

Advances in Computer Vision and Pattern Recognition



Kaiyang Zhou
Ziwei Liu
Peng Gao *Editors*

Large Vision-Language Models

Pre-training, Prompting, and
Applications



Springer

Advances in Computer Vision and Pattern Recognition

Founding Editor

Sameer Singh

Series Editor

Srinivasa Narasimhan, Carnegie Mellon University, Pittsburgh, PA, USA

Advisory Editors

Richard Bowden, University of Surrey, Guildford, UK

Sven Dickinson, University of Toronto, Toronto, ON, Canada

Jiaya Jia, The Chinese University of Hong Kong, Shatin, Hong Kong

Zhouchen Lin , Peking University, Beijing, China

Bernt Schiele, Max Planck Institute for Informatics, Saarbrücken, Germany

The field of computer vision and pattern recognition has a rich history of nearly 50 years. In the past decade, however, the field has experienced remarkable advances in scene understanding and image generation. This advancement is driven by three key factors: (a) the availability of large, diverse datasets, (b) the accessibility of cloud and personal computing, and (c) the open release of advanced neural network architectures and models. These breakthroughs have led to significant successes across numerous application domains, including intelligent transportation, augmented reality, healthcare, agriculture, oceanography, and more. The ACVPR book series aims to introduce, analyze, and synthesize recent and foundational research, offering valuable references to both beginners and expert practitioners. The series covers timely topics such as:

- Deep Learning for Vision
- Large-scale Foundational Models
- Generative Methods
- Multimodal Learning (vision, audio, language, action, etc.)
- Neural Fields for Vision
- 3D Computer Vision
- Computational Photography, Display and Illumination
- Video Understanding and Synthesis
- Virtual, Mixed and Augmented Reality
- Biological and Human Vision
- Physics-based Vision
- Vision for Graphics
- Ethics in Computer Vision
- Applications (Robotics, Agriculture, Health, Intelligent Transportation, Oceanography, Safety/Security, etc.)

This series includes monographs, introductory and advanced textbooks, and state-of-the-art collections. Furthermore, it supports Open Access publication mode.

Kaiyang Zhou • Ziwei Liu • Peng Gao
Editors

Large Vision-Language Models

Pre-training, Prompting, and Applications



Editors

Kaiyang Zhou
Hong Kong Baptist University
Kowloon, Hong Kong

Ziwei Liu
Nanyang Technological University
Singapore, Singapore

Peng Gao
Shanghai Artificial Intelligence Laboratory
Shanghai, China

ISSN 2191-6586

ISSN 2191-6594 (electronic)

Advances in Computer Vision and Pattern Recognition

ISBN 978-3-031-94968-5

ISBN 978-3-031-94969-2 (eBook)

<https://doi.org/10.1007/978-3-031-94969-2>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2026

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.

Preface

The pursuit of multimodal understanding has emerged as one of the most transformative frontiers in artificial intelligence. The ability to create models that can interpret, reason about, and generate both multimodal data is not only a technical milestone but also an intellectual leap in the way we approach machine cognition. Vision-Language Models (VLMs) have brought about a paradigm shift, bridging the previously distinct domains of computer vision and natural language processing to open new avenues for intelligent systems.

This book, *Foundations of Vision-Language Models: Concepts and Roadmap*, aims to provide a comprehensive overview of the theoretical underpinnings, progress, and challenges in this rapidly evolving field. We explore the foundational concepts that power VLMs, shedding light on the unique architectures, the role of large-scale pre-training, and the essential multimodal representations that form the backbone of these models for both understanding and generation. Through this exploration, we aim to present a clear understanding of how VLMs operate, their capabilities, and the nuances of aligning visual and textual data in ways that allow for complex reasoning and generation.

The applications of VLMs are vast and growing—from enhancing image recognition systems to enabling sophisticated visual content generation, and even creating systems that can interact with humans in more natural and intuitive ways. However, alongside these opportunities come significant challenges. As the field progresses, issues such as feature alignment, scalability, data requirements, and evaluation metrics require ongoing attention and innovation. Furthermore, concerns regarding computational inefficiency and ethical implications also demand careful consideration as we look toward the future of these technologies.

In this book, we offer a roadmap for both newcomers and experts interested in understanding the current landscape of VLMs, their limitations, and the exciting directions they may take. We aim to provide not only a technical guide but also a reflection on the broader impact of this technology, laying the groundwork for the next wave of advancements in AI.

We hope this book serves as a valuable resource for researchers, practitioners, and students who wish to engage with the foundational concepts and future

directions of VLMs. It is our belief that the continued development of this field will play a critical role in shaping the future of artificial intelligence, enabling machines to interact with, understand, and generate the world in ways that are richer, more accurate, and more human-like than ever before.

Kowloon, Hong Kong
Singapore, Singapore
Shanghai, China
December 2024

Kaiyang Zhou
Ziwei Liu
Peng Gao

Acknowledgements The completion of this book has been a truly collaborative effort. First and foremost, we thank all the contributors whose expertise, dedication, and insightful chapters form the foundation of this work. We also thank the reviewers for their constructive feedback, which was instrumental in enhancing the quality of this book.

Special thanks go to Prof. Chen Change Loy for sharing book writing tips that enhanced the clarity and presentation of this work, and to Dr. Bin Fu for his assistance with LaTeX editing, which greatly facilitated the manuscript preparation.

Finally, we are deeply grateful to Paul Drougas, Kalai Shahethya, and Katherine Moretti at Springer for their invaluable support and guidance throughout the preparation of this book.

Contents

1	Foundations of Vision-Language Models: Concepts and Roadmap	1
	Kaiyang Zhou, Ziwei Liu, and Peng Gao	
1.1	Introduction	2
1.2	The Vision-Language Modeling Paradigm	3
1.3	Navigating the Complexities	7
1.4	Perspectives and Progress	11
	References	15
 Part I Scaling Intelligence: Pre-Training Strategies for Vision-Language Models		
2	InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks	23
	Wenhai Wang, Zhe Chen, Yangzhou Liu, Yue Cao, Weiyun Wang, Xizhou Zhu, Lewei Lu, Tong Lu, Yu Qiao, and Jifeng Dai	
2.1	Introduction	24
2.2	Related Work	27
2.3	Proposed Method	28
2.4	Experiments	38
2.5	Conclusion	48
	References	49
3	Multimodal Large Language Models for Video Understanding	59
	Yi Wang, Jiashuo Yu, Yinan He, Limin Wang, and Yu Qiao	
3.1	Introduction	59
3.2	Related Work	62
3.3	Method	63
3.4	Training Multimodal Data Overview	67
3.5	Experiments	69
3.6	Conclusion and Discussion	84
	References	86

4	Generative Multimodal Models Are In-Context Learners	93
	Yufeng Cui, Quan Sun, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yuezhe Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang	
4.1	Introduction	94
4.2	Related Work	95
4.3	Method	96
4.4	Experiments	101
4.5	Conclusion	106
	References	107
 Part II Shaping Intelligence: Prompting Techniques for Multimodal Adaptation		
5	Differentiable Prompt Learning for Vision-Language Models	115
	Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu	
5.1	Introduction	116
5.2	Related Work	117
5.3	Preliminaries	119
5.4	Context Optimization	121
5.5	Experiments	123
5.6	Conclusion	130
	References	131
6	Test-Time Prompt Tuning for Vision-Language Models	135
	Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao	
6.1	Introduction	136
6.2	Related Work	137
6.3	TPT: Test-Time Prompt Tuning	139
6.4	Experiments	143
6.5	Ablation Study	149
6.6	Conclusion	150
	Appendix	152
	References	160
7	Learning Efficient Feature Adapters for Vision-Language Models ...	165
	Renrui Zhang and Peng Gao	
7.1	Introduction	165
7.2	Related Work	167
7.3	Method	170
7.4	Experiments	175
	References	181

8	Efficient Tuning of Vision Foundation Models with Neural Prompt Search	187
	Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu	
8.1	Introduction	187
8.2	Related Work	189
8.3	Neural Prompt Search	190
8.4	Experiments	195
8.5	Conclusion	203
	References	204
9	Confidence Calibration in Contrastive Vision-Language Models	207
	Shuoyuan Wang, Kaiyang Zhou, and Hongxin Wei	
9.1	Introduction	208
9.2	Background	209
9.3	An Empirical Study	210
9.4	Open-Vocabulary Calibration	212
9.5	Experiments	216
9.6	Discussion	220
9.7	Related Work	221
9.8	Conclusion	222
	References	223
 Part III Applying Intelligence: Real-World Applications of Vision-Language Models		
10	Open-Vocabulary Object Detection Based on Detection Transformers	229
	Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy	
10.1	Introduction	230
10.2	Related Work	232
10.3	OV-DETR	233
10.4	Experiments	239
10.5	Conclusion	245
	References	246
11	Unlocking CLIP for Zero-Shot Dense Segmentation	249
	Chong Zhou, Chen Change Loy, and Bo Dai	
11.1	Introduction	250
11.2	Related Work	251
11.3	Methodology	253
11.4	Experiments	257
11.5	Conclusion	263
	References	263

12	Adapting CLIP for 3D Understanding	267
	Xiangyang Zhu, Renrui Zhang, and Peng Gao	
12.1	Introduction	268
12.2	Related Work	270
12.3	PointCLIP for 3D Understanding	271
12.4	PointCLIP V2 for 3D Open-World Understanding	274
12.5	Experiments	278
12.6	Conclusion	284
	References	284
13	Multimodal Face Generation and Manipulation with Collaborative Diffusion Models	289
	Ziqi Huang, Kelvin C. K. Chan, Yuming Jiang, and Ziwei Liu	
13.1	Introduction	289
13.2	Methodology	290
13.3	Experiments	300
13.4	Conclusion	311
	References	311
14	Boosting Diffusion U-Net with Free Lunch for Text-to-Image and Text-to-Video Generation	315
	Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu	
14.1	Introduction	316
14.2	Related Work	318
14.3	Methodology	319
14.4	Experiments	325
14.5	Conclusion	335
	References	336
15	Text-Conditioned Zero-Shot 3D Avatar Creation and Animation	341
	Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu	
15.1	Introduction	342
15.2	Related Work	344
15.3	Our Approach	345
15.4	Experiments	360
15.5	Discussion	370
	References	371
16	Text-Driven 3D Human Motion Generation	377
	Mingyuan Zhang, Zhongang Cai, Liang Pan, Chenyang Gu, Fangzhou Hong, Xinying Guo, Jiawei Ren, Lei Yang, and Ziwei Liu	
16.1	Introduction	378
16.2	Text-Driven Motion Diffusion Model	379
16.3	Retrieval-Augmented Motion Generation	385
16.4	Fine-Grained Text-Driven Generation	388
16.5	Experiments	391

16.6 Conclusion	398
References	400
17 Text-Driven Scene Generation	403
Zhaoxi Chen, Guangcong Wang, and Ziwei Liu	
17.1 Introduction	404
17.2 Related Work	406
17.3 Panoramic Scene Representation	407
17.4 Scene Generation from Texts	408
17.5 Evaluations	414
17.6 Downstream Applications	420
17.7 Conclusion	423
References	423
Index	427

Contributors

Anima Anandkumar California Institute of Technology, Pasadena, CA, USA

Zhongang Cai S-Lab, Nanyang Technological University and SenseTime Research, Singapore, Singapore

Yue Cao School of Computer Science, Nanjing University, Nanjing, China

Kelvin C.K. Chan S-Lab, Nanyang Technological University, Singapore, Singapore

Zhaoxi Chen Nanyang Technological University, Singapore, Singapore

Zhe Chen School of Computer Science, Nanjing University, Nanjing, China

Yufeng Cui Beijing Academy of Artificial Intelligence, Beijing, China

Bo Dai Shanghai Artificial Intelligence Laboratory, Shanghai, China

Jifeng Dai Department of Electronic Engineering, Tsinghua University, Beijing, China

Peng Gao Shanghai Artificial Intelligence Laboratory, Shanghai, China

Tom Goldstein University of Maryland, College Park, MD, USA

Chenyang Gu S-Lab, Nanyang Technological University, Singapore, Singapore

Xinying Guo S-Lab, Nanyang Technological University, Singapore, Singapore

Yinan He Shanghai Artificial Intelligence Laboratory, Shanghai, China

Fangzhou Hong S-Lab, Nanyang Technological University, Singapore, Singapore

Chen Huang Apple Inc., Cupertino, CA, USA

De-An Huang Nvidia Corporation, Santa Clara, CA, USA

Tiejun Huang Peking University, Beijing, China

Ziqi Huang S-Lab, Nanyang Technological University, Singapore, Singapore

Yuming Jiang S-Lab, Nanyang Technological University, Singapore, Singapore

Wei Li S-Lab, Nanyang Technological University, Singapore, Singapore

Jingjing Liu Tsinghua University, Beijing, China

Yangzhou Liu School of Computer Science, Nanjing University, Nanjing, China

Ziwei Liu S-Lab, Nanyang Technological University, Singapore, Singapore

Chen Change Loy S-Lab, Nanyang Technological University, Singapore, Singapore

Lewei Lu SenseTime Research, Hong Kong, China

Tong Lu School of Computer Science, Nanjing University, Nanjing, China

Zhengxiong Luo Beijing Academy of Artificial Intelligence, Beijing, China

Weili Nie Nvidia Corporation, Santa Clara, CA, USA

Liang Pan S-Lab, Nanyang Technological University, Singapore, Singapore

Yu Qiao Shanghai AI Laboratory, Shanghai, China

Yongming Rao Beijing Academy of Artificial Intelligence, Beijing, China

Jiawei Ren S-Lab, Nanyang Technological University, Singapore, Singapore

Manli Shu University of Maryland, College Park, MD, USA

Chenyang Si Nanyang Technological University, Singapore, Singapore

Quan Sun Beijing Academy of Artificial Intelligence, Beijing, China

Guangcong Wang Great Bay University, Dongguan, China

Limin Wang Nanjing University, Nanjing, China

Shuoyuan Wang Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen, China

Weiyun Wang School of Computer Science, Fudan University, Shanghai, China
Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China

