of segmentation results across various methods on ShapeNetPart, with areas of inaccurate segmentation highlighted by a red box. (Color figure online)

Table 2 presents the performance comparison of various algorithms on the ShapeNetPart dataset. Although the performance is quite saturated, our method achieves the best performance as measured by Ins.mIoU and Cat.mIoU. Compared to PCT [7], which also employs the concept of transformer, our algorithm achieves 1.1% improvement in Ins.mIoU. In addition, it achieved a 2.5% improvement in cat.mIoU, further demonstrating the effectiveness of our approach. Meanwhile, our method achieves the best performance on 9 out of 16 categories, such as airplane, bag and chair. Since the geometric discrepancies exhibited in the local regions of these categories are distinct. This property facilitates the local geometric transformer module in extracting information from local neighbor points. The corresponding segmentation results are shown in Fig. 6. It can be observed that other algorithms display significant segmentation errors at the boundaries of various components in the lamp and headphone targets. In contrast, the proposed algorithm exhibits minimal segmentation errors. At the



**Fig. 7.** Visualization of the global similarity and the corresponding segmentation results from various methods. The red dashed box indicates interfering points with high response weights. The red solid line box indicates the incorrect segmentation result. (Color figure online)

same time, the global similarity computed in our method achieves a high level of accuracy compared to other algorithms, as shown in Fig. 7. In other algorithms, high response weights are assigned not only to points belonging to the wing part, but also to interfering points in the tail part. In contrast, in the proposed algorithm, high response weights are assigned exclusively to points belonging to the wing part. Moreover, no segmentation error is detected in the wing section in our method.

### 3.3 Ablation Study

**Effects of Different Components.** To further illustrate the validity of the transformer modules in our method, we design an ablation study on the ShapeNetPart dataset as shown in Table 3. It can be observed from model A and B that the improvement is minimal when replacing the distance weights with geometric weights alone. It suggests that while geometric weight diminishes the effect of neighbor points with geometric disparity to the query point, the model struggles to focus on meaningful neighbor points. This is evident since the weights assigned to neighbor points that share similar geometry with the query point are nearly equal. In contrast, model C shows a significant performance improvement. Since only neighbor points that share similar geometry and close to the query point are assigned large weights.

Furthermore, from the results of model D and E, it is evident that modeling global dependencies is crucial for point cloud segmentation. Also, we can see that

**Table 3.** Ablation study on ShapeNetPart.

Models	Local Geometric Transformer		Global Semantic Transformer		Ins.mIoU	Cat.mIoU
	Dis.weight	Geo.weight	Glo.similarity	Glo.mask		
A	✓				85.1	84.2
B		✓			85.5	84.5
C	✓	✓			86.6	85.0
D	✓	✓	✓		87.0	85.3
E	✓	✓		✓	87.2	85.4
F	✓	✓	✓	✓	<b>87.5</b>	<b>85.6</b>

Note: Dis.weight: distance weight. Geo.weight: geometric weight. Glo.similarity: global similarity. Glo.mask: global mask.

**Table 4.** Effect of different neighbor size settings.

$k$	8	16	24	32	48
Ins.mIoU	86.5	87.2	<b>87.5</b>	87.4	87.2
Cat.mIoU	84.7	85.3	<b>85.6</b>	85.3	85.1

**Table 5.** Effect of different head number settings.

Heads	1	2	4	6	8
Ins.mIoU.	87.1	87.3	<b>87.5</b>	87.4	87.2
Cat.mIoU.	85.2	85.4	<b>85.6</b>	85.5	85.3

solely employing the global mask generated by the multi-head voting strategy results in superior performance compared to solely using global similarity. At last, the optimal accuracy is achieved when the global mask is combined with the global similarity. Since the weight information belonging to interfering points is filtered out in the refined similarity.

**Effect of Different Neighborhood Size.** The number of neighbor points becomes a crucial parameter when employing the local geometric transformer module. It determines the size of the receptive field for the local point cloud region. We test our model on the ShapeNetPart benchmark with various settings to ascertain the optimal value. As shown in Table 4, the performance improves as the parameters  $k$  increase. This suggests that expanding the receptive field by considering more neighbor points enhances the model’s ability to capture relevant features and contexts. However, further increasing the value of  $k$  may cause the performance of the model to degrade. Because it may introduce some geometrically similar but irrelevant point information, which could impact the local geometric transformer module’s ability to extract valid features.

**Table 6.** Investigation of different operators.

Operators	Concat	Summation	Average	Hadamard product
Cat.mIoU	85.3	85.3	85.4	<b>85.6</b>
Ins.mIoU	87.0	87.1	87.2	<b>87.5</b>

**Effects of Different Head Number.** We test our model on ShapeNetPart to evaluate the impact of different head number settings on the model performance. The relevant results are shown in Table 5. When the number of attention heads is set to one, the multi-head voting strategy fails to reduce the weight of interfering points. As a result, interference points may still exhibit high response weight in refined similarity. Meanwhile, As the number of heads increases, the model’s performance gradually improves. Since multiple heads aid the model to preserve meaningful global information. However, the performance deteriorates when the number of heads become larger.

**Effects of Different Operator.** We employ different operators that integrate geometric weights within the local geometric transformer module to evaluate their performance. The results are presented in Table 6. Concat, Summation, Average, and Hadamard product denote the element-wise operations of concatenating over the channel, adding, averaging, and multiplying the geometric weight and distance weight, respectively. As can be seen, four operations have relatively minor effects on the final performance. But the Hadamard product obtains the best results. Since the Hadamard product achieves a substantial reduction in the weights of neighbor points with geometrical disparity. Simultaneously, it preserves the weights of points within the same class neighborhood, effectively balancing the influence of various data points.

## 4 Conclusion

In this paper, to better leverage local geometric information and accurately capture long-range semantic relationships within the transformer framework, we propose a novel transformer network named GSTran for point cloud segmentation. GSTran mainly consists of two essential modules: a local geometric transformer and a global semantic transformer. In the local geometric transformer module, we explicitly compute the geometric disparity. This allows us to amplify the affinity with geometrically similar neighbors and simultaneously suppress the association with other neighbors. In the global semantic transformer module, we design a multi-head voting strategy. This strategy computes the semantic similarity for each point over a global spatial range, capturing more accurate semantic information. Experiments with competitive performance on public datasets and further analysis demonstrate the effectiveness of our method.

**Acknowledgements.** This work was supported in part by the National Natural Science Foundation of China under Grants 62132006, 62441203, 62311530101 and 62271237, Natural Science Foundation of Jiangxi Province of China under Grants 20223AEI91002, the PNR project FAIR- Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU and Postgraduate Innovation Special Fund of Jiangxi Province under Grant YC2023-B184.

## A Appendix

Experiments on robustness, outdoor point cloud scenarios, and others are included in the supplementary material. For more details, please refer to the links below. Either of the following two links can be chosen.