Xinlong Wang Beijing Academy of Artificial Intelligence, Beijing, China

Yi Wang Shanghai AI Laboratory, Shanghai, China

Yueze Wang Beijing Academy of Artificial Intelligence, Beijing, China

Hongxin Wei Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen, China

Chaowei Xiao University of Wisconsin, Madison, WI, USA

Jingkang Yang S-Lab, Nanyang Technological University, Singapore, Singapore

Lei Yang SenseTime Research, Hong Kong, China

Jiashuo Yu Shanghai AI Laboratory, Shanghai, China

Qiyang Yu Tsinghua University, Beijing, China

Zhidong Yu Nvidia Corporation, Santa Clara, CA, USA

Yuhang Zang Shanghai AI Laboratory, Shanghai, China

Fan Zhang Beijing Academy of Artificial Intelligence, Beijing, China

Mingyuan Zhang S-Lab, Nanyang Technological University, Singapore, Singapore

Renrui Zhang The Chinese University of Hong Kong, Hong Kong, China

Xiaosong Zhang Beijing Academy of Artificial Intelligence, Beijing, China

Yuanhan Zhang S-Lab, Nanyang Technological University, Singapore, Singapore

Chong Zhou S-Lab, Nanyang Technological University, Singapore, Singapore

Kaiyang Zhou Hong Kong Baptist University, Hong Kong, China

Xiangyang Zhu Shanghai Artificial Intelligence Laboratory, Shanghai, China

Xizhou Zhu Department of Electronic Engineering, Tsinghua University, Beijing, China

Chapter 1

Foundations of Vision-Language Models: Concepts and Roadmap



Kaiyang Zhou, Ziwei Liu, and Peng Gao

Abstract Vision-language models have significantly advanced the field of artificial intelligence by bridging the gap between visual and textual understanding. These models can enable wide-ranging applications including image recognition, object detection, scene understanding, visual content generation and editing in both 2D and 3D, and visual question answering, to name a few. This chapter introduces the foundational concepts underlying these models, emphasizing their unique ability to learn multimodal representations through novel neural network architectures and large-scale data pre-training. We explore the vision-language modeling paradigm, highlight key challenges in feature alignment, scalability, and data and evaluation, and review notable progress in the field. In addition, we discuss the limitations of current approaches, from computational inefficiencies to ethical concerns, and outline potential directions for future research. This chapter serves as a roadmap for understanding the field's core principles and its transformative potential in AI applications.

Keywords Vision-Language Model · Multimodal Learning · Feature Alignment · Large-Scale Pre-training

K. Zhou (✉)

CS Department, Hong Kong Baptist University, Hong Kong, China
e-mail: kychou@hkbu.edu.hk

Z. Liu

S-Lab, Nanyang Technological University, Singapore, Singapore
e-mail: ziwei.liu@ntu.edu.sg

P. Gao

Shanghai Artificial Intelligence Laboratory, Shanghai, China
e-mail: gaopeng@pjlab.org.cn

1.1 Introduction

Vision and language are two deeply intertwined capabilities in human intelligence. In artificial intelligence (AI) research, these two capabilities have traditionally been studied in divided fields, which are known as *computer vision* and *natural language processing*. Specifically, computer vision focuses on interpreting images, such as recognizing objects in images [1, 2] or identifying their pixel-wise locations [3, 4]; on the other hand, natural language processing aims to analyze and generate language, such as predicting the sentiment of customer reviews [5] or summarizing news articles or books [6].

Fundamentally, humans do not learn concepts from a single modality alone. Rather, the learning process often involves interactions between vision and language. For example, when children learn the concept of *apple*, they usually receive a combination of visual and linguistic information: a parent may show a real apple or a picture of an apple and says something related to it, like “this is an apple,” “do you want to eat apple,” “it is red, round, and sweet”; overtime, this process is repeated in various contexts, such as during a meal or bedtime story, and as a result, the child begins to associate the visual characteristics of apples with the linguistic label. Moreover, existing cognitive research shows that children acquire concepts by forming associations between words and visual scenes through everyday communication [7, 8]. Therefore, integrating vision and language is a natural step toward building artificial general intelligence or AGI.

The emergence of large vision-language models (VLMs) has dramatically changed the landscape of AI research, which enables numerous novel applications, such as detecting objects of arbitrary categories [9], generating photorealistic images based on text descriptions [10], or using natural language to instruct robots to perform grasping [11], navigation [12], or even more complicated operations like surgery [13]. Compared with early “VLMs,” such as those developed during the 2010s [14–16], modern VLMs are built with a much larger *scale* in terms of both model architecture and training data [17–21]. For example, the size of modern VLMs has grown exponentially, evolving from just a few million parameters to hundreds of millions, and even billions. Similarly, the scale of training data has expanded to encompass billions of examples [22], a practice that has now become standard for training commercial models. Research has suggested that it is the sheer scale of modern VLMs—both their parameter size and the volume of training data—that allows these models to effectively learn extensive and generalizable world knowledge [23, 24]. With the broad knowledge learned from massive datasets, VLMs are versatile and can be adapted to a wide range of downstream applications, spanning both discriminative and generative tasks, as well as extending from 2D to 3D domains.

While the scale of modern VLMs has unlocked unprecedented capabilities, achieving this scale has presented significant challenges, spanning algorithmic, computational, and data-related dimensions. From an algorithmic perspective, designing architectures capable of integrating vision and language is nontrivial.

Furthermore, when adapting VLMs to downstream applications, one also needs to handle problems arisen from task-specific designs and modality gaps, such as connecting vision and language to human pose [25] or motion [26]. Computationally, training VLMs at such scales demands enormous compute resources, which limit the widespread adoption of these models in practice. To address this challenge, advancements in efficient training techniques and adaptation methods, such as prompting [27, 28], are essential. On the data front, curating and managing billion-level training data need to address various issues like data noise, bias, and diversity to ensure robust and safe learning.

This chapter introduces fundamental concepts of VLMs, with a particular focus on feature alignment, scalability, and data and evaluation. These concepts aim to provide readers with a solid foundation for understanding the problems and algorithms presented later in this book. The chapter is organized as follows. In Sect. 1.2, we discuss the evolution of the vision-language modeling paradigm. In Sect. 1.3, we highlight key challenges related to feature alignment, scalability, and data and evaluation. In Sect. 1.4, we review representative advancements and provide insights into the current state of progress.

1.2 The Vision-Language Modeling Paradigm

This section aims to provide a background on the learning paradigm behind vision-language models (VLMs). Specifically, this section discusses the evolution of the learning paradigm, core components, popular model architectures, and finally, training objectives and strategies widely used by the community.

1.2.1 The Evolution

The rise of VLMs can be largely attributed to the advancements in *pre-training*, which has become a cornerstone of modern AI systems. Pre-training involves training a model on large-scale data for a generic task, such as predicting masked words in text [29] or missing patches in images [30]. This step allows the model to learn generic patterns, structures, and relationships in the data, enabling it to develop representations that are broadly applicable across different downstream tasks.

Evolution of Vision Models

The rise of pre-training in computer vision technically began with ImageNet [31], a large-scale benchmark that spurred numerous breakthroughs in visual recognition. AlexNet’s success with convolutional neural networks (CNNs) marked a turning point [1], which demonstrated the power of deep learning by setting new performance records on ImageNet. Subsequently, the introduction of ResNet [2] addressed the notorious vanishing gradient problem with residual connections,

enabling deeper, more expressive networks. More recently, Vision Transformer (ViT) [32] has redefined the field by treating image patches as sequences and learning representations in a fully self-attention manner, achieving state-of-the-art performance on various vision domains [33–36].

The development of these models was underpinned by the growing availability of massive datasets and computational power, both of which enabled increasingly sophisticated models to be trained at scale. Essentially, the remarkable generalization ability of these pre-trained models marks a broader trend in the evolution of modern computer vision. As these models continued to push the boundaries of visual recognition, they also inspired a rethinking of model architectures and learning paradigms. With advances in unsupervised and self-supervised learning techniques [37–44], the vision community began exploring ways to pre-train models on vast amounts of unlabeled data, which has not only expanded the scope of vision models but also laid the foundation for VLMs.

Evolution of Language Models

The evolution of pre-training in NLP occurred later than in computer vision. However, the trajectory from early word representation models to the emergence of large language models has not only transformed NLP but also had a profound impact on the vision domain, exemplified by models like the previously mentioned ViT.

Initially, foundational models like Word2Vec [45] and GloVe [46] represented words as dense, fixed vectors, capturing semantic relationships through word co-occurrence patterns. Then, the community shifted toward contextual embeddings led by models like ELMo [47]. ELMo’s dynamic word representations, derived from a bidirectional LSTM, addressed the issue of static embeddings by taking into account the context in which a word appeared. This breakthrough allowed models to produce different vector representations of a word depending on its surrounding words, significantly improving performance on tasks requiring fine-grained semantic understanding.

The advent of sequence-to-sequence (seq2seq) models [48] and the subsequent development of Transformers [49] introduced another pivotal transformation. Seq2seq models, which are based on encoder–decoder architectures, enabled significant progress in tasks like machine translation, as they could encode an entire sequence of words into a fixed-size vector and then decode it into another sequence. However, the reliance on recurrent layers weakens seq2seq models’ ability of handling long-range dependencies. To remedy this weakness, transformers replaced recurrence with self-attention mechanisms, which better capture long-sequence relationships and enable faster training due to parallelization. Representative language transformers include BERT [29] and GPT [50], which are based on masked language modeling and autoregressive learning (a.k.a. next-token prediction), respectively.

Convergence of Vision and Language Models

The advancements of pre-trained models in both vision and language communities have largely facilitated the development of VLMs, which integrate both visual

and linguistic modalities in a unified framework. Early pioneering efforts, e.g., DeViSE [15], sought to bridge the modality gap by aligning visual and textual features through shared embeddings. However, these models were constrained by their limited scalability and lacked the capacity to ingest large, diverse datasets.

The landscape of VLMs has been dramatically reshaped by the emergence of models like CLIP [17] and ALIGN [51], which learned dual-encoder architectures via large-minibatch contrastive learning to build joint representations of images and text. These models demonstrated unprecedented generalization ability in zero-shot recognition, unlocking the potential for diverse downstream applications, such as image classification, retrieval, and even generative tasks like image generation from textual descriptions [10].

The field continued to progress with more powerful and versatile VLMs like Flamingo [20] and GPT-4V [52]. Flamingo, for instance, was developed by first attaching a pre-trained vision model to a pre-trained language model and then fine-tuning the combined model with image captioning data. The resulting model achieved impressive performance on perception and reasoning tasks, such as image captioning and visual question answering, highlighting the model's capacity to integrate vision with language. GPT-4V, which was probably built in a similar way,¹, demonstrated far more powerful capabilities in perception and reasoning. As VLMs continue to evolve, they hold the promise of creating more intuitive and versatile AI systems that can better understand and interact with the world as humans do, processing and relating visual and linguistic information in a manner that is more akin to human cognition.

1.2.2 Core Components

This section discusses the primary building blocks that enable VLMs to process, align, and integrate visual and textual information. The building blocks are categorized into three groups: feature representation, feature alignment and fusion, training objectives, and data.

Feature Representation

Modern VLMs rely on specialized architectures to extract compact features from both images and text.

For vision, commonly used backbones include CNNs such as ResNet [2], which excel in extracting spatial hierarchies, and ViT [32], which process images as sequences of patches and leverage self-attention to capture global dependencies. These backbones are often pre-trained on large-scale datasets, such as ImageNet [31], to learn general-purpose visual features.

¹ The technical detail of GPT-4V is not available to the public.

For language, transformer-based architectures dominate the field as they have been proved extremely effective in processing sequential data [49]. The language model is either learned jointly with the vision model or pre-trained first on text datasets with self-supervised learning and then combined with a vision model for further fine-tuning. For instance, in CLIP [17], the language model is a transformer trained jointly with the vision model on massive image-pairs data; in contrast, models like Flamingo [20] and LLaVA [18] employ a pre-trained language model and condition the textual output on visual features.

Feature Alignment and Fusion

Bridging the gap between vision and language requires specialized mechanisms to align and fuse features from these modalities. There are three popular architectures:

- **Dual-Encoder Architectures:** These models use separate encoders to process visual and textual information, respectively, and align these two modalities in a shared embedding space, often via contrastive learning (e.g., CLIP [17]).
- **Cross-Attention Architectures:** These mechanisms enable direct interactions between visual and textual features, allowing models to reason jointly about both modalities. Cross-attention is typically applied to intermediate self-attention layers in a language transformer. Representative work includes Flamingo [20] and InstructBLIP [53].
- **Unified Architectures:** In this paradigm, both modalities are processed together within a unified transformer model (e.g., UNITER [54]). This is achieved by tokenizing both images and text and concatenating their features in a sequence.

Training Objectives

Training objectives play a key role in the process of aligning different modalities. Further, we discuss popular choices:

- **Contrastive Learning:** This approach aligns image-text pairs by maximizing their similarity in a joint embedding space (e.g., CLIP [17]).
- **Masked Modeling:** Representative masked modeling methods include masked image modeling [30] and masked language modeling [29]. The key idea in masked modeling is to train the model to predict masked tokens across both modalities (e.g., UNITER [54], BLIP [55], and VideoBERT [56]).
- **Autoregressive Learning:** This paradigm is similar to language modeling where the objective is to predict the next token given previous ones. In VLMs, this paradigm is often applied with image captioning data, i.e., training the model to caption images (e.g., LLaVA [18] and Flamingo [20]).

Data

Data are fundamental to VLMs, as they provide the essential information the model needs to learn about the world. Key types of data include image-text pairs, visual question answering (VQA) datasets, instructional data, and video-text datasets. Each type targets specific skills, from captioning and reasoning to

temporal understanding, collectively advancing multimodal learning. We provide more detailed discussions further.

- **Image-Text Pair Data:** This type of data provides aligned visual and textual inputs for learning multimodal representations. Medium-sized datasets like COCO Captions [57] and Flickr30k [58] offer human-written captions for hundreds of thousands of images, while large-scale datasets such as LAION-5B [22] provide billions of web-collected pairs. These datasets drive tasks like image captioning and retrieval and have greatly benefited VLMs like CLIP [17] and ALIGN [51], which use contrastive learning to map images and text into a shared space.
- **VQA Data:** VQA datasets (e.g., VQA v2 [59], OK-VQA [60], and GQA [61]) evaluate a model's ability to reason about images by answering questions in natural language. These datasets push models to integrate visual understanding, object relationships, and commonsense reasoning.
- **Instructional and Dialogue Data:** This type of data bridges perception and interaction, supporting applications like multimodal assistants and task-based reasoning. For instance, LLaVA-Instruct [18] provides multimodal instructions grounded in visual inputs; Visual Dialog [62] introduces conversational reasoning; RefCOCO/RefCOCOG [63] focus on object identification through language.
- **Video-Text Data:** Video-text datasets extend VLMs into the temporal domain, pairing videos with captions, annotations, or subtitles. Examples include HowTo100M [64], which pairs instructional videos with subtitles, and YouCook2 [65], which provides step-by-step video annotations.

1.3 Navigating the Complexities

1.3.1 Algorithmic Challenges

Building effective VLMs is nontrivial, which requires addressing multiple algorithmic challenges. This section explores these challenges in depth, offering insights into the complexities and opportunities that lie at the heart of algorithmic innovation.

Bridging the Modality Gap

One of the most fundamental challenges in vision-language modeling is bridging the gap between visual and textual modalities. In particular, images and videos encode spatial and perceptual information in pixels, while language is inherently symbolic and abstract. Effective models must reconcile these differences to create unified representations that capture the essence of both modalities. In the literature, this has often been achieved by learning a joint embedding space through, for example, large-scale contrastive learning [17, 51]. Yet, achieving robust alignment remains elusive, particularly when dealing with ambiguous or sparse data. For example, the

word *jaguar* can refer to the animal or the car, which requires the model to figure out the visual context and linguistic cues.

Model Architecture Design

Designing architectures tailored to multimodal tasks is another significant challenge. Unlike unimodal models, VLMs must process and integrate information from vastly different sources. Recent CLIP-like models [17] that incorporate modality-specific encoders for learning shared latent spaces have shown promise. However, optimizing these architectures for vision-language tasks requires careful attention to cross-modal interactions. The emergence of transformers has dramatically changed the landscape of multimodal learning as different types of data can be cast into a sequence, which are then processed together by the self-attention mechanism [49]. However, transformers also have notable drawbacks. In particular, transformers are computationally and memory-intensive, with quadratic complexity in sequence length, making it inefficient for very long sequences [66, 67].

Task-Specific Adaptation

Adapting VLMs to downstream tasks presents significant challenges due to their massive parameter counts and the risk to overfitting. Specifically, VLMs often consist of hundreds of millions of parameters while downstream tasks typically involve datasets of moderate scales, not to mention the domain gap between pre-training and fine-tuning datasets that further complicates the adaptation process. A popular solution is to employ parameter-efficient fine-tuning methods, such as prompt learning [28] or adapter tuning [68]. However, these methods are often criticized for their weak generalization ability [69]. Test-time prompt tuning can improve generalization, but it would increase computational overhead [70]. In addition to the generalization issue, practitioners also need to develop task-specific modules when adapting image-based VLMs to, for example, videos, medical images, or even 3D data.

Interpretability and Explainability

In the context of VLMs, interpretability involves designing mechanisms that explain how a model processes and integrates multimodal information, while explainability focuses on generating human-readable justifications for predictions. As VLMs are increasingly deployed in sensitive and high-stakes applications, the need for interpretability and explainability has grown because users and stakeholders must be able to understand and trust the decisions made by these models. However, research along this line largely lags behind. Traditional methods like saliency maps [71] are unreliable for VLMs as the visualization explains nothing about decision-making except where the model is looking at—which does not answer why a particular image region is deemed important by the model.

Continual Learning Capability

Continual learning is essential for VLMs to adapt to the dynamic and ever-changing demands of real-world applications without forgetting previously acquired knowledge. For instance, a VLM deployed in autonomous vehicles must continuously update its understanding of new road signs, changes in infrastructure,

and evolving environmental conditions. However, achieving this adaptability is far from straightforward. Updating VLMs incrementally poses challenges such as managing the increasing computational and memory demands, ensuring scalability, and addressing imbalances between old and new data. Advancing continual learning in VLMs requires innovative approaches, including memory-efficient architectures, selective fine-tuning strategies, and algorithms that maintain a delicate balance between preserving past knowledge and integrating new information seamlessly.

Proprietary Models

Proprietary models pose significant challenges to customization, transparency, and research reproducibility. Many state-of-the-art models, such as GPT-4o and other commercial systems, withhold access to their weights and detailed architectures. While these models deliver exceptional performance, their closed nature impedes adaptation to domain-specific tasks. This limitation has spurred interest in black-box machine learning techniques [72, 73], where users interact with models exclusively through APIs, with no visibility into underlying parameters or intermediate computations. Without access to model weights, developers are often forced to rely on costly, trial-and-error prompt engineering, which is time-intensive and frequently yields suboptimal outcomes. Furthermore, the black-box nature of these systems exacerbates challenges in debugging, interpreting outputs, and addressing inherent biases, hindering their broader applicability and trustworthiness.

1.3.2 Computational Challenges

Building VLMs is computationally intensive, requiring significant resources for training, inference, and deployment. Addressing these computational challenges is essential for scaling VLMs while ensuring efficiency, accessibility, and sustainability. This section delves into the core computational challenges.

Scalability in Training and Inference

The unprecedented size of modern VLMs, with billions of parameters, poses serious scalability issues during training and inference. Training such models requires massive computational resources, often relying on high-performance GPUs or TPUs and vast amounts of memory. For instance, training GPT-4-like systems demands weeks of computation across thousands of accelerators, with costs running into millions of dollars [52]. Inference, particularly for real-time applications, is equally demanding, as each query involves processing a large model with substantial memory and compute overhead. Practitioners may resort to memory reduction methods, such as zeroth-order optimizers [74], to cut memory usage during training, and quantization techniques [75, 76] to compress model weights, thus facilitating deployment.

Handling Long Sequences and High-Resolution Inputs

VLMs frequently deal with high-dimensional inputs, such as long videos or high-resolution images, which significantly increase computational costs. For example, analyzing a feature-length film or processing gigapixel medical imagery requires substantial memory and compute resources. Transformers, the most popular backbone of VLMs, face quadratic complexity in sequence length, making them inefficient for processing long sequences.

Latency and Real-Time Constraints

Many applications of VLMs, such as autonomous vehicles, interactive assistants, or video analytics, require real-time or low-latency inference. However, the computational demands of large models often lead to delays, rendering them unsuitable for latency-sensitive tasks. Optimizations like model caching, adaptive inference, and edge computing can reduce response times, but they also introduce complexities in deployment.

Distributed and Federated Learning

Training VLMs in a distributed environment across multiple computational nodes introduces challenges in coordination, synchronization, and communication overhead. Gradient updates must be efficiently shared across nodes, often leading to bottlenecks in large-scale setups. Federated learning adds another layer of complexity, enabling decentralized training where data remain on local devices to preserve privacy. This approach is particularly relevant for applications involving sensitive multimodal data, such as medical imaging or personal photos.

1.3.3 Data Challenges

High-quality and diverse datasets are critical to the success of VLMs. However, collecting, curating, and utilizing such datasets present significant challenges that can impact model performance and generalizability. This section explores the key data challenges in building VLMs.

Dataset Diversity and Representativeness

The effectiveness of VLMs relies heavily on the diversity and representativeness of their training data. Multimodal datasets must encompass a wide range of visual and linguistic contexts to enable generalization across different domains and tasks. However, most publicly available datasets are biased toward specific geographic regions, cultures, or domains, such as Western-centric imagery [77]. This bias can cause the model to fail in underrepresented contexts. Addressing this issue requires curating datasets that are more inclusive and representative of global diversity.

Scale and Quality of Multimodal Data

The sheer size of multimodal datasets often comes at the expense of quality. For instance, LAION-5B [22] aggregated images and text captions from diverse sources, but the captions can be noisy, incomplete, or irrelevant to the associated images.

Training on such data risks propagating errors and biases into the model. Cleaning and filtering multimodal datasets at scale are nontrivial, requiring robust automated pipelines or substantial human oversight, both of which are resource-intensive.

Long-Tail and Rare Concepts

A common challenge in VLMs is capturing rare or long-tail concepts that occur infrequently in datasets. For instance, identifying rare species in wildlife images or understanding niche technical terminology in scientific text requires exposure to relevant examples during training. However, the uneven distribution of data often leads to models overfitting to frequent concepts while underperforming on rare ones [78]. Potential solutions include data augmentation, balanced sampling, and synthetic data generation.

Ethical and Privacy Concerns

The collection of multimodal data often involves scraping content from the internet, raising significant ethical and privacy concerns. Datasets may inadvertently include sensitive or private information, such as personal photos, location metadata, or copyrighted material, often used without explicit consent. Moreover, unregulated datasets risk propagating biases or offensive content, leading to harmful outputs in sensitive applications like healthcare or law enforcement. To address these concerns, developers are strongly suggested to establish clear guidelines for data collection and annotation and promote transparency through detailed dataset documentation. Ensuring ethical data practices is essential for building socially responsible and trustworthy AI systems.

1.4 Perspectives and Progress

1.4.1 Pre-training and Representation Learning

A major breakthrough in VLM pre-training is CLIP [17], in which two modality-specific encoders were trained from scratch on 400 million image-text pairs using contrastive learning. The core idea of CLIP, i.e., learning a shared representation space for images and text, has actually been investigated more than a decade ago [14, 16]. The success of CLIP can be attributed to three advances in the field: transformers [49], contrastive learning [42], and web-scale multimodal datasets. Transformers, particularly their self-attention mechanism, played a key role in advancing pre-training in natural language processing. Compared with the recurrence design in recurrent neural networks—which once were the mainstream backbone of language models—self-attention in transformers enables parallel processing of sequential data and thus largely facilitates the learning of contextualized representations suitable for language understanding. When it comes to contrastive learning, this concept has already been extensively studied in the literature of self-supervised representation learning in computer vision where the main idea is

to train a model to generate similar features for the same image under different augmentations [44, 79, 80]. The innovations achieved in prior contrastive learning research laid the foundation of CLIP. Chapter 2 studies the scaling problem by extending CLIP with a larger vision encoder and an ability to handle high-resolution images.