**Link 1: Google Drive.**

<https://drive.google.com/file/d/1rS36mBizZS4yHw4tcuOc5JAYDYLR1SUk/view?usp=sharing>

**Link 2: Baidu Drive. Password: 1234**

<https://pan.baidu.com/s/1T3hOOrgMKvwmQOvGTzaeVQ>  
Password: 1234

## References

1. Bazazian, D., Casas, J.R., Ruiz-Hidalgo, J.: Fast and robust edge extraction in unorganized point clouds. In: 2015 International Conference on Digital Image Computing: Techniques and applications, pp. 1–8. IEEE (2015)
2. Chen, J., Kakillioglu, B., Velipasalar, S.: Background-aware 3-D point cloud segmentation with dynamic point feature aggregation. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–12 (2022)
3. Du, Z., Ye, H., Cao, F.: A novel local-global graph convolutional method for point cloud semantic segmentation. *IEEE Trans. Neural Netw. Learn. Syst.* (2022)
4. Fan, S., Dong, Q., Zhu, F., Lv, Y., Ye, P., Wang, F.Y.: SCF-net: learning spatial contextual features for large-scale point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14504–14513 (2021)
5. Feng, M., Zhang, L., Lin, X., Gilani, S.Z., Mian, A.: Point attention network for semantic segmentation of 3D point clouds. *Pattern Recogn.* **107**, 107446 (2020)
6. Guo, B., Deng, L., Wang, R., Guo, W., Ng, A.H.M., Bai, W.: MCTNet: multiscale cross-attention based transformer network for semantic segmentation of large-scale point cloud. *IEEE Trans. Geosci. Remote Sens.* (2023)
7. Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R.R., Hu, S.-M.: PCT: point cloud transformer. *Comput. Vis. Media* **7**(2), 187–199 (2021). <https://doi.org/10.1007/s41095-021-0229-5>
8. Li, Y., Duan, Y.: Multi-scale network with attentional multi-resolution fusion for point cloud semantic segmentation. In: 2022 26th International Conference on Pattern Recognition, pp. 3980–3986. IEEE (2022)
9. Li, Z., et al.: Geodesic self-attention for 3d point clouds. *Adv. Neural. Inf. Process. Syst.* **35**, 6190–6203 (2022)
10. Liu, M., et al.: PartSLIP: low-shot part segmentation for 3d point clouds via pre-trained image-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21736–21746 (2023)
11. Lu, D., Xie, Q., Gao, K., Xu, L., Li, J.: 3DCTN: 3D convolution-transformer network for point cloud classification. *IEEE Trans. Intell. Transp. Syst.* **23**(12), 24854–24865 (2022)



12. Ma, X., Qin, C., You, H., Ran, H., Fu, Y.: Rethinking network design and local geometry in point cloud: a simple residual MLP framework. In: International Conference on Learning Representations (2021)
13. Mei, G., Riz, L., Wang, Y., Poiesi, F.: Geometrically-driven aggregation for zero-shot 3D point cloud understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
14. Mei, G., et al.: Unsupervised point cloud representation learning by clustering and neural rendering. *Int. J. Comput. Vision* 1–19 (2024)
15. Park, J., Lee, S., Kim, S., Xiong, Y., Kim, H.J.: Self-positioning point-based transformer for point cloud understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21814–21823 (2023)
16. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **30** (2017)
17. Qiu, S., Anwar, S., Barnes, N.: Geometric back-projection network for point cloud classification. *IEEE Trans. Multimedia* **24**, 1943–1955 (2021)
18. Qiu, S., Anwar, S., Barnes, N.: Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1757–1767 (2021)
19. Ran, H., Liu, J., Wang, C.: Surface representation for point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18942–18952 (2022)
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
21. Song, H., Feng, H.Y.: A progressive point cloud simplification algorithm with preserved sharp edge data. *Int. J. Adv. Manuf. Technol.* **45**, 583–592 (2009)
22. Srivastava, S., Sharma, G.: Exploiting local geometry for feature and graph construction for better 3d point cloud processing with graph neural networks. In: 2021 IEEE International Conference on robotics and Automation, pp. 12903–12909. IEEE (2021)
23. Tang, L., Zhan, Y., Chen, Z., Yu, B., Tao, D.: Contrastive boundary learning for point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8489–8499 (2022)
24. Wang, C., Ning, X., Sun, L., Zhang, L., Li, W., Bai, X.: Learning discriminative features by covering local geometric space for point cloud analysis. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–15 (2022)
25. Wu, C., Zheng, J., Pfrommer, J., Beyerer, J.: Attention-based point cloud edge sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5333–5343 (2023)
26. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5589–5598 (2020)
27. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: PoinTr: diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12498–12507 (2021)
28. Zhang, H., Wang, C., Yu, L., Tian, S., Ning, X., Rodrigues, J.: PointGT: a method for point-cloud classification and segmentation based on local geometric transformation. *IEEE Trans. Multimed.* (2024)

29. Zhao, H., Jiang, L., Fu, C.W., Jia, J.: PointWeb: enhancing local neighborhood features for point cloud processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5565–5573 (2019)
30. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16259–16268 (2021)
31. Zhou, W., et al.: GTNet: graph transformer network for 3D point cloud classification and semantic segmentation. arXiv preprint [arXiv:2305.15213](https://arxiv.org/abs/2305.15213) (2023)
32. Zhu, H., Yang, H., Wu, X., Huang, D., Zhang, S., et.al.: PonderV2: pave the way for 3D foundataion model with a universal pre-training paradigm. arXiv preprint [arXiv:2310.08586](https://arxiv.org/abs/2310.08586) (2023)



# SPiKE: 3D Human Pose from Point Cloud Sequences

Irene Ballester<sup>(✉)</sup> , Ondřej Peterka , and Martin Kampel 

Computer Vision Lab, TU Wien, Vienna, Austria  
{irene.ballester,martin.kampel}@tuwien.ac.at

**Abstract.** 3D Human Pose Estimation (HPE) is the task of locating keypoints of the human body in 3D space from 2D or 3D representations such as RGB images, depth maps or point clouds. Current HPE methods from depth and point clouds predominantly rely on single-frame estimation and do not exploit temporal information from sequences. This paper presents SPiKE, a novel approach to 3D HPE using point cloud sequences. Unlike existing methods that process frames of a sequence independently, SPiKE leverages temporal context by adopting a Transformer architecture to encode spatio-temporal relationships between points across the sequence. By partitioning the point cloud into local volumes and using spatial feature extraction via point spatial convolution, SPiKE ensures efficient processing by the Transformer while preserving spatial integrity per timestamp. Experiments on the ITOP benchmark for 3D HPE show that SPiKE reaches 89.19% mAP, achieving state-of-the-art performance with significantly lower inference times. Extensive ablations further validate the effectiveness of sequence exploitation and our algorithmic choices. Code and models are available at: <https://github.com/iballester/SPiKE>.

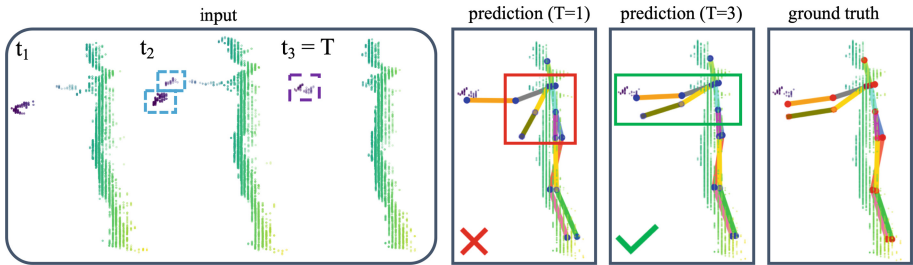
**Keywords:** 3D human pose estimation · point cloud · depth maps

## 1 Introduction

3D Human Pose Estimation (HPE) aims to localize body keypoints or joints in the 3-dimensional space from images, videos and 3D representations, such as point clouds. This task faces significant challenges due to the occlusion of body parts, diversity in human postures, and the wide range of human appearances and shapes. Achieving precise joint localisation is critical for numerous applications in the real world [47], including human activity recognition [31], gait analysis [18, 32] and motion forecasting [9].