In addition to image–text contrastive learning, the community has also identified several other effective learning objectives, including image–text matching, masked language modeling, and image-conditioned text generation. Image–text matching focuses on determining whether a given image and text pair are semantically aligned and has often been formulated as a binary classification problem, which enhances the model’s ability to understand fine-grained cross-modal relationships. Masked language modeling, widely popularized by BERT [29], has been extended to multimodal settings by masking text tokens and conditioning their reconstruction on both the visual context and the surrounding text. This approach has proven effective for learning contextualized representations that incorporate both modalities [55, 81]. Similarly, image-conditioned text generation tasks, such as captioning, train models to produce descriptive text based on visual input, further improving the fluency and relevance of language generation. These learning objectives can be used together to enhance pre-training. Chapter 3 discusses a video-language modeling approach, which integrates masked video modeling, cross-modal contrastive learning, and next-token prediction into a unified paradigm.

Unlike CLIP-like models that are trained from scratch, another line of research focuses on combining a vision encoder, typically pre-trained on ImageNet to capture generic visual features, with a pre-trained large language model (LLM). This was largely motivated by the rise of powerful LLMs that can generate natural language indistinguishable from human writings [82]. The architecture is analogous to the encoder–decoder widely used in natural language processing: the vision model can be seen as an encoder, which extracts features from raw images; while the language model serves as a decoder. A common strategy for aligning the features of these models is to connect them with a linear projection layer [18] and train the combination to perform image-conditioned text generation, followed by domain-specific fine-tuning (e.g., fine-tuning on chatbot datasets consisting of dialogs). Instead of using a simple linear layer, the community has also developed other sophisticated connectors to bridge the modality gap, such as cross-modal attention [20] or learnable-query transformer [83]. Chapter 4 presents a large VLM of this kind, which demonstrates strong in-context learning and reasoning abilities.

1.4.2 Prompting and Fine-Tuning for Customization

As VLMs continue to grow in scale and capability, a critical challenge lies in adapting these models to downstream tasks and domains. Prompting has emerged as a popular paradigm for model customization. The key idea is to tune the prompt to steer the model’s behavior without changing the massive pre-trained weights, which

can significantly reduce the risk of overfitting as downstream datasets often have moderate sizes, which are insufficient for full-parameter fine-tuning.

There are two groups of prompting approaches: prompt engineering and prompt learning. Prompt engineering involves manually crafting input templates to guide the model toward desired outputs. For example, by framing an image classification task as a fill-in-the-blank problem, the prompt can include descriptive text with placeholders that contain the names of objects of interest, such as “a sketch photo of a {CLASS}.” While straightforward and often effective in zero-shot settings, this method can be extremely time-consuming because the selection of prompts solely relies on trial-and-error, i.e., developers need to manually tune the prompt using a validation set while closely observing the change of performance. To improve the model’s reasoning ability, practitioners can adopt more advanced prompting techniques, such as chain-of-thought prompting [24] that breaks down complex tasks into a series of intermediate reasoning steps.

In contrast, prompt learning automates the process by treating prompts as learnable parameters optimized during fine-tuning. Here, prompts can take the form of text embeddings or visual tokens, which are taken as input by the model. These learnable prompts are trained alongside task-specific data, enabling the model to adapt flexibly to new tasks or domains without modifying its pre-trained parameters. Since the size of learnable prompts is much smaller than that of the pre-trained weights, training can be done under resource-constrained settings. Chapter 5 discusses a differentiable prompt learning approach for adapting VLMs, which achieves promising results in few-shot learning and domain generalization. Chapter 8 introduces a neural architecture search-based approach to explore the space of potential prompt designs and identify the optimal configuration. Chapter 7 presents an alternative approach based on learning a lightweight adapter module for task adaptation. Chapter 9 investigates the calibration ability of different prompt learning methods.

A major challenge in prompt learning is the generalization problem, which means the prompt learned from one dataset may not generalize well to new concepts of the same dataset. For example, a prompt learned for the classification of scenes like airport and cathedral may not work well for the classification of new scenes like wind farm and railway station, despite the nature of the task remaining the same, i.e., scene recognition. One solution for solving this problem is dynamic prompt learning, which adjusts the prompt in real-time based on the image input, e.g., one can learn a conditional neural network that generates a prompt based on each image input [69]. Another solution is to apply test-time training techniques. Chapter 6 discusses a test-time prompt learning approach, which updates the prompt parameters using self-supervised learning.

1.4.3 *Real-World Impact and Use Cases*

The power of VLMs mainly comes from their design that connects images with natural language, which creates numerous possibilities in wide-ranging multimodal applications. This section reviews novel applications enabled by the advances of VLMs and discusses their real-world impact.

Open-Vocabulary Perception

Open-vocabulary perception refers to the ability of a model to recognize and interact with a potentially infinite set of categories, including those unseen during training. VLMs like CLIP have shown great potential in zero-shot recognition where the model can be repurposed to any recognition tasks by simply changing the category names in the input prompt. However, adapting these models to *dense* prediction tasks like detection and segmentation presents significant challenges—this is because VLMs are typically trained on noisy image-level labels, such as coarse captions, without using dense annotations like bounding boxes. Dense prediction tasks require region-specific or pixel-level understanding, which CLIP is not explicitly trained for. This lack of spatial granularity in training data results in poor performance when directly applying CLIP to tasks requiring precise localization. Chapters 10 and 11 discuss how CLIP can be applied to dense prediction tasks. Specifically, Chap. 10 introduces a transformer-based open-vocabulary detection model, in which the detection output is conditioned on CLIP’s text encoder, allowing the model to handle arbitrary categories. Chapter 11 presents a segmentation model called MaskCLIP, which leverages CLIP’s image encoder to extract dense features and the text encoder to synthesize classification weights.

3D Understanding and Generation

3D understanding and generation involve enabling models to process, interpret, and create three-dimensional content, which is crucial for applications such as virtual reality, gaming, and robotics. Unlike 2D tasks, which are primarily based on image representations, 3D tasks require understanding complex spatial relationships, depth, and geometric structures. This added complexity poses significant challenges for VLMs as their pre-training relies on 2D image–text pairs that lack the detailed spatial and volumetric context inherent to 3D data. Bridging this gap requires innovative techniques to adapt 2D-trained VLMs for 3D tasks or to develop hybrid approaches that combine 3D-specific representations with existing 2D capabilities. Chapter 12 discusses a simple approach that projects 3D data onto the 2D space where CLIP can be directly applied for object recognition. Chapter 15 discusses text-conditioned 3D avatar creation, introducing a framework that generates and animates realistic 3D avatars driven by natural language descriptions. Chapter 16 focuses on text-driven 3D human motion generation, presenting a model capable of producing diverse and controllable motion sequences from textual inputs.

Generative Models and Creativity

Generative models represent a pivotal advancement in AI, empowering systems to synthesize new content across modalities such as images, videos, and 3D

scenes. These models are not merely tools for creating novel outputs but also act as enablers of human creativity, enhancing applications in design, entertainment, and storytelling. By leveraging VLMs, generative systems can integrate textual instructions with visual understanding, providing fine-grained control over the generated content. However, various challenges need to be addressed, such as maintaining coherence, ensuring fidelity to input prompts, and balancing creative freedom with realism in image content. Chapter 13 explores collaborative diffusion models that enable multi-modal face generation and manipulation by combining text and mask-driven inputs. Chapter 14 discusses techniques for boosting diffusion models to enhance the quality of text-to-image and text-to-video generation without increasing computational overhead. Chapter 17 focuses on text-driven scene generation, presenting methods to synthesize complex visual environments based on descriptive textual prompts, advancing the applicability of generative models in immersive applications.

Further Reading

Interested readers may refer to the following material:

- [84] for a tutorial on multimodal foundation models
- [85] for a survey on VLMs and their applications in computer vision
- [86] for a survey on prompting methods in natural language processing

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Proces. Syst.* **25** (2012)
2. 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)
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2016)
4. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440 (2015)
5. Dos Santos, C., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78 (2014)
6. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252* (2016)
7. Bloom, P.: *How Children Learn the Meanings of Words*. MIT Press, New York (2002)
8. Smith, L., Yu, C.: Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition* **106**(3), 1558–1568 (2008)
9. Gu, X., Lin, T.-Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921* (2021)

10. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning, pp. 8821–8831. PMLR, New York (2021)
11. Stone, A., Xiao, T., Lu, Y., Gopalakrishnan, K., Lee, K.-H., Vuong, Q., Wohlhart, P., Kirmani, S., Zitkovich, B., Xia, F., et al.: Open-world object manipulation using pre-trained vision-language models. arXiv preprint arXiv:2303.00905 (2023)
12. Huang, C., Mees, O., Zeng, A., Burgard, W.: Visual language maps for robot navigation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 10608–10615. IEEE, New York (2023)
13. Wang, G., Bai, L., Nah, W.J., Wang, J., Zhang, Z., Chen, Z., Wu, J., Islam, M., Liu, H., Ren, H.: Surgical-LVLM: learning to adapt large vision-language model for grounded visual question answering in robotic surgery. arXiv preprint arXiv:2405.10948 (2024)
14. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. *Adv. Neural Inf. Proces. Syst.* **26** (2013)
15. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. *Adv. Neural Inf. Proces. Syst.* **26** (2013)
16. Elhoseiny, M., Saleh, B., Elgammal, A.: Write a classifier: zero-shot learning using purely textual descriptions. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2584–2591 (2013)
17. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR, New York (2021)
18. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. *Adv. Neural Inf. Proces. Syst.* **36** (2024)
19. Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
20. Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. *Adv. Neural Inf. Proces. Syst.* **35**, 23716–23736 (2022)
21. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altschmidt, J., Altman, S., Anadkat, S., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
22. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: an open large-scale dataset for training next generation image-text models. *Adv. Neural Inf. Proces. Syst.* **35**, 25278–25294 (2022)
23. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020)
24. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Proces. Syst.* **35**, 24824–24837 (2022)
25. Hong, F., Zhang, M., Pan, L., Cai, Z., Yang, L., Liu, Z.: Avatarclip: zero-shot text-driven generation and animation of 3d avatars. arXiv preprint arXiv:2205.08535 (2022)
26. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: text-driven human motion generation with diffusion model. arXiv preprint arXiv:2208.15001 (2022)
27. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
28. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *Int. J. Comput. Vis.* **130**(9), 2337–2348 (2022)
29. Devlin, J.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

30. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009 (2022)
31. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, New York (2009)
32. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N.: An image is worth 16×16 words: transformers for image recognition at scale. In: *International Conference on Learning Representations* (2021)
33. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: *European Conference on Computer Vision*, pp. 280–296. Springer, Berlin (2022)
34. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6881–6890 (2021)
35. Mehta, S., Rastegari, M.: Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178* (2021)
36. 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: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022 (2021)
37. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430 (2015)
38. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802 (2015)
39. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5147–5156 (2016)
40. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: feature learning by inpainting. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544 (2016)
41. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. *arXiv preprint arXiv:1605.09782* (2016)
42. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018)
43. Chen, T., Zhai, X., Ritter, M., Lucic, M., Houlsby, N.: Self-supervised gans via auxiliary rotation loss. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12154–12163 (2019)
44. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738 (2020)
45. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
46. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (2014)
47. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018)
48. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014)

49. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., New York (2017)
50. 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. Proces. Syst.* **33**, 1877–1901 (2020)
51. Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: *International Conference on Machine Learning*, pp. 4904–4916. PMLR, New York (2021)
52. OpenAI: Gpt-4v(ision) system card, tech. rep., OpenAI (2023)
53. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards General-Purpose Vision-Language Models with Instruction Tuning (2023)
54. Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Universal image-text representation learning. In: *European Conference on Computer Vision*, pp. 104–120. Springer, Berlin (2020)
55. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: bootstrapping language-image pre-training for unified vision-language understanding and generation. In: *International Conference on Machine Learning*, pp. 12888–12900. PMLR, New York (2022)
56. Sun, C., Myers, A., Vondrick, C., Murphy, K.P., Schmid, C.: Videobert: a joint model for video and language representation learning. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019)
57. Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: data collection and evaluation server. *arXiv preprint arXiv:1504.00325* (2015)
58. 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. Linguistics* **2**, 67–78 (2014)
59. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: elevating the role of image understanding in visual question answering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913 (2017)
60. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Ok-VQA: a visual question answering benchmark requiring external knowledge. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3195–3204 (2019)
61. Hudson, D.A., Manning, C.D.: GQA: a new dataset for real-world visual reasoning and compositional question answering. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709 (2019)
62. Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M., Parikh, D., Batra, D.: Visual dialog. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 326–335 (2017)
63. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: referring to objects in photographs of natural scenes. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 787–798 (2014)
64. Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: learning a text-video embedding by watching hundred million narrated video clips. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2630–2640 (2019)
65. Zhou, L., Xu, C., Corso, J.: Towards automatic learning of procedures from web instructional videos. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)
66. Zhang, B., Zhang, P., Dong, X., Zang, Y., Wang, J.: Long-clip: unlocking the long-text capability of clip. In: *European Conference on Computer Vision*, pp. 310–325. Springer, Berlin (2025)
67. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are RNNs: fast autoregressive transformers with linear attention. In: *International Conference on Machine Learning*, pp. 5156–5165. PMLR, New York (2020)

68. Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., Qiao, Y.: Clip-adapter: better vision-language models with feature adapters. *Int. J. Comput. Vis.* **132**(2), 581–595 (2024)
69. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825 (2022)
70. Shu, M., Nie, W., Huang, D.-A., Yu, Z., Goldstein, T., Anandkumar, A., Xiao, C.: Test-time prompt tuning for zero-shot generalization in vision-language models. *Adv. Neural Inf. Proces. Syst.* **35**, 14274–14289 (2022)
71. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929 (2016)
72. Diao, S., Huang, Z., Xu, R., Li, X., Lin, Y., Zhou, X., Zhang, T.: Black-box prompt learning for pre-trained language models. *arXiv preprint arXiv:2201.08531* (2022)
73. Yu, L., Chen, Q., Lin, J., He, L.: Black-box prompt tuning for vision-language model as a service. In: *IJCAI*, pp. 1686–1694 (2023)
74. Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J.D., Chen, D., Arora, S.: Fine-tuning language models with just forward passes. *Adv. Neural Inf. Proces. Syst.* **36**, 53038–53075 (2023)
75. Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., Han, S.: Smoothquant: accurate and efficient post-training quantization for large language models. In: *International Conference on Machine Learning*, pp. 38087–38099. PMLR, New York (2023)
76. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., Han, S.: AWQ: activation-aware weight quantization for on-device LLM compression and acceleration. In: *Proceedings of Machine Learning and Systems*, vol. 6, pp. 87–100 (2024)
77. Zou, J., Schiebinger, L.: Design ai so that it’s fair. *Nature* **559**(7714), 324–326 (2018)
78. Parashar, S., Lin, Z., Liu, T., Dong, X., Li, Y., Ramanan, D., Caverlee, J., Kong, S.: The neglected tails in vision-language models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12988–12997 (2024)
79. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: *International Conference on Machine Learning*, pp. 1597–1607. PMLR, New York (2020)
80. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI* 16, pp. 776–794. Springer, New York (2020)
81. Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., Hoi, S.C.H.: Align before fuse: vision and language representation learning with momentum distillation. *Adv. Neural Inf. Proces. Syst.* **34**, 9694–9705 (2021)
82. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. OpenAI (2018)
83. 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, New York (2023)
84. Li, C., Gan, Z., Yang, Z., Yang, J., Li, L., Wang, L., Gao, J., et al.: Multimodal foundation models: from specialists to general-purpose assistants. *Found. Trends® Comput. Graphics Vision* **16**(1–2), 1–214 (2024)
85. Zhang, J., Huang, J., Jin, S., Lu, S.: Vision-language models for vision tasks: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(8), 5625–5644 (2024)
86. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **55**(9), 1–35 (2023)

Part I
**Scaling Intelligence: Pre-Training
Strategies for Vision-Language Models**

Chapter 2

InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks



Wenhai Wang, Zhe Chen, Yangzhou Liu, Yue Cao, Weiyun Wang,
Xizhou Zhu, Lewei Lu, Tong Lu, Yu Qiao, and Jifeng Dai

Abstract The exponential growth of large language models (LLMs) has opened up numerous possibilities for multi-modal AGI systems. However, progress in vision and vision-language foundation models, critical for multi-modal AGI, lags behind LLMs. In this chapter, we discuss a large-scale vision-language foundation model (InternVL), which scales up the vision foundation model to 6 billion parameters and progressively aligns it with the LLM, using web-scale image-text data from various sources. This model demonstrates state-of-the-art performance across 32 generic visual-linguistic benchmarks, including visual perception tasks such as image-level and pixel-level recognition, vision-language tasks like zero-shot image/video classification and zero-shot image/video-text retrieval, and integration with LLMs to develop powerful multi-modal dialog systems, which closes the performance gap from open-sourced MLLMs to commercial multimodal models like GPT-4V.

W. Wang (✉)

Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China

e-mail: wangwenhai@pjlab.org.cn

Z. Chen · Y. Liu · Y. Cao · T. Lu

School of Computer Science, Nanjing University, Nanjing, China

e-mail: lutong@nju.edu.cn

W. Wang

School of Computer Science, Fudan University, Shanghai, China

e-mail: wangweiyun@pjlab.org.cn

X. Zhu · J. Dai

Department of Electronic Engineering, Tsinghua University, Beijing, China

e-mail: zhuwalter@sensetime.com; daijifeng@tsinghua.edu.cn

L. Lu

SenseTime Research, Hong Kong, China

e-mail: luotto@sensetime.com

Y. Qiao

Shanghai AI Laboratory, Shanghai, China

e-mail: yu.qiao@siat.ac.cn

Keywords Vision Foundation Model · Multimodal Large Language Model · Image–Text Alignment · Chatbot · Instruction Following

2.1 Introduction

Large language models (LLMs) largely promote the development of artificial general intelligence (AGI) systems with their impressive capabilities in open-world language tasks, and their model scale and performance are still increasing at a fast pace. Leveraging the advancements in LLMs, multi-modal large language models (MLLMs) [5, 27, 35, 88, 90, 113, 125, 154, 192] make significant strides, facilitating complex vision-language dialogs and interactions that bridge the gap between textual and visual information. However, progress in vision and vision-language foundation models, which are also crucial for MLLMs, lags behind the rapid growth of LLMs. There remains a noticeable gap between the capabilities of open-source models and proprietary commercial models, e.g., GPT-4V [116], Gemini series [125, 142].

This gap is mainly reflected in the following three aspects: (1) *Parameter Scale*: Recent proprietary commercial MLLMs [5, 116, 125, 137] typically scale not less than 100 billion parameters, while open-source models commonly employ a 300 million parameter vision foundation model (VFM), which is integrated with either a 7 billion or 13 billion LLMs. In addition, vision models trained on pure-vision data or aligned with the BERT series [57, 62, 94] often exhibit representation inconsistencies with LLMs. This may lead to the under-use of LLM’s capacity. (2) *Image Resolution*: Proprietary commercial models typically employ a dynamic resolution approach, preserving the original aspect ratio to facilitate detailed scene and document understanding. In contrast, open-source models generally train with fixed resolutions [27, 35, 88, 99, 156, 192], such as 336×336 and 448×448 , leading to a considerable gap in capabilities relative to commercial counterparts. (3) *Multilingual Capability*: Proprietary models often leverage extensive multilingual datasets for training, enhancing their performance across diverse languages. However, open-source models predominantly utilize English data, relying on the zero-shot capabilities of LLMs for other languages, e.g. LLaVA-NeXT [89]. This results in suboptimal performance in non-English scene understanding and OCR tasks.

To bridge this gap, *our inspiration lies in scaling up the vision encoder to match the parameter scale of the LLM and subsequently aligning their representations*. However, the training of such large-scale models requires a vast amount of image–text data obtained from the Internet. The significant heterogeneity and quality variations within this data pose considerable challenges to the training process. To enhance the efficiency of the training, generative supervision is considered as a complementary approach to contrastive learning, as depicted in Fig. 2.1. Contrastive learning allows models to extract knowledge from large-scale noisy data, whereas

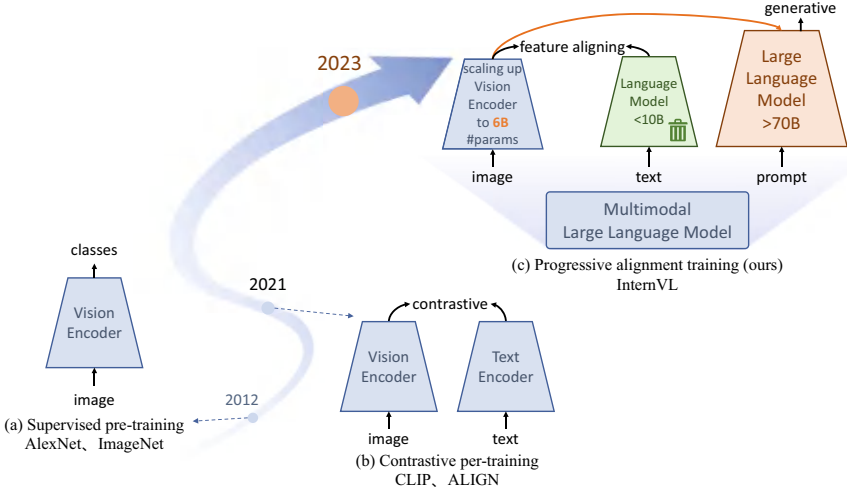


Fig. 2.1 Comparisons of different vision and vision-language foundation models. (a) indicates the traditional vision foundation model, e.g. ResNet [48] pre-trained on classification tasks. (b) represents the vision-language foundation models, e.g. CLIP [123] pre-trained on image-text pairs. (c) is our InternVL, which presents a workable way to align the large-scale vision foundation model (i.e., InternViT-6B) with the large language model and is versatile for both contrastive and generative tasks

generative supervision endows the model with creativity, enabling it to generate responses based on user instructions, which require high-quality training data.

Based on this point, we discuss *InternVL*, an open-source multimodal large language model, which aligns the representation of the scaled-up vision encoder with the LLM and achieves state-of-the-art performance on various visual and vision-language tasks. As shown in Fig. 2.1c, InternVL has two key designs: (1) *Large-scale VFM*: it includes a vision encoder scaled to 6 billion parameters, which provides a strong and generic representation that aligns with LLMs, and can be used across various visual tasks and domains. (2) *Progressive Image-Text Alignment*: it can facilitate the full alignment of the two large-scale components, VFM and LLM, across heterogeneous datasets with varying scales and qualities. Additionally, it enables the gradual expansion of the model from 10 billion to 100 billion parameters, maximizing the utilization of network noisy data.

In addition, to bridge the performance gap with large commercial multimodal models, we discuss InternVL-Chat, a model specifically for multimodal conversations. (1) *Simple “VFM-MLP-LLM” Architecture*: It uses a lightweight MLP projector to connect VFM and LLM, effectively combining the strengths of each component in a compact and efficient structure. (2) *Dynamic High-Resolution*: It segments the image into 448×448 tiles, with the number of tiles ranging from 1 to 40 (i.e., 4K resolution) based on the aspect ratio and resolution of the images. Additionally, a thumbnail view is included to capture the global context. (3) *High-*

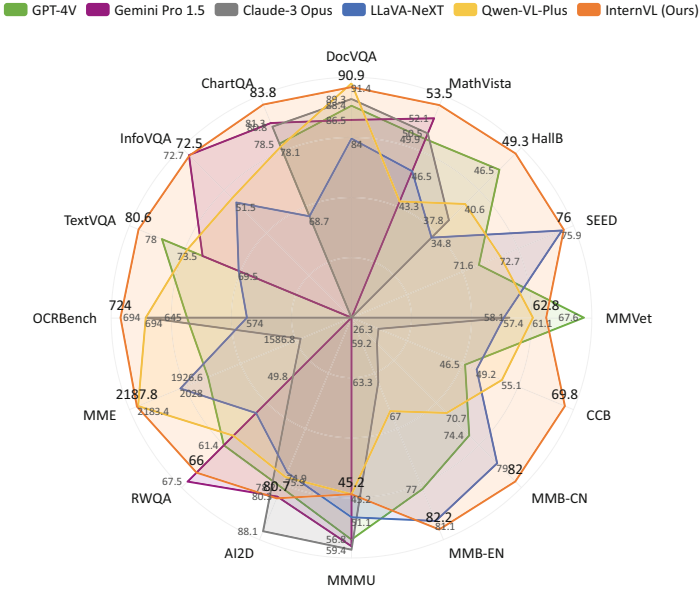


Fig. 2.2 Comparison with SoTA models on 16 multi-modal benchmarks. These benchmarks are categorized into four specific groups: OCR-related, general multimodal, mathematical, and multi-turn conversation benchmarks. The results of these benchmarks show that InternVL-Chat achieves performance comparable to leading proprietary models

quality Bilingual Dataset: A diverse collection of public datasets is gathered, covering high-quality natural scenes, charts, documents, and conversations in both English and Chinese. Furthermore, a data translation pipeline is developed using open-source LLMs, which can be easily extended to more languages.

The experimental results of InternVL and InternVL-Chat across a wide range of datasets are summarized as follows. Comparison with state-of-the-art visual foundation models demonstrates that InternVL achieves leading performance on a broad range of vision and vision-language tasks, including image classification (ImageNet), semantic segmentation (ADE20K), video classification (Kinetics), image-text retrieval (Flickr30K & COCO), and image captioning (COCO & Flickr & Nocaps). Meanwhile, InternVL-Chat performs competitively compared to multi-modal open source and proprietary models, achieving state-of-the-art results in 8 of 18 benchmarks. These benchmarks are categorized into four specific groups: OCR-related, general multimodal, mathematical, and multi-turn conversation benchmarks. Notably, as shown in Fig. 2.2, InternVL-Chat even surpasses leading proprietary models like GPT-4V [116], Claude-3 Opus [3], and Gemini Pro 1.5 [125] in four specific benchmarks, particularly in OCR-related datasets such as TextVQA [134]. These results indicate that InternVL-Chat has effectively narrowed the gap between open-source models and leading commercial models.