Methods for HPE from RGB have received more attention than depth-based approaches [21, 46, 49]. However, the depth modality offers distinct advantages for 3D HPE, encoding inherent 3D information and exhibiting robustness to diverse lighting conditions while preserving privacy better than RGB [2, 25]. Depth maps serve as a representation of 3D space in a 2D image, allowing the

direct extraction of 3D HPE from a single 2D depth map [16,42]. However, using 2D depth maps encounters challenges such as perspective distortion and a non-linear mapping between depth maps and 3D coordinates, which hinder the learning process [24,48]. To overcome this limitation, 3D representations are derived from depth maps, such as voxels [24] or point clouds [48], facilitating the direct estimation of 3D keypoints. In line with these works, we employ point clouds from depth maps as input for our method.



**Fig. 1. Importance of exploiting sequence information.** When considering only the current frame (sequence length  $T=1$ ), only the right hand is visible in the input point cloud, leading to an incorrect prediction. On the contrary, if we consider past frames ( $T=3$ ), in particular  $t_2$  where both hands are visible, SPiKE estimates the position of both arms more accurately. Timestamp ID: 3\_02244.

Different from direct methods, 2D-3D lifting methods [7,44,45], inspired by lifting approaches in RGB [23,46,49], first estimate the 2D keypoints from the depth map. Then, they use the z-coordinate of the estimated 2D keypoints in the depth map to project its coordinates into 3D space and refine the 3D prediction, typically using a point cloud representation. This comes at the cost of increased complexity, as methods must deal with more than one input type. Furthermore, since depth-based methods rely on dense maps to extract the coordinates of the joints, they are not compatible with sparse point clouds [45] and would require additional algorithms to densify the point cloud [5,33]. In this work, we propose a pure point cloud method, that achieves state-of-the-art performance with a reduced inference time.

Most of the presented works process the timestamps of sequences of depth maps or 3D representations independently without using sequence information and thus without taking advantage of temporal relationships between frames. In the RGB domain, sequence information proves beneficial for pose estimation to cope with occlusion [21,40]. Inspired by recent advances in dynamic point cloud processing [12,13,39], we propose to use a Transformer [34] to process sequential point clouds for 3D HPE. As illustrated in Fig. 1, sequence information makes the model more robust against occluded body parts and noise.

More specifically, we divide each point cloud of the sequence into local volumes and extract spatial features within them. The input tokens for the Transformer are generated by combining the 4D coordinates of the centroid of the

local volumes and the features of that volume extracted by a point convolution. This is fed into a Transformer [34] that performs global self-attention to encode the spatio-temporal relationships of the points along the sequence to predict the 3D coordinates of the joints. Our method, SPiKE (Sequential Point clouds for Keypoint Estimation), is validated on the ITOP dataset [19] and outperforms the state of the art, confirming its suitability for 3D HPE from sequential point clouds. Our contributions are as follows:

- We introduce SPiKE, a novel approach for 3D HPE from point cloud sequences. Unlike previous works that process timestamps independently, our method leverages temporal information by employing a Transformer to encode the spatio-temporal structure along the sequence. To ensure efficient processing by the Transformer while preserving spatial integrity per frame, SPiKE partitions each point cloud of the sequence into local volumes for feature extraction through point spatial convolution.
- Experiments, qualitative results and comparisons with the state of the art confirm the effectiveness of our approach. SPiKE achieves an mAP of 89.19% for 3D HPE on the ITOP dataset [19], outperforming existing direct models from depth maps, voxels or point clouds. Furthermore, our model performs similarly to 2D-3D lifting approaches with a significantly lower inference time since no depth-branch is required.
- Extensive ablation studies confirm the value of leveraging sequence information, retaining spatial structure per timestamp, and our algorithmic choices.

## 2 Related Work

### 2.1 Human Pose Estimation from 3D Information

#### **Direct Methods for 3D HPE: Depth Maps, Voxels and Point Clouds.**

The depth modality is different from RGB in that, by its nature, it already contains 3D information in its 2D form. Early works in HPE from depth extract the 3D coordinates directly from the 2D depth map [3, 26, 35, 36, 42]. Arguing a lack of generalisation capabilities between different perspectives, DECA [16] utilizes Capsule Networks [20] to model inherent geometric relations in human skeletons to achieve viewpoint-equivariance. Depth maps offer the advantage of lightweight data storage and processing, and their 2D nature facilitates the adaptation of RGB models and pre-trained feature extractors, broadening the scope of available methods and datasets. However, they suffer from perspective distortion [24, 37]. To overcome this limitation, Moon et al. [24] propose to voxelise the depth map to obtain a volumetric representation and generate per-voxel likelihoods for each keypoint. Despite effectively solving the problem of perspective distortion, voxels also present challenges in terms of computational demands and unavoidable quantisation errors during voxelisation, which is particularly relevant for HPE where precise scene geometry measurement is crucial.

Point clouds require memory relative to the number of points and provide arbitrary precision. In this line, Zhou et al. [48], adapt stacked EdgeConv layers

from DGCNN [38] and T-Net from PointNet [27] to regress the 3D positions of the joints. More recently, Weng et al. [41] propose an unsupervised pre-training strategy for 3D HPE from point clouds. LiDAR-HMR [11] estimates 3D human body mesh from sparse point clouds by first estimating the 3D human pose to then employ a sparse-to-dense 3D mesh reconstruction approach. LPFormer [43] proposes a top-down multitask approach for 3D HPE from sparse point clouds.

Despite the advances in 3D HPE from depth maps and point clouds resulting from the approaches presented, they all process each depth map or point cloud independently and cannot directly process sequences due to the lack of an inter-frame feature fusion approach. Encouraged by the success of integrating sequence information in the RGB domain [1, 21, 40], we propose the use of a Transformer architecture [34] to encode sequence information.

### **2D-3D Lifting Models for 3D HPE: Depth Maps + Point Clouds.**

Extracting an intermediate 2D pose from depth maps and then refining its 3D projection with point clouds proves to be an effective strategy. In this line, inspired by RGB 2D-3D lifting methods, D’Eusano et al. [7] evaluate the modular refinement network RefiNet [6] using as starting point 2D keypoints from HRNet [38]. Following a similar strategy for hand pose estimation, Ren et al. [30] iteratively correct the 3D projection of the estimated 2D hand keypoints by taking a feature set from a local region around each estimated joint. Zhang et al. [44] use depth maps to obtain an intermediate 2D pose estimate and sampled point cloud, and then refine the estimates by processing point clouds through PointNet [27]. An ablation study presented in this work shows that using 2D predictions as a starting point instead of direct 3D estimation from point clouds improves the overall accuracy by almost 14 points on the ITOP dataset, demonstrating that combining these modalities is an effective strategy. Building on this work, Adapose [45] adds to this pipeline 1) an adaptive sampling strategy for point clouds and 2) an LSTM module to capture inter-frame features and enforce temporal smoothness, demonstrating the benefits of using sequential point cloud processing. This last finding, coupled with the evidence from the RGB modality, further strengthens our argument for the use of sequence information.

One of the main contributions is that SPiKE takes only point clouds as input, without requiring depth maps for intermediate 2D estimation. This not only provides versatility by allowing seamless integration with different point cloud acquisition methods, but also significantly reduces inference time as SPiKE performs direct estimation in 3D.

## **2.2 Deep Learning for Dynamic Point Clouds**

Point cloud sequences, unlike grid-based RGB video, lack regularity in spatial arrangement as points appear inconsistently over time. One approach to address this lack of order is to voxelise the 3D space and apply 4D grid-based convolutions. In this line, Choy et al. [4] extend the temporal dimension of 3D sparse convolutions [17] to extract spatio-temporal features on 4D occupancy grids.