2.2 Related Work

2.2.1 Vision Foundation Models

Vision foundation models (VFMs) have undergone significant evolution. From the pioneering AlexNet [65] to advanced CNN architectures like ResNet [48], DenseNet [55], and InternImage [155], breakthroughs such as residual connections [48] enabled deeper and more scalable networks. The advent of Vision Transformers (ViT)[37] revolutionized the field, with models like Swin Transformer[98] and PVT [159, 160] excelling in diverse visual tasks. The VFM becomes an integral part of the multimodal system, connected to the LLM through a lightweight “glue” layers [75, 90, 192]. However, a gap exists as these models primarily derive from visual-only datasets like ImageNet [34] or JFT [183], or are aligned with the BERT series [57, 62, 94] using image–text pairs, lacking direct alignment with LLMs. Additionally, models like CLIP-ViT [123] are typically limited to 1 billion parameters[40, 56], constraining their capacity in vision-language tasks.

To address these challenges, researchers have explored hybrid and dual-encoder approaches. Tong et al. [146] combined features from CLIP and DINOv2 [117] to leverage their complementary visual patterns. LLaVA-HR [104] employed a dual-branch encoder, integrating CLIP-ViT for low-resolution and CLIP-ConvNext for high-resolution inputs. Similarly, DeepSeek-VL [99] used SigLIP-L for low-resolution and SAM-B for high-resolution images. These advancements emphasize the drive toward VFMs that align better with LLMs, enabling more robust multi-modal applications.

2.2.2 Large Language Models

Large language models (LLMs) have revolutionized the field of artificial intelligence, enabling natural language processing tasks that were previously thought exclusive to humans [1, 148, 164]. The emergence of GPT-3 [164] brought a significant leap in capabilities, particularly in few-shot and zero-shot learning, highlighting the immense potential of LLMs. This promise was further realized with the advancements of ChatGPT and GPT-4 [1]. The progress in the field has been further accelerated by the emergence of open-source LLMs, including the LLaMA series [39, 148, 149], Vicuna [190], InternLM series [15, 143], MOSS [139], ChatGLM [38], Qwen series [4, 171], Baichuan [7], and Falcon [120], among others [31, 141, 165]. However, in real scenarios, interactions are not limited to natural language. The vision modality can bring additional information, which means more possibilities. Therefore, exploring how to utilize the excellent capabilities of LLMs for multi-modal interactions is poised to become the next research trend.

2.2.3 *Multimodal Large Language Models*

Multimodal Large language models (MLLMs) [1, 4, 7, 11, 15, 139, 141, 143, 148, 149, 164, 190] have greatly advanced AGI by enabling complex language tasks previously thought human-exclusive. Building on this, the development of proprietary commercial MLLMs represents a significant evolution. For example, OpenAI’s GPT-4V [116] extends GPT-4’s capabilities by incorporating visual inputs, allowing it to handle both text and image content, which stands as a significant development in the domain of MLLMs. Afterward, Google’s Gemini series progresses from Gemini 1.0 [142] to Gemini 1.5 [125], enhancing MLLMs with the ability to process text, images, and audio and support up to 1 million tokens, which boosts performance significantly. Other notable proprietary MLLMs include Anthropic’s Claude-3V series [3], HyperGAI’s HPT Pro [54], Apple’s MM1 [113], StepFun’s Step-1V [137], and xAI’s Grok-1.5V [167].

The rise of open-source MLLMs [2, 72, 76, 80, 81, 96, 97, 145, 157, 162, 166, 186, 188] has also greatly influenced the AGI landscape by integrating and enhancing capabilities in processing both visual and textual data. Many open-source multimodal large language models (MLLMs) have gained prominence, including the LLaVA series [88–90], MiniGPT-4 [192], VisionLLM [154], Qwen-VL [5], and others [27, 35, 121, 158]. However, these models are typically trained on images with small, fixed resolutions (e.g., 336×336 or 448×448), which leads to suboptimal performance on images with unusual aspect ratios or document data. To address the issue of training on high-resolution images, two common approaches have been explored: one involves designing dual-branch image encoders [51, 78, 104, 105, 163], and the other involves dividing the image into many low-resolution tiles [36, 52, 71, 83, 89, 95, 169, 172]. Despite these explorations in high-resolution training, these open-source models still exhibit significant gaps in understanding documents, charts, and infographics, as well as recognizing scene texts, compared to leading commercial models.

2.3 *Proposed Method*

2.3.1 *Overall Architecture*

As shown in Fig. 2.3, unlike traditional vision-only backbones [48, 98, 155] and dual-encoder models [56, 123, 138], InternVL is designed with a large-scale vision encoder, InternViT-6B, coupled with a language middleware. Specifically, InternViT-6B is a vision transformer with 6 billion parameters, customized to achieve a favorable trade-off between performance, efficiency, and stability. The language middleware is initially set up as the multilingual-enhanced LLaMA [31]. This middleware is first used for image–text contrastive pre-training, and after image–text alignment, it will gradually be replaced with larger LLMs (e.g.,

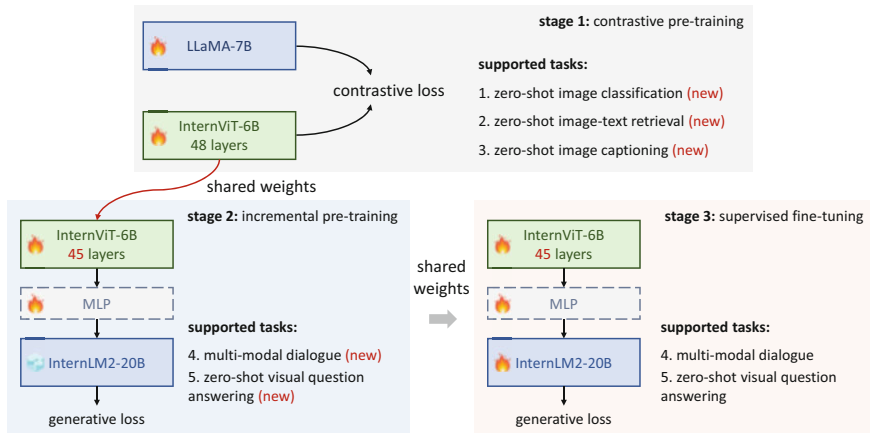


Fig. 2.3 The training strategy of the proposed InternVL model. It consists of three progressive stages, including vision-language contrastive training, vision-language generative training, and supervised fine-tuning. These stages effectively leverage public data from diverse sources, ranging from noisy image-text pairs on the web to high-quality caption, VQA, and multi-modal dialog datasets

LLaMA or its variants [148, 149, 190], InternLM series [15, 143], and Qwen series [4, 171]. This middleware can function both as a language encoder, providing strong multilingual representations for image-text contrastive learning, and as a text decoder, generating responses based on user queries.

Large-Scale Vision Encoder We implement the vision encoder of InternVL with vanilla vision transformer (ViT) [37]. To match the scale of LLMs, the vision encoder is scaled up to 6 billion parameters, resulting in the InternViT-6B model. To obtain a good trade-off between accuracy, speed, and stability, we conduct a hyper-parameter search for InternViT-6B. We vary the model depth within {32, 48, 64, 80}, the head dimension within {64, 128}, and the MLP ratio within {4, 8}. The model width and the head number are calculated based on the given model scale and other hyper-parameters.

We employ contrastive learning on a 100M subset of the LAION-en dataset [126] to measure the accuracy, speed, and stability of InternViT-6B variants with different configurations. We report the following findings: (1) *Speed*. For different model settings, when computation is not saturated, the models with smaller depths exhibit faster speed per image. However, as the GPU computation is fully utilized, the speed difference becomes negligible; (2) *Accuracy*. With the same number of parameters, the depth, head dimension, and MLP ratio have little impact on the performance. Based on these findings, the most stable configuration for the final model is identified, as shown in Table 2.1.

Language Middleware The language middleware is used to align visual and language representations. As shown in Fig. 2.3, it plays different roles in various

Table 2.1 Architecture details of the InternViT-6B model

Name	Width	Depth	MLP	#heads	#param (M)
ViT-G [183]	1664	48	8192	16	1843
ViT-e [25]	1792	56	15360	16	3926
EVA-02-ViT-E [138]	1792	64	15360	16	4400
ViT-6.5B [136]	4096	32	16384	32	6440
ViT-22B [33]	6144	48	24576	48	21743
InternViT-6B (ours)	3200	48	12800	25	5903

training stages. In the image–text alignment pre-training stage, the language middleware is initialized as the pre-trained multilingual LLaMA-7B [31]. During this stage, the language middleware provides a holistic text representation to compute the similarity with the image representation provided by the visual encoder, thereby facilitating contrastive learning. In the subsequent generative training stages, since the representation of the visual encoder is aligned with the LLM in the prior stage, we can leverage a larger and more powerful LLM as the language middleware to enhance the model’s performance in multi-modal dialogue tasks.

The core idea underlying the language middleware is *the consistent representations across different LLMs*. Despite the varying designs of different LLMs, they are all trained on text corpora collected from the entire Internet, which gives their representation spaces homogeneity. A vision encoder only needs to align with one of these LLMs. In subsequent training, even if it switches to another LLM, only a small amount of sample data is needed for incremental pre-training or supervised fine-tuning (SFT) to achieve alignment.

“Swiss Army Knife” Model: InternVL By flexibly combining the vision encoder and the language middleware, InternVL can support various vision or vision-language tasks.

- (1) *For visual perception tasks*, the vision encoder of InternVL, i.e. InternViT-6B, can be used as the backbone for vision tasks. Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, our model can generate a feature map $F \in \mathbb{R}^{H/14 \times W/14 \times D}$ for dense prediction tasks, or work with global average pooling and linear projection to make image classification.
- (2) *For image-text contrastive tasks*, we first apply attention pooling to the image features of InternViT-6B, to derive the global visual feature I_f . Next, we encode text as T_f by extracting the feature from the [EOS] token of the language middleware. By computing similarity scores between I_f and T_f , we support various contrastive tasks such as image–text retrieval.
- (3) *For text generative tasks*, we introduce InternVL-Chat by replacing the language middleware from LLaMA-7B with larger LLMs. Specifically, we follow a “VFT-MLP-LLM” architecture similar to LLaVA. Using a lightweight MLP projector, we connect InternViT-6B with different LLMs (e.g., LLaMA or its variants [148, 149, 190], InternLM series [15, 143], and Qwen series [4, 171]),

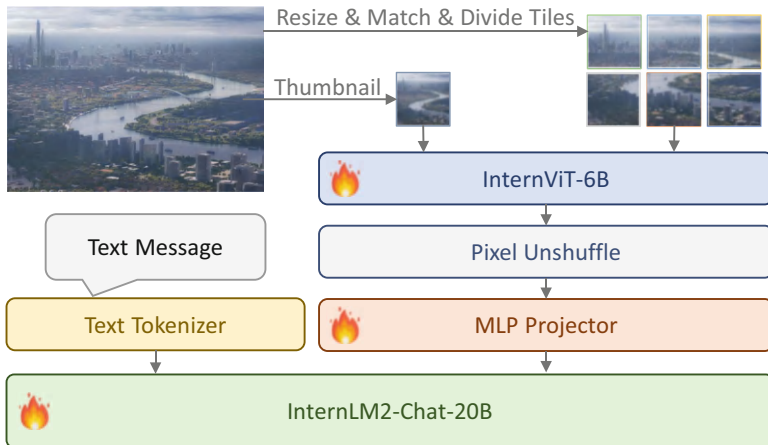


Fig. 2.4 Overall architecture of InternVL-Chat. InternVL-Chat adopts the “VFM-MLP-LLM” architecture similar to popular MLLMs [88, 89], combining a pre-trained InternViT-6B [27] with InternLM2-20B [15] through a MLP projector. Here, we employ a simple pixel shuffle to reduce the number of visual tokens to one-quarter

as shown in Fig. 2.4. Complemented by a dynamic resolution strategy and high-quality incremental pre-training and SFT training sets, InternVL-Chat achieves performance comparable to GPT-4V [116] on multiple key multi-modal benchmarks.

2.3.2 Progressive Image–Text Alignment

To align the two large-scale components with substantial gaps in modalities and structures, we introduce a progressive alignment training strategy. The training strategy is conducted progressively, beginning with contrastive learning on large-scale noisy data, gradually moving toward generative learning on high-quality data. In this way, we ensure the effective organization and full utilization of web-scale image–text data from a variety of sources. As shown in Fig. 2.3, the training of InternVL is divided into three progressive stages: image–text contrastive training, incremental pre-training, and supervised fine-tuning (SFT). The three stages effectively leverage public data from diverse sources, ranging from noisy image-text pairs on the web to high-quality caption, VQA, and multi-modal dialog datasets.

Image–Text Contrastive Training In the first stage, we conduct contrastive learning to align InternViT-6B with a multilingual LLaMA-7B [31] on web-scale, noisy image–text pairs. During training, we adopt the LLaMA to encode the text as T_f and

use InternViT-6B to extract the image feature I_f . Following the objective function of CLIP [123], we minimize a symmetric cross-entropy loss on the similarity scores of image–text pairs in a batch. This stage allows InternVL to excel on contrastive tasks like zero-shot image classification and image–text retrieval, and the vision encoder of this stage can also perform well on visual perception tasks.