3DV [37] combines voxel-based and point-based modelling by first integrating 3D motion information into a regular compact voxel set and then applying PointNet++ [28] to extract representations via temporal rank pooling [15].

An alternative to voxelisation is to operate directly on point sets, avoiding the quantisation errors inherent in the voxelisation process. In this line, MeteorNet [22] extends PointNet++ [28] for 4D point cloud processing to collect information from neighbours and relies on point tracking to merge points across timestamps. PSTNet [14] decomposes spatial and temporal data and proposes a hierarchical point-based convolution. To avoid point tracking, P4Transformer [12] proposes to use a Transformer architecture [8, 34] to perform self-attention over the whole sequence after encoding spatio-temporal local regions by a 4D point convolution. PST-Transformer [13] modifies the Transformer architecture of [12] to preserve the spatio-temporal encoding structure.

These methods are effective in downstream tasks such as semantic segmentation and activity recognition. In this work, we show that point convolutions and attention-based architectures are also suitable for HPE. Similarly to [12, 13], we use a Transformer to relate local volumes from different timestamps. However, different from [12, 13], we propose to use spatial local regions instead of spatio-temporal ones. The rationale behind this is that while temporal merging is suitable for action recognition, allowing for longer sequences that yield better performance, this strategy is not directly applicable to HPE. For HPE, we show that while sequence information is beneficial, longer sequences beyond a certain length do not improve performance (consistent with [40] in RGB). This choice preserves spatial structure by merging only spatial information within the same timestamp before passing local features to the Transformer.

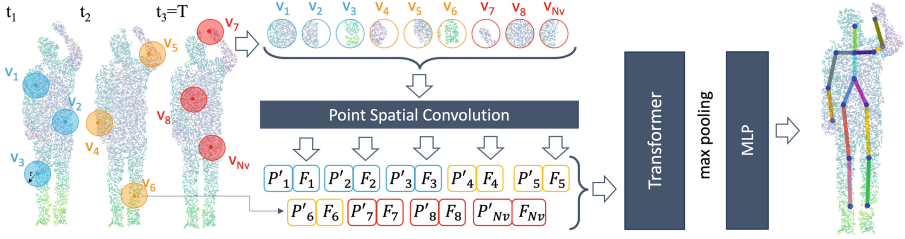
### 3 Method

SPiKE processes a sequence of point clouds,  $[P_1, P_2, \dots, P_T]$ , each containing  $N$  randomly sampled points that are represented as  $P_t = \{p_i(t)\}_{i=1}^N$  for timestamp  $t$  within total sequence length  $T$ . Each point  $p_i(t)$  is defined by its Euclidean coordinates in  $\mathbb{R}^3$ . Our goal is to predict the 3D locations of  $M$  key body joints, represented as  $J = [j_1, j_2, \dots, j_M]$ .

The entire pipeline is illustrated in Fig. 2. First, we extract spatial features by applying a point spatial convolution to local regions in the point cloud of each timestamp (described in Sect. 3.1). These local features, together with a positional encoding, are then processed by a Transformer architecture that merges information across different timestamps (described in Sect. 3.2). Subsequently, a max pooling operation merges the transformed local features into a single global feature representation. Finally, a multilayer perceptron (MLP) regresses the 3D coordinates of the  $M$  joints.

#### 3.1 Point Spatial Convolution in Local Regions

Point cloud sequences represent dynamic 3D environments with a large number of individual points. The direct application of self-attention across all points



**Fig. 2. SPiKE pipeline.** First, each point cloud of the sequence (sequence length =  $T$ ) is partitioned by selecting  $N_v$  reference points  $P'_i$  and creating local volumes  $V_{N_v}$  around them by sampling points within a radius  $r$ . Point Spatial Convolution extracts spatial features  $F_i$  from each local volume. These features are then embedded with the coordinates of their respective reference point  $P'_i$  and fed into the Transformer. After a max pooling layer, an MLP regresses the 3D coordinates of the  $M$  joints.

in such sequences proves to be computationally expensive and demanding in terms of running memory. Following [12, 14, 28], we construct  $N_v$  local regions (hereinafter referred to as local volumes) and perform a point spatial convolution to encode the local structure of the points within these volumes.

Each point cloud is divided into  $N_v$  local volumes  $V_1, V_2, \dots, V_{N_v}$  centred around reference points. The selection of reference points  $P'_1, P'_2, \dots, P'_{N_v}$  is carried out using Farthest Point Sampling (FPS) [28], ensuring that these points are strategically distributed across the point cloud. For each reference point  $P'_i$ , we then sample  $N_s$  neighboring points within a radius  $r$ , again using FPS.

After creating the local volumes, we apply a point spatial convolution to encode the spatial relationships among the  $N_s$  neighboring points contained within. This process transforms the original point cloud sequence, designated as  $[P_1, P_2, \dots, P_T]$ , with  $P_t \in \mathbb{R}^{3 \times N}$  representing the set of point coordinates at the  $t$ -th frame into a sequence of encoded features. Each timestamp in the transformed sequence is represented as  $[P'_1; F_1], [P'_2; F_2], \dots, [P'_T; F_T]$ , where  $P'_t \in \mathbb{R}^{3 \times N_v}$  and  $F_t \in \mathbb{R}^{C \times N_v}$ , with  $C$  denoting the number of feature channels. For any given reference point  $P'_i$ , located at  $(x, y, z, t)$ , its feature vector  $F(x, y, z, t) \in \mathbb{R}^{C \times 1}$  is derived from the spatial convolution as follows:

$$F(x, y, z, t) = \max_{\|(\delta_x, \delta_y, \delta_z)\| \leq r} (\text{MLP}(W_s \cdot (\delta_x, \delta_y, \delta_z)^T)) \tag{1}$$

Here,  $W_s \in \mathbb{R}^{C' \times 3}$  represents the transformation matrix applied to the 3D displacements  $(\delta_x, \delta_y, \delta_z)$ , encapsulating the spatial differences relative to the reference point. This matrix multiplication facilitates the projection of spatial displacements into a higher-dimensional feature space, which is subsequently processed by an MLP to enhance the representation. Finally, we aggregate the features by performing max pooling within the local region.



### 3.2 Transformer

**Positional Embedding.** After the point spatial convolution, the local volumes of the  $t$ -th frame are encoded to features  $F(x, y, z, t)$ . These features, however, solely capture the local spatial features without explicitly accounting for the absolute positions of the reference points within the global structure of the point cloud. To address this limitation and ensure the preservation of the spatio-temporal structure inherent to the point cloud sequence, we combine the coordinates of the reference point, i.e.,  $P'(x, y, z, t)$ , and local area features as input to the Transformer.

$$I(x, y, z, t) = W_i \cdot P'(x, y, z, t)^T + F(x, y, z, t) \quad (2)$$

In this equation,  $W_i \in \mathbb{R}^{C \times 4}$  represents a weight matrix that transforms the four-dimensional coordinates  $P'(x, y, z, t)$  into a feature space that is compatible with the encoded local features  $F(x, y, z, t)$ . The result of this transformation,  $I(x, y, z, t)$ , serves as the input to the Transformer, where  $I \in \mathbb{R}^{C \times TN_v}$  are the transformed input features ready for further processing. By embedding the spatial and temporal coordinates directly into the feature representation, the subsequent Transformer layer can take advantage of both the local feature information and the positional context of each reference point.

**Multi-head Self-attention.** The multi-head self-attention mechanism [34] enables the model to capture spatial and temporal dependencies within the sequences of point clouds. Input features  $I(x, y, z, t) \in \mathbb{R}^{C \times TN_v}$  representing local spatial features and positional embeddings are transformed into query ( $Q$ ), key ( $K$ ), and value ( $V$ ) matrices through linear transformations. Specifically, for each local volume centered at  $P'(x, y, z)$  at timestamp  $t$ , we compute:

$$Q = I(x, y, z, t) \cdot W_Q, \quad K = I(x, y, z, t) \cdot W_K, \quad V = I(x, y, z, t) \cdot W_V,$$

where  $W_Q \in \mathbb{R}^{C_k \times C}$ ,  $W_K \in \mathbb{R}^{C_k \times C}$ ,  $W_V \in \mathbb{R}^{C_v \times C}$  are learnable weight matrices, and  $C_k$  and  $C_v$  are the dimensions of key and value, respectively.

The attention mechanism computes attention scores based on the similarity between queries and keys, determining the relevance of different spatial positions and timestamps within the input sequence. For each local volume centered at  $P'(x, y, z)$  at timestamp  $t$ , the attention scores are calculated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V \quad (3)$$

where  $d_k$  represents the dimensionality of the key vectors. The softmax function normalizes the attention scores across the key vectors, indicating the importance of each value vector relative to the given query.

To capture diverse patterns and dependencies within the data, the multi-head mechanism splits the query, key, and value matrices into  $h$  separate heads, each operating independently. This parallel processing enables the model to attend

to different parts of the input simultaneously, enhancing its ability to capture both local and global dependencies. The outputs of the individual heads are then concatenated and linearly transformed by  $W_o$ , resulting in the final output of the multi-head self-attention mechanism. Finally, the model incorporates  $m$  Transformer blocks, each equipped with a multi-head self-attention mechanism.

### 3.3 Implementation Details

SPiKE is trained end-to-end with L1 loss and SGD optimizer (batch size = 24, learning rate = 0.01) for 150 epochs. We use 4096 randomly sampled points, with  $r = 0.2$ ,  $N_v = 128$ , and  $N_s = 32$ . Furthermore,  $C = 1024$ , and the Transformer has  $m = 5$  self-attention blocks with  $h = 8$  heads each. Point clouds are centred by subtracting the mean of 3D coordinates of the points per sequence, with rotation in the y-axis of  $[-90, 90]$  degrees and x-axis mirroring for augmentation. Training and testing are performed on a single NVIDIA GeForce RTX 3090.

To isolate the points belonging to the human, following [48], we use depth thresholding to remove the background and discard the first 10 bins of the y-coordinate histogram to exclude floor points. Then, clusters are formed using DBSCAN [10] with a 15 cm inter-cluster distance. Since humans may not always form a single cluster, we select the largest cluster and include clusters below, above and between the largest cluster and the sensor, offset by 20 cm.

## 4 Evaluation

We describe the dataset and evaluation metrics, and systematically evaluate SPiKE by comparing it to the state of the art, discussing qualitative results, and providing ablations to illustrate our contributions.

### 4.1 Datasets and Metrics

**ITOP Human Pose Dataset** [19] is a collection of 100k depth maps from two camera viewpoints captured with Asus Xtion Pro sensors. It consists of 15 action sequences performed by 20 subjects. All depth maps are labelled with 3D coordinates of 15 body joints from the camera viewpoint. We train and test SPiKE on ITOP front-view and adopt the original division proposed in [19], i.e., using subjects 00-04 for testing and subjects 05-19 for training, so that our evaluation reflects a scenario where testing is performed on unseen subjects.

In ITOP, only about 45% of the annotated joints are human-validated (referred to as “valid joints”), and methods typically evaluate performance only on these valid joints [44, 45, 48], hence, we train and test our method only on validated ground truth annotations. The point clouds of instances with invalid joints are incorporated into the sequence to predict subsequent valid joint positions, but the invalid joints are never used for training or testing.

**Mean Average Precision (mAP).** Following previous work [7, 16, 24, 42], we use mean Average Precision (mAP) as the evaluation metric with a threshold of 10 cm. mAP is the percentage of all predicted joints that fall within an interval of less than 0.10 metres of the 3D coordinates of the ground truth joints.

## 4.2 Comparison with State-of-the-Art Methods

We evaluate SPiKE on ITOP front-view against state-of-the-art methods for 3D HPE on depth maps, point clouds and voxels in Table 1. For better comparison, the different approaches are classified as direct and 2D-3D lifting methods (SPiKE belongs to the former). Specifically, we compare our approach with the following direct methods: V2V [24], A2J [42], Zhou et al. [48] and DECA [16]. For 2D-3D lifting methods, we consider WSM [44], AdaPose [45], and HRNet+RefiNet [7]. For reference, we also include the ablation study by [44] as “WSMa” as part of the direct methods, since in this ablation the 3D pose is estimated from the point clouds without relying on an intermediate extraction of 2D keypoints.

**Table 1.** Comparison with the state-of-the-art methods on ITOP front-view (0.1m mAP). (\*) identifies the methods using additional training data.

Method	direct methods						2D-3D lifting methods		
	V2V 2018	A2J 2019	WSMa 2020	Zhou et al. 2020	DECA 2021	SPiKE (Ours) -	WSM* 2020	AdaPose 2021	HRNet+ RefiNet 2023
Modality	voxels	depth	points	points	depth	points	depth+points		
Head	98.29	98.54	-	96.73	93.87	98.42	98.15	98.42	-
Neck	99.07	99.20	-	98.05	97.90	99.47	99.47	98.67	-
Shoulders	97.18	96.23	-	94.38	95.22	97.48	94.69	95.39	-
Elbows	80.42	78.92	-	73.67	84.53	81.64	82.80	90.74	-
Hands	67.26	68.35	-	54.95	56.49	71.71	69.10	82.15	-
Torso	98.73	98.52	-	98.35	99.04	99.24	99.67	99.71	-
Hips	93.23	90.85	-	91.77	97.42	93.68	95.71	96.43	-
Knee	91.80	90.75	-	90.74	94.56	91.56	91.00	94.41	-
Feet	87.60	86.91	-	86.30	92.04	84.30	89.96	92.84	-
Upper B.	-	-	-	80.10	83.03	88.75	-	-	80.8
Lower B.	-	-	-	89.60	95.30	89.85	-	-	88.1
Mean	88.74	88.00	75.64	85.11	88.75	<b>89.19</b>	89.59	<b>93.38</b>	84.2

SPiKE ( $T = 3$  and only past timestamps) achieves an overall mAP of 89.19%, outperforming existing direct methods using any of the modalities: depth maps,

points and voxels. Our method shows significant improvements, most notably for the upper limbs, which are prone to occlusion when the person is sideways or moving their arms. In these cases of occlusion, the sequence information becomes valuable as certain timestamps can reveal visible joints that are occluded at that particular timestamp, providing crucial context for accurate estimation.

Compared to direct methods working with point clouds, lifting methods can leverage 2D pre-trained backbones with additional data (marked with \* in Table 1). This is an effective strategy to improve performance, but prevents a direct and fair comparison with methods trained only in ITOP. Nevertheless, despite using additional data in the WSM 2D HPE network training, SPiKE (trained only on ITOP) achieves comparable performance with a difference of only 0.4 points. The reliance of 2D-3D lifting approaches on the estimation of an intermediate pose from the depth map is evident from the performance drop of 14 points between WSM (mAP = 89.59%) and WSMa (mAP = 75.64%) when no intermediate pose is considered. In standard WSM, intermediate 2D keypoints are first extracted, reprojected in 3D to obtain an intermediate 3D pose estimate, and then refined by processing the point clouds. In contrast, in the WSMa ablation study, the 3D pose is estimated directly from the point clouds (as in SPiKE, mAP = 89.19%). This illustrates the heavy reliance on an intermediate 2D pose in WSM, while SPiKE can accurately regress the 3D pose directly from point cloud sequences alone.