In this stage, the image encoder, InternViT-6B, is randomly initialized [8], while the text encoder, LLaMA-7B, is initialized with pre-trained weights from [31]. All model parameters are fully trainable, allowing for fine-tuning of both the visual and textual components to optimize multimodal alignment.

Incremental Pre-Training In the second stage of training, to enable seamless multimodal integration, we drop the LLaMA in stage 1, and connect the InternViT-6B vision encoder with a larger LLM (e.g., InternLM2-20B [15]) via MLP. During training, the LLM is frozen, and InternViT-6B and MLP are fine-tuned. To support high-resolution image input, we introduce a dynamic high-resolution strategy, where the training images are divided into 448×448 pixel tiles, with up to 12 tiles based on the input resolution and aspect ratio. This setup enables the model to scale up to 40 tiles (i.e., 4K resolution) in a zero-shot manner during testing. This alignment strategy ensures robust multi-modal capabilities.

The training dataset in this stage is expanded to include more diverse detection and large-scale OCR datasets, surpassing the dataset used in the previous image-text contrastive training stage. The focus during this stage is on optimizing the InternViT-6B vision encoder and the MLP projector to improve visual feature extraction. This enhanced training setup ensures that the model can effectively extract and align both visual and textual features, enabling robust performance across various multi-modal tasks.

Supervised Fine-Tuning During the supervised fine-tuning (SFT) stage, the entire InternVL-Chat model’s parameters are fine-tuned to enhance multi-modal capabilities. we meticulously select datasets to enhance model performance across a wide range of multi-modal tasks in this stage. The SFT datasets encompass various domains, including image captioning, general question answering, scientific charts, mathematics, knowledge, document understanding, and more. This expansion ensures the model becomes more versatile, effectively handling tasks that require a deep understanding of both visual and textual data.

2.3.3 Details of InternVL-Chat

Model Design In most multi-modal large models (MLLMs), as described in [5, 35, 88–90, 106, 192], the most commonly used visual foundation models are typically contrastively pre-trained ViTs (e.g., openclip, chen2023internvl, radford2021clip, zhai2023siglip). However, these ViTs are usually trained on fixed low-resolution image–text pairs (i.e., 224×224) scraped from the internet, which leads to perfor-

mance degradation when processing high-resolution images or images from sources other than the internet, such as document images.

Our InternViT-6B does not have this problem. In the second and third stages, the model increases the resolution from a fixed 224×224 to a dynamic 448×448 , with the basic block size being 448×448 , and the number of blocks ranging from 1 to 12. It is fully trained using a mixed dataset of image captioning [14, 24, 121, 126, 134] and OCR-specific datasets [44, 127].

Additionally, we found that the features from the fourth to the fourth-last layers perform best in multimodal tasks, so we directly discard the weights of the last three layers, reducing InternViT-6B from 48 layers to 45 layers. As shown in Fig. 2.3, we connect the 45 layers of InternViT-6B to the LLM through an MLP. To reduce the excessive image tokens caused by high resolution, we use a pixel-unshuffle operation to downsample both the width and height by a factor of 2.

Dynamic High-Resolution Inspired by UReader [172], we adapt a dynamic high-resolution training approach that effectively adapts to the varying resolutions and aspect ratios of input images. This method leverages the flexibility of segmenting images into tiles, enhancing the model’s ability to process detailed visual information while accommodating diverse image resolutions. It mainly consists of the following steps:

- (1) *Dynamic Aspect Ratio Matching*. As shown in Fig. 2.5, to maintain natural aspect ratios during processing, we dynamically match the optimal aspect ratio from a pre-defined set of aspect ratios. Due to limited computational resources, we allow a maximum of 12 tiles during training. Consequently, this set includes all 35 possible combinations of aspect ratios formed by 1 to 12 tiles, such as



Fig. 2.5 Illustration of dynamic high resolution. We dynamically match an optimal aspect ratio from pre-defined ratios, dividing the image into tiles of 448×448 pixels and creating a thumbnail for global context. This method minimizes aspect ratio distortion and accommodates varying resolutions during training

Table 2.2 Details of the training data for InternVL in the pre-training stage. Among them, LAION-en [126], LAION-multi [126], COYO [14], and Wukong [44] are web-scale image-text pairs data. LAION-COCO [127] is a synthetic dataset with high-quality captions from LAION-en. CC12M [21], CC3M [131], SBU [118] are academic caption datasets. “Multi” means multilingual

Dataset	Characteristics		Cleaned	Remain
	Language	Original		
LAION-en [126]	English	2.3B	1.94B	84.3%
LAION-COCO [127]		663M	550M	83.0%
COYO [14]		747M	535M	71.6%
CC12M [21]		12.4M	11.1M	89.5%
CC3M [131]		3.0M	2.6M	86.7%
SBU [118]		1.0M	1.0M	100%
Wukong [44]	Chinese	100M	69.4M	69.4%
LAION-multi [126]	Multi	2.2B	1.87B	85.0%
Total	Multi	6.03B	4.98B	82.6%

{1:1, 1:2, 2:1, 3:1, ..., 2:6}. During the matching process, for each input image, we calculate its aspect ratio and compare it with the 35 pre-defined aspect ratios by measuring the absolute difference. If multiple predefined aspect ratios match (e.g., 1:1 and 2:2), we prioritize the one not exceeding twice the input image’s area, thereby preventing excessive enlargement of low-resolution images.

- (2) *Image Division & Thumbnail*. Once an appropriate aspect ratio is determined, the image is resized to the corresponding resolution. For example, an 800×1300 image will be resized to 896×1344 . The resized image is then divided into tiles of 448×448 pixels. Alongside the tiles, we include a thumbnail of the entire image to capture the global context. This thumbnail is scaled down to 448×448 , aiding the model in understanding the overall scene. Therefore, during training, the number of visual tokens ranges from 256 to 3,328. During testing, the number of tiles can increase to a maximum of 40, resulting in 10,496 visual tokens.

2.3.4 Data Preparation

Stage 1: Pre-training Dataset The data used in Stage 1 for image-text contrastive training are publicly available and consist of multilingual content, including datasets such as LAION-en [126], LAION-multi [126], LAION-COCO [127], COYO [14], Wukong [44], and others. We combine these datasets and apply a data filtering process to remove extremely low-quality entries. As summarized in Table 2.2, the original dataset contains 6.03 billion image–text pairs, with 4.98 billion remaining after cleaning. During this stage, we apply only minimal data filtering, ensuring that the majority of the data was retained. We consider six factors for filtering: CLIP

Table 2.3 Summary of Datasets used in the incremental pre-training stage of InternVL. To construct large-scale OCR datasets, we utilized PaddleOCR [73] to perform OCR in Chinese on images from Wukong [44] and in English on images from LAION-COCO [127]

Task	Ratio	Dataset
Captioning	53.9%	Laion-EN (en) [126], Laion-ZH (zh) [126], COYO (zh) [14], GRIT (zh) [121], COCO (en) [24], TextCaps (en) [133]
Detection	5.2%	Objects365 (en&zh) [130], GRIT (en&zh) [121], All-Seeing (en&zh) [158]
OCR (large)	32.0%	Wukong-OCR (zh) [44], LaionCOCO-OCR (en) [127], Common Crawl PDF (en&zh)
OCR (small)	8.9%	MMC-Inst (en) [87], LSVT (zh) [140], ST-VQA (en) [13] RCTW-17 (zh) [132], ReCTs (zh) [187], ArT (en&zh) [28], SynthDoG (en&zh) [63], COCO-Text (en) [150], ChartQA (en) [110], CTW (zh) [181], DocVQA (en) [112], TextOCR (en) [135], PlotQA (en) [114], InfoVQA (en) [111]

similarity, watermark probability, unsafe content probability, aesthetic score, image resolution, and caption length. This helps remove extreme data points and maintain training stability. Additionally, we exclude data duplicated from ImageNet-1K/22K [34], Flickr30K [122], and COCO [82] to ensure the reliability of our zero-shot evaluations. Due to download failures and the use of our filtering pipeline, the total amount of data retained in Stage 1 is 4.98 billion image–text pairs.

Stage 2: Incremental Pre-training Dataset The incremental pre-training dataset utilized in our InternVL-Chat encompasses a diverse range of publicly accessible sources. We provide an overview of these datasets in Table 2.3. These datasets span multiple tasks, including captioning, which predominantly uses datasets such as Laion-EN [126], Laion-ZH [126], COYO [14], and GRIT [121], constituting 53.9% of the total data. Detection and grounding tasks utilize datasets like Objects365 [130], GRIT [121], and All-Seeing [158], making up 5.2%. For OCR tasks, we utilize large-scale datasets such as Wukong-OCR, LaionCOCO-OCR, and Common Crawl PDFs, which constitute 32.0% of our data. These datasets are constructed using PaddleOCR [73] to perform OCR on Chinese images from Wukong [44] and on English images from LaionCOCO [127]. Smaller OCR datasets include MMC-Inst [87], LSVT [140], ST-VQA [13], RCTW-17 [132], ArT [28], and others, accounting for 8.9% of the data, which focus on more specific or constrained OCR challenges. This diverse dataset assembly ensures robust model pre-training of InternVL-Chat, catering to varied linguistic and visual elements across tasks.

Stage 3: Supervised Fine-tuning Dataset During the SFT stage, we meticulously select datasets to enhance model performance across a wide range of multi-modal tasks. The datasets used in this phase are summarized in Table 2.4.

For image captioning, we include TextCaps [133] and bilingual ShareGPT4V [23], which help the model learn to generate descriptive captions in both English and

Table 2.4 Summary of Datasets used in the supervised fine-tuning stage of InternVL

Task	Dataset
Captioning	TextCaps (en) [133], ShareGPT4V (en&zh) [23]
General QA	VQAv2 (en) [43], GQA (en) [53], OKVQA (en) [109], VSR (en) [85], VisualDialog (en) [32]
Science	AI2D (en) [60], ScienceQA (en) [102], TQA (en) [61]
Chart	ChartQA (en) [110], MMC-Inst (en) [87], DVQA (en) [59], PlotQA (en) [114], LRV-Instruction (en) [86]
Mathematics	GeoQA+ (en) [16], TabMWP (en) [103], MathQA (en) [177], CLEVR-Math/Super (en) [79, 84], Geometry3K (en) [101]
Knowledge	KVQA (en) [129], A-OKVQA (en) [128], ViQuAE (en) [69], Wikipedia (en&zh) [46]
OCR	OCRVQA (en) [115], InfoVQA (en) [111], TextVQA (en) [134], ArT (en&zh) [28], COCO-Text (en) [150], CTW (zh) [181], LSVT (zh) [140], RCTW-17 (zh) [132], ReCTs (zh) [187], SynthDoG (en&zh) [63], ST-VQA (en) [13]
Document	DocVQA (en) [29], Common Crawl PDF (en&zh)
Grounding	RefCOCO/+g (en) [108, 178], Visual Genome (en) [64]
Conversation	LLaVA-150K (en&zh) [90], LVIS-Instruct4V (en) [152], ALLaVA (en&zh) [22], Laion-GPT4V (en) [66], TextOCR-GPT4V (en) [58], SVIT (en&zh) [189]
Text-only	OpenHermes2.5 (en) [144], Alpaca-GPT4 (en) [141], ShareGPT (en&zh) [190], COIG-CQIA (zh) [6]

Chinese. In the domain of general QA, datasets such as VQAv2 [43], GQA [53], and VisualDialog [32] teach the model to handle diverse question-answering scenarios.

For scientific image understanding, datasets like AI2D [60], ScienceQA [102], and TQA [61] provide content-rich scenarios to enhance the model’s ability to interpret scientific diagrams and texts. Chart interpretation is bolstered by ChartQA [110], MMC-Inst [87], and PlotQA [114], which train the model to analyze and understand chart images. Mathematics datasets such as GeoQA+ [16], TabMWP [103], and MathQA [177] introduce complex numerical and geometric problem-solving tasks. Knowledge-based QA benefits from the inclusion of datasets like KVQA [129] and bilingual Wikipedia [46], enabling the model to extract and reason with factual information across multiple languages.