Direct point cloud methods have the advantage of being independent of depth maps, allowing them to handle data from different acquisition methods, such as LiDAR sensors, which produce sparse point clouds. In contrast, 2D-3D lifting methods depend on depth maps (or dense point clouds) to derive an intermediate pose, making them incompatible with sparse point clouds [45].

### 4.3 Computational Efficiency

The independence from intermediate 2D keypoint extraction eliminates the need for additional processing of the depth maps, reducing the network complexity and computational needs. Table 2 shows a comparison with 2D-3D lifting methods in terms of inference time (ms) and performance on ITOP (mAP). Since Adapose [45] omits its 2D HPE network in their released code, for our comparison, we add the runtime associated with the released code plus the runtime for HRNet, as a representative network for 2D HPE.

Table 2 shows that SPiKE has a comparable runtime to the 2D-3D lifting modules, but it holds a considerable advantage as it operates independently from a 2D HPE network. This independence provides a significant computational advantage, regardless of the efficiency of the 2D-3D lifting approach employed.

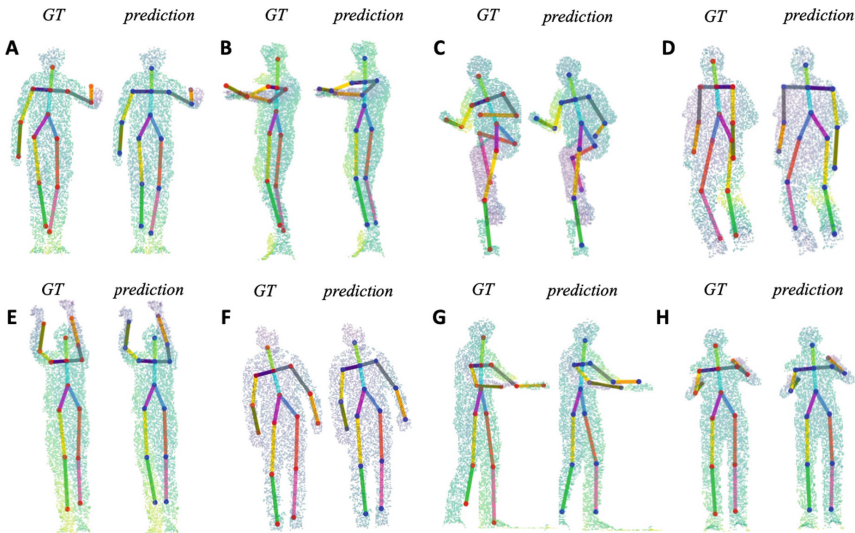
### 4.4 Qualitative Results

Fig. 3 shows a comparison between ground truth joint coordinates (left, with keypoints in red) and the model’s predicted joint positions (right, with keypoints in blue) across a spectrum of poses. This side-by-side view illustrates the model’s

**Table 2.** Inference time (ms) per frame and performance (mAP) on ITOP.

Methods	2D HPE (HRNet)	2D-3D Lifting	Total	mAP (ITOP)
HRNet+Refinet	30.34 ms	5.18 ms	35.52 ms	84.2
AdaPose	30.34 ms	13.23 ms	43.57 ms	93.38
SPiKE (ours)	-	5.98 ms	5.98 ms	89.19

accuracy in predicting body keypoints. This accurate performance is evident not only in standard poses, such as sample A but also in complex situations where limbs are in motion or partially occluded, as shown in samples B, C, E, G and H. The model is also adept at recognising poses in which the person is turned away from the camera, as shown in sample D, where the colours of the limb joints are inverted from left to right and vice versa, indicating body orientation.



**Fig. 3. Qualitative results.** Each pair represents the groundtruth skeletons on the left (keypoints in red) and the joints predicted by the model on the right (keypoints in blue). ID top row: A: 0\_01439, B: 2\_00220, C: 1\_00587, D: 3\_02966. ID bottom row: E: 0\_01712, F: 2\_02827, G: 0\_00168, H: 1\_01611.

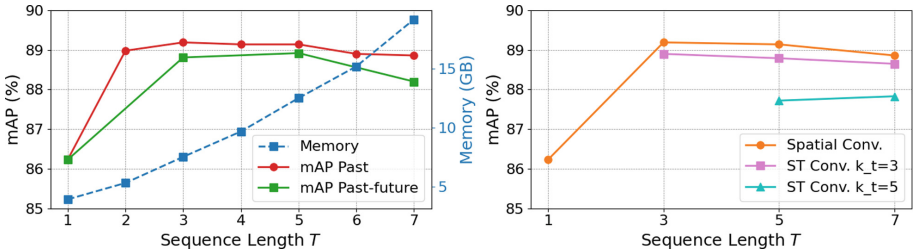
## 4.5 Ablations

We isolate our contributions and algorithmic choices and construct a set of experiments to measure their effect. Specifically, we examine the following aspects of our algorithm: the length of the point cloud sequence, using past or including

also future timestamps for pose estimation, and the effect of spatio-temporal convolution instead of spatial convolution to encode local volume features.

**Past vs. Past-Future Timestamps.** When using only past timestamps, the model relies solely on historical data, potentially missing context from future frames, especially in occlusion scenarios. In theory, including future timestamps allows the model to use both past and future information, providing a more complete understanding of the temporal context.

However, Fig. 4 (Left) shows that the performance difference between these approaches is marginal, regardless of sequence length. Thus, while considering future timestamps may theoretically enrich the temporal context, the practical advantage appears to be limited, and models can rely predominantly on past timestamps for efficiency without sacrificing significant performance gains, opening up our method for real-world applications.



**Fig. 4. Ablations** Left: Effect on performance (mAP) and running memory (GB) vs. sequence length  $T$ , using only past or past and future timestamps. Spatial convolutions are employed for this ablation study. Right: Effect on performance (mAP) vs. sequence length  $T$  for spatial convolutions, spatio-temporal (ST) convolutions with temporal kernel size  $k_t = 3$  and  $k_t = 5$ . For this ablation, only past timestamps are considered.

**Sequence Length.** In action recognition, longer point cloud sequences consistently yield superior results [12–14, 22, 39]. This superiority arises from the uneven distribution of action-related information over time. Consequently, short sequences may overlook critical frames necessary for accurate action inference.

However, the relationship between sequence length and HPE is not as direct, as the influence of distant timestamps on the current pose may be minimal. Moreover, processing long sequences in HPE requires more memory without necessarily adding significant new information. This phenomenon is illustrated in the left plot of Fig. 4 where the mAP peaks at a certain sequence length ( $T = 3$ ), beyond which the memory requirement continues to increase without a significant improvement in performance. This finding aligns with previous work in HPE from egocentric RGB videos [40].

**Spatial vs. Spatio-Temporal Convolution.** We compare the effectiveness of spatial against spatio-temporal convolutions [12, 13] for feature encoding from local volumes as input to the Transformer. Spatial convolutions maintain the spatial structure within each timestamp, while spatio-temporal convolutions merge information across timestamps, allowing for processing longer sequences.

Fig. 4 (Right) shows the performance (mAP) using spatial convolutions and spatio-temporal convolutions with temporal kernel sizes  $k_t = 3$  and  $k_t = 5$ . Our findings confirm that spatial convolutions are more effective than spatio-temporal convolutions for HPE due to the fine-grained nature of the task.

## 5 Limitations and Future Work

Despite the multiple contributions of our work, SPiKE is not without limitations. First, similar to [44, 45, 48], we apply depth thresholding and clustering to isolate the points belonging to the human. While this strategy is effective for the ITOP dataset, it may not be sufficient for real-world applications. In addition, SPiKE currently focuses on single-human pose estimation and does not address multi-human scenarios. Therefore, future work is needed to address the effective integration of human instance detection as part of the HPE framework. A second line of future work arises from the versatility of SPiKE, which requires only point cloud sequences as input, allowing HPE from point clouds acquired by different sensing devices. Our evaluation is limited to point clouds derived from depth maps, and future work will investigate its performance on datasets of sparse LiDAR point clouds. Finally, future research directions include the adaptation of auto-regressive motion models, such as HuMoR [29], for 3D HPE from point cloud sequences.

## 6 Conclusion

We presented SPiKE, a novel approach to 3D HPE from point cloud sequences employing point spatial convolutions and a Transformer architecture to encode spatio-temporal relationships between points along the sequence. We demonstrated that exploiting temporal information by processing sequential point clouds yields superior results compared to treating each timestamp independently.

To ensure efficient processing while preserving per-timestamp spatial integrity, SPiKE partitions each point cloud of the sequence into local volumes and extracts spatial features through point spatial convolution. Ablation studies confirmed the effectiveness of this strategy and highlighted the superiority of spatial convolutions over spatio-temporal convolutions for HPE.

Experiments on ITOP validated SPiKE’s effectiveness, outperforming existing direct approaches with an 89.19% mAP. Using only point clouds, SPiKE performed comparably to lifting approaches with significantly faster inference.

Qualitative analysis further underscored SPiKE accuracy across a wide range of poses, including complex scenarios involving occlusion or varying orientations.

These findings collectively illustrate the robustness of SPiKE in accurately estimating 3D human poses from point cloud sequences.

**Acknowledgements.** This work is supported by the Vienna Science and Technology Fund (AlgoCare - grant agreement No. ICT20-055) and the European Union’s H2020 (VisuAAL - grant agreement No. 861091). The publication reflects the views only of the authors, and the European Union cannot be held responsible for any use which may be made of the information contained therein.

## References

1. Arnab, A., Doersch, C., Zisserman, A.: Exploiting temporal context for 3D human pose estimation in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3395–3404 (2019)
2. Ballester, I., Kampel, M.: Action Recognition from 4D Point Clouds for Privacy-Sensitive Scenarios in Assistive Contexts. In: International Conference on Computers Helping People with Special Needs. pp. 359–364. Springer (2024)
3. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human Pose Estimation with Iterative Error Feedback. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4733–4742 (2016)
4. Choy, C., Gwak, J., Savarese, S.: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)
5. Cui, Y., Chen, R., Chu, W., Chen, L., Tian, D., Li, Y., Cao, D.: Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review. *IEEE Trans. Intell. Transp. Syst.* **23**(2), 722–739 (2021)
6. D’Eusanio, A., Pini, S., Borghi, G., Vezzani, R., Cucchiara, R.: RefiNet: 3D Human Pose Refinement with Depth Maps. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 2320–2327. IEEE (2021)
7. D’Eusanio, A., Simoni, A., Pini, S., Borghi, G., Vezzani, R., Cucchiara, R.: Depth-based 3D human pose refinement: Evaluating the RefiNet framework. *Pattern Recogn. Lett.* **171**, 185–191 (2023)
8. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT. pp. 4171–4186. Association for Computational Linguistics (2019)
9. Diller, C., Funkhouser, T., Dai, A.: Forecasting Characteristic 3D Poses of Human Actions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15914–15923 (2022)
10. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Knowledge Discovery and Data Mining. p. 226–231. AAAI Press (1996)
11. Fan, B., Zheng, W., Feng, J., Zhou, J.: LiDAR-HMR: 3D Human Mesh Recovery from LiDAR. arXiv preprint [arXiv:2311.11971](https://arxiv.org/abs/2311.11971) (2023)
12. Fan, H., Yang, Y., Kankanhalli, M.: Point 4D Transformer Networks for Spatio-Temporal Modeling in Point Cloud Videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14204–14213 (2021)
13. Fan, H., Yang, Y., Kankanhalli, M.: Point Spatio-Temporal Transformer Networks for Point Cloud Video Modeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(2), 2181–2192 (2023)



14. Fan, H., Yu, X., Ding, Y., Yang, Y., Kankanhalli, M.: PSTNet: Point Spatio-Temporal Convolution on Point Cloud Sequences. In: International Conference on Learning Representations (2021)
15. Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., Tuytelaars, T.: Rank pooling for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 773–787 (2016)
16. Garau, N., Bisagno, N., Bródka, P., Conci, N.: DECA: Deep viewpoint-Equivariant human pose estimation using Capsule Autoencoders. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11677–11686 (2021)
17. Graham, B., Van der Maaten, L.: Submanifold Sparse Convolutional Networks. arXiv preprint [arXiv:1706.01307](https://arxiv.org/abs/1706.01307) (2017)
18. Gu, X., Guo, Y., Yang, G.Z., Lo, B.: Cross-Domain Self-Supervised Complete Geometric Representation Learning for Real-Scanned Point Cloud Based Pathological Gait Analysis. *IEEE J. Biomed. Health Inform.* **26**(3), 1034–1044 (2021)
19. Haque, A., Peng, B., Luo, Z., Alahi, A., Yeung, S., Fei-Fei, L.: Towards Viewpoint Invariant 3D Human Pose Estimation. arXiv preprint [arXiv:1603.07076](https://arxiv.org/abs/1603.07076) (2016)
20. Hinton, G.E., Sabour, S., Frosst, N.: Matrix capsules with EM routing. In: International Conference on Learning Representations (2018)
21. Jeong, D.C., Liu, H., Salazar, S., Jiang, J., Kitts, C.A.: SoloPose: One-Shot Kinematic 3D Human Pose Estimation with Video Data Augmentation. arXiv preprint [arXiv:2312.10195](https://arxiv.org/abs/2312.10195) (2023)
22. Liu, X., Yan, M., Bohg, J.: Meteornet: Deep Learning on Dynamic 3D Point Cloud Sequences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9246–9255 (2019)
23. Mehraban, S., Adeli, V., Taati, B.: MotionAGFormer: Enhancing 3D Human Pose Estimation with a Transformer-GCNFormer Network. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6920–6930 (2024)
24. Moon, G., Chang, J.Y., Lee, K.M.: V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5079–5088 (2018)
25. Mucha, W., Kampel, M.: Addressing privacy concerns in depth sensors. In: International Conference on Computers Helping People with Special Needs. pp. 526–533. Springer (2022)
26. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7025–7034 (2017)
27. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
28. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems* **30** (2017)
29. Rempe, D., Birdal, T., Hertzmann, A., Yang, J., Sridhar, S., Guibas, L.J.: HuMoR: 3D Human Motion Model for Robust Pose Estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11488–11499 (2021)
30. Ren, P., Chen, Y., Hao, J., Sun, H., Qi, Q., Wang, J., Liao, J.: Two Heads Are Better than One: Image-Point Cloud Network for Depth-Based 3D Hand Pose

- Estimation. Proceedings of the AAAI Conference on Artificial Intelligence **37**(2), 2163–2171 (2023)
31. Sun, Z., Ke, Q., Rahmani, H., Bennamoun, M., Wang, G., Liu, J.: Human Action Recognition from Various Data Modalities: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(3), 3200–3225 (2022)
  32. Teepe, T., Gilg, J., Herzog, F., Hörmann, S., Rigoll, G.: Towards a Deeper Understanding of Skeleton-based Gait Recognition. [arXiv:2204.07855](https://arxiv.org/abs/2204.07855) (2022)
  33. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity Invariant CNNs. In: 2017 International Conference on 3D Vision. pp. 11–20 (2017)
  34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is All you Need. *Advances in Neural Information Processing systems* **30** (2017)
  35. Wang, K., Lin, L., Ren, C., Zhang, W., Sun, W.: Convolutional Memory Blocks for Depth Data Representation Learning. In: *IJCAI*. pp. 2790–2797 (2018)
  36. Wang, K., Zhai, S., Cheng, H., Liang, X., Lin, L.: Human Pose Estimation from Depth Images via Inference Embedded Multi-task Learning. In: *Proceedings of the ACM International Conference on Multimedia*. pp. 1227–1236 (2016)
  37. Wang, Y., Xiao, Y., Xiong, F., Jiang, W., Cao, Z., Zhou, J.T., Yuan, J.: 3D Dynamic Voxel for Action Recognition in Depth Video. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 511–520 (2020)
  38. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics* **38**(5), 1–12 (2019)
  39. Wen, H., Liu, Y., Huang, J., Duan, B., Yi, L.: Point Primitive Transformer for Long-Term 4D Point Cloud Video Understanding. In: *European Conference on Computer Vision*. pp. 19–35. Springer (2022)
  40. Wen, Y., Pan, H., Yang, L., Pan, J., Komura, T., Wang, W.: Hierarchical Temporal Transformer for 3D Hand Pose Estimation and Action Recognition from Egocentric RGB Videos. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 21243–21253 (2023)
  41. Weng, Z., Gorban, A.S., Ji, J., Najibi, M., Zhou, Y., Anguelov, D.: 3D Human Keypoints Estimation From Point Clouds in the Wild Without Human Labels. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 1158–1167 (2023)
  42. Xiong, F., Zhang, B., Xiao, Y., Cao, Z., Yu, T., Zhou, J.T., Yuan, J.: A2J: Anchor-to-Joint Regression Network for 3D Articulated Pose Estimation from a Single Depth Image. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 793–802 (2019)
  43. Ye, D., Xie, Y., Chen, W., Zhou, Z., Ge, L., Foroosh, H.: LPFormer: LiDAR pose estimation transformer with multi-task network. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 16432–16438. IEEE (2024)
  44. Zhang, Z., Hu, L., Deng, X., Xia, S.: Weakly Supervised Adversarial Learning for 3D Human Pose Estimation from Point Clouds. *IEEE Trans. Visual Comput. Graphics* **26**(5), 1851–1859 (2020)
  45. Zhang, Z., Hu, L., Deng, X., Xia, S.: Sequential 3D Human Pose Estimation Using Adaptive Point Cloud Sampling Strategy. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. pp. 1330–1337 (2021)
  46. Zhao, Q., Zheng, C., Liu, M., Wang, P., Chen, C.: PoseFormerV2: Exploring Frequency Domain for Efficient and Robust 3D Human Pose Estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8877–8886 (2023)

47. Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., Kehtarnavaz, N., Shah, M.: Deep Learning-Based Human Pose Estimation: A Survey. *ACM Comput. Surv.* **56**(1), 1–37 (2023)
48. Zhou, Y., Dong, H., El Saddik, A.: Learning to Estimate 3D Human Pose From Point Cloud. *IEEE Sensors Journal* pp. 1–1 (2020)
49. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: MotionBERT: A Unified Perspective on Learning Human Motion Representations. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 15085–15099 (2023)

# Author Index

## A

Alagrami, Ali 197  
Aslan, Sinem 197

## B

Baldwin-McDonald, Thomas 436  
Ballan, Lamberto 292  
Ballester, Irene 470  
Baraldi, Lorenzo 209  
Bhandarkar, Suchendra M. 161  
Bing, Qi 259  
Bozma, H. Işıl 31

## C

Cai, Weidong 259  
Chakraborty, Devjyoti 161  
Chattopadhyay, Pratik 62  
Chen, Mingkai 226  
Chen, Zibo 1  
Cornia, Marcella 209  
Cucchiara, Rita 209

## F

Fang, Yi 389  
Fang, Yuming 453  
Feng, Jiawei 1

## G

Ghosh, Kriti 161  
Gong, Minglun 130  
Guo, Yanqing 275  
Guo, Yu 242

## H

Han, Xin 325  
Hao, Yu 389  
Hu, Xiaoxu 242  
Huang, Hao 389  
Huang, Xinjian 242

Huang, Yanjia 389  
Huang, Yisiyuan 389  
Huang, Zhentao 130

## I

Izzo, Elena 292

## J

Jad, Sarah 16  
Jaswanth, Bandam Sai 62  
Jiang, YiHeng 180  
Jin, Zijian 226  
Jutla, Charanjit 373

## K

Kampel, Martin 470  
Kaushal, Rohan 62  
Kaushik, Arjun Ramesh 373  
Kawakami, Rei 80  
Khalafallah, Ayman 16  
Kim, In Kee 161  
Kojima, Mizuki 80

## L

Laaksonen, Jorma 308  
Lai, Shang-Hong 405  
Li, Abiao 453  
Li, Hui 325  
Li, Yang 180  
Li, Yi 275  
Lin, Hui 389  
Liu, ChunYan 180  
Liu, Guisheng 275  
Liu, Han 325  
Liu, Yating 275  
Liu, Yuanqiu 325  
Liu, Yu-Shen 389  
Liu, Zhixuan 1  
Lourakis, Manolis 97

Luo, Jiebo 226  
 Lv, Chenlei 453  
 Lyu, Hanjia 226

**M**

Mei, Guofeng 453  
 Miao, Yubin 420  
 Mishra, Deepak R. 161  
 Moratelli, Nicholas 209

**N**

Nakamura, Tsubasa 113  
 Nakano, Gaku 113

**O**

Okutomi, Masatoshi 80

**P**

Palmieri, Luca 197  
 Park, In Kyu 145  
 Parolari, Luca 292  
 Pathak, Stuti 436  
 Pehlivan, Selen 308  
 Pelillo, Marcello 197  
 Penne, Rudi 436  
 Peterka, Ondřej 470

**Q**

Qin, Xugong 242

**R**

Ramaswamy, Lakshmish 161  
 Ratha, Nalini 373  
 Reddy, Sana Vishnu Karthikeya 62

**S**

Sakurada, Ken 113  
 Sandhan, Tushar 46  
 Sels, Sepp 436  
 Shi, Yukun 130  
 Singh, Binit 62  
 Singh, Divij 62  
 Sukma, Zaki 161

**T**

Tang, Hao 358  
 Tarakeswara Rao, Landa 46  
 Terzakis, George 97  
 Tiwari, Hans 308  
 Torki, Marwan 16  
 Tsai, Cheng-Yu 405

**V**

Vascon, Sebastiano 197

**W**

Wan, Junkang 420  
 Wang, ZhiPeng 180  
 Wei, Xiangpeng 275  
 Wen, Congcong 389  
 Wu, Hang 420  
 Wu, Xinyu 242

**X**

Xie, Shangjin 1

**Y**

Yan, Yan 358  
 Yang, Junming 226  
 Yao, Yuan 226  
 Yoon, Soyoung 145  
 Yu, Hong 325  
 Yuan, Shuaihang 389

**Z**

Zeng, Gangyan 242  
 Zeng, Jianshu 341  
 Zhang, Chaoyi 259  
 Zhang, Chi 341  
 Zhang, Daoan 226  
 Zhang, Jian 453  
 Zhang, Peng 242  
 Zhao, Runbo 242  
 Zhao, YunLong 180  
 Zhao, Zhenghao 358  
 Zheng, Wei-Shi 1  
 Zuo, Yifan 453