For tasks involving OCR, we utilize OCRVQA [115], TextVQA [134], and several datasets focused on Chinese and English text recognition, such as SynthDoG [63], to improve text recognition from images. Document understanding is advanced through datasets like DocVQA [112] and Common Crawl PDFs, which help the model for real-world document analysis. Visual grounding is trained using RefCOCO [108, 178] and Visual Genome [64], aiding the model in precise object localization within images. In the realm of multimodal conversation, datasets like LLaVA-150K [90] and ALLaVA [22] enhance the model’s dialogic capabilities

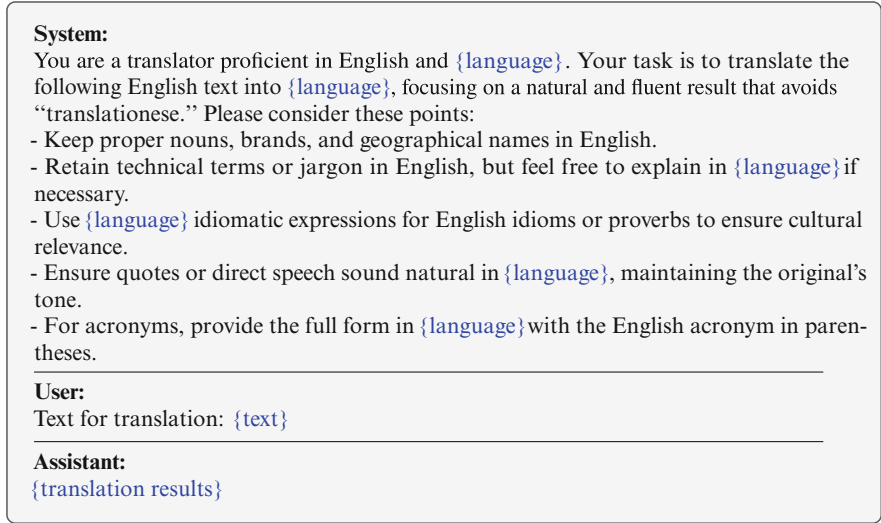


Fig. 2.6 Explanation of data translation pipeline. Based on this prompt, we translate English data into Chinese while keeping the language natural and smooth. Here, {language} represents the target language, {text} refers to the original English text, and {translation results} indicates the translated text

by simulating interactive and engaging scenarios. Lastly, text-only datasets include OpenHermes2.5 [144], Alpaca-GPT4 [141], among others [6, 190], which are used to maintain the original linguistic capabilities of the LLM.

In summary, these datasets together establish a rich and diverse foundation for fine-tuning, which enhances our model’s ability to handle a wide range of multi-modal tasks and ensures its readiness for practical applications.

Data Translation Pipeline As shown in Fig. 2.6, to enhance our model’s multilingual capabilities, we implemented a data translation pipeline. This pipeline utilizes state-of-the-art open-source LLMs [4, 15, 175] or GPT-3.5 to convert English datasets to another language (e.g., Chinese), maintaining consistency and precision in bilingual labeling. Moreover, it can readily expand to encompass more languages by adjusting the language prompt, without relying on manual annotation processes.

In Tables 2.3 and 2.4, we have annotated the language for each dataset. For a dataset that was originally in English, an annotation as “zh” indicates that we have translated it into Chinese using the translation pipeline. For example, COYO [14] and GRIT [121] were originally English datasets, and we have translated them into Chinese. By leveraging this translation pipeline, the Chinese capabilities of InternVL-Chat have been greatly enhanced.

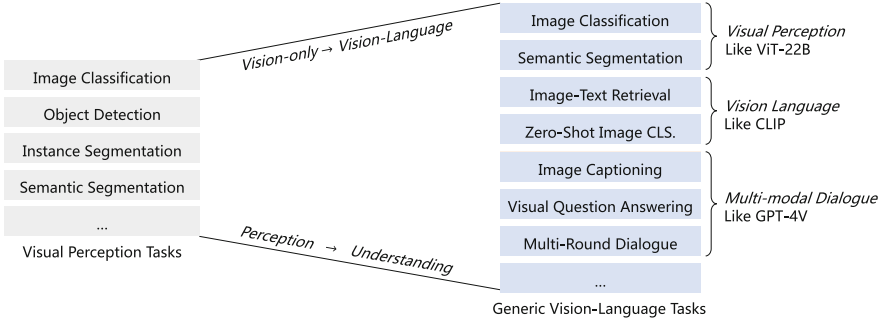


Fig. 2.7 Capability of InternVL. By combining the vision encoder and language middleware, InternVL supports both visual perception and vision-language understanding tasks

2.4 Experiments

2.4.1 Evaluation Settings

As shown in Fig. 2.7, our InternVL model supports a wide range of vision and vision-language tasks. (1) *For visual perception tasks*, we evaluate the visual capabilities of InternViT-6B, the most core component of InternVL. We use image classification [10, 34, 49, 50, 124, 151] and semantic segmentation [191] tasks. (2) *For image-text contrastive tasks*, we assess the performance of InternViT-6B combined with the language middleware LLaMA-7B through zero-shot image classification [9, 34, 49, 50, 124, 151], zero-shot video classification [18–20], and zero-shot image-text retrieval [24, 68, 77, 122]. (3) *For text generative tasks*, we conduct an extensive assessment across various benchmarks to measure InternVL-Chat’s multi-modal understanding and reasoning abilities. The benchmarks are categorized into four types: OCR-related [93, 110–112, 134], general multimodal [41, 45, 60, 70, 92, 167, 174, 179, 182], mathematical [100], and multi-turn conversation tasks [91]. The multi-modal dialog dataset is evaluated using the VLMEvalKit [30].

2.4.2 Visual Perception Benchmarks

First of all, we validate the visual perception capabilities of InternViT-6B, the most core component of InternVL.

Transfer to Image Classification We evaluate the quality of visual representation produced by InternViT-6B using the ImageNet-1K [34] dataset. Following common practices [33, 47, 117], we adopt the linear probing evaluation, i.e. training a linear classifier while keeping the backbone frozen. In addition to the ImageNet-1K validation set, we also report performance metrics on several ImageNet variants [10,

Table 2.5 Linear evaluation on image classification. We report the top-1 accuracy on ImageNet-1K [34] and its variants [10, 49, 50, 124, 151]. *ViT-22B [33] uses the private JFT-3B dataset [183]

Method	#param	IN-1K	IN-ReaL	IN-V2	IN-A	IN-R	IN-Ske	Avg.
OpenCLIP-H [56]	0.6B	84.4	88.4	75.5	—	—	—	—
OpenCLIP-G [56]	1.8B	86.2	89.4	77.2	63.8	87.8	66.4	78.5
DINOv2-g [117]	1.1B	86.5	89.6	78.4	75.9	78.8	62.5	78.6
EVA-01-CLIP-g [40]	1.1B	86.5	89.3	77.4	70.5	87.7	63.1	79.1
MAWS-ViT-6.5B [136]	6.5B	87.8	—	—	—	—	—	—
ViT-22B* [33]	21.7B	89.5	90.9	83.2	83.8	87.4	—	—
InternViT-6B (ours)	5.9B	88.2	90.4	79.9	77.5	89.8	69.1	82.5

Table 2.6 Semantic segmentation on ADE20K. Few-shot semantic segmentation with limited training data. Following ViT-22B [33], we fine-tune the InternViT-6B with a linear classifier

Method	#param	Crop size	1/16	1/8	1/4	1/2	1
ViT-L [147]	0.3B	504 ²	36.1	41.3	45.6	48.4	51.9
ViT-G [183]	1.8B	504 ²	42.4	47.0	50.2	52.4	55.6
ViT-22B [33]	21.7B	504 ²	44.7	47.2	50.6	52.5	54.9
InternViT-6B (ours)	5.9B	504 ²	46.5	50.0	53.3	55.8	57.2

Table 2.7 Semantic segmentation on ADE20K. Semantic segmentation performance in three different settings, from top to bottom: linear probing, head tuning, and full-parameter tuning

Method	Decoder	#param (train/total)	Crop size	mIoU
OpenCLIP-G _{frozen} [56]	Linear	0.3M / 1.8B	512 ²	39.3
ViT-22B _{frozen} [33]	Linear	0.9M / 21.7B	504 ²	34.6
InternViT-6B _{frozen} (ours)	Linear	0.5M / 5.9B	504 ²	47.2
ViT-22B _{frozen} [33]	UperNet	0.8B / 22.5B	504 ²	52.7
InternViT-6B _{frozen} (ours)	UperNet	0.4B / 6.3B	504 ²	54.9
ViT-22B [33]	UperNet	22.5B / 22.5B	504 ²	55.3
InternViT-6B (ours)	UperNet	6.3B / 6.3B	504 ²	58.9

49, 50, 124, 151], to benchmark the domain generalization capability. As shown in Table 2.5, InternViT-6B achieves a very significant improvement on linear probing.

Transfer to Semantic Segmentation To investigate the pixel-level perceptual capacity of InternViT-6B, we conduct extensive experiments of semantic segmentation on the ADE20K [191] dataset. Following ViT-22B [33], we begin with few-shot learning experiments, i.e. fine-tuning the backbone with a linear head on a limited dataset. As indicated in Table 2.6, InternViT-6B consistently outperforms ViT-22B across five experiments with varying proportions of training data. Additionally, Table 2.7 presents our further verification in three distinct settings, including linear probing, head tuning, and full-parameter tuning. Notably, in the case of linear probing, InternViT-6B attains 46.5 mIoU, a substantial +11.9 mIoU improvement over ViT-22B. These results underscore the strong out-of-the-box pixel-level perceptual capacity of our InternViT-6B.

Table 2.8 Comparison of zero-shot image classification performance of ImageNet variants [34, 49, 50, 124, 151] and ObjectNet [9]. “ $\Delta \downarrow$ ”: The gap between the averaged top-1 accuracy and the IN-1K top-1 accuracy. *CoCa [176] and LiT-22B [33] use the private JFT-3B dataset [183] during training

Method	IN-1K	IN-A	IN-R	IN-V2	IN-Sketch	ObjectNet	$\Delta \downarrow$	avg.
OpenCLIP-g [56]	78.5	60.8	90.2	71.7	67.5	69.2	5.5	73.0
OpenAI CLIP-L+ [123]	76.6	77.5	89.0	70.9	61.0	72.0	2.1	74.5
EVA-01-CLIP-g [138]	78.5	73.6	92.5	71.5	67.3	72.3	2.5	76.0
OpenCLIP-G [56]	80.1	69.3	92.1	73.6	68.9	73.0	3.9	76.2
EVA-01-CLIP-g+ [138]	79.3	74.1	92.5	72.1	68.1	75.3	2.4	76.9
MAWS-ViT-2B [136]	81.9	—	—	—	—	—	—	—
EVA-02-CLIP-E+ [138]	82.0	82.1	94.5	75.7	71.6	79.6	1.1	80.9
CoCa* [176]	86.3	90.2	96.5	80.7	77.6	82.7	0.6	85.7
LiT-22B* [33, 184]	85.9	90.1	96.0	80.9	—	87.6	—	—
InternVL (ours)	83.2	83.8	95.5	77.3	73.9	80.6	0.8	82.4

Table 2.9 Comparison of zero-shot image classification performance on ImageNet-1K of Multilingual [34, 67]. Multilingual evaluation involves five languages, including English (EN), Chinese (ZH), Japanese (JP), Arabic (AR), and Italian (IT)

Method	EN	ZH	JP	AR	IT	Avg.
M-CLIP [17]	—	—	—	—	20.2	—
CLIP-Italian [12]	—	—	—	—	22.1	—
Japanese-CLIP-ViT-B [107]	—	—	54.6	—	—	—
Taiyi-CLIP-ViT-H [185]	—	54.4	—	—	—	—
WuKong-ViT-L-G [44]	—	57.5	—	—	—	—
CN-CLIP-ViT-H [170]	—	59.6	—	—	—	—
AltCLIP-ViT-L [26]	74.5	59.6	—	—	—	—
EVA-02-CLIP-E+ [138]	82.0	3.6	5.0	0.2	41.2	—
OpenCLIP-XLM-R-H [56]	77.0	55.7	53.1	37.0	56.8	55.9
InternVL (ours)	83.2	64.5	61.5	44.9	65.7	64.0

2.4.3 Vision-Language Benchmarks

In this section, we evaluate the inherent capabilities of InternVL on various vision-language tasks.

Zero-Shot Image Classification We conduct thorough validation of the zero-shot image classification capability of InternVL. As depicted in Table 2.8, InternVL attains leading performance on various ImageNet variants [34, 49, 50, 124, 151] and ObjectNet [9]. Compared to EVA-02-CLIP-E+ [138], it exhibits stronger robustness to distribution shift, manifesting in a more consistent performance across ImageNet variants. Additionally, as shown in Table 2.9, our model showcases robust multilingual capabilities, outperforming competing models [17, 26, 56, 170] on the multilingual ImageNet-1K benchmark.

Table 2.10 Comparison of zero-shot video classification results on Kinetics 400/600/700. We report the top-1 accuracy and the mean of top-1 and top-5 accuracy

Method	#frame	K400 [20]		K600 [18]		K700 [19]	
		Top-1	Avg.	Top-1	Avg.	Top-1	Avg.
OpenCLIP-g [56]	1	—	63.9	—	64.1	—	56.9
OpenCLIP-G [56]	1	—	65.9	—	66.1	—	59.2
EVA-01-CLIP-g+ [138]	1	—	66.7	—	67.0	—	60.9
EVA-02-CLIP-E+ [138]	1	—	69.8	—	69.3	—	63.4
InternVL (ours)	1	—	71.0	—	71.3	—	65.7
ViCLIP [161]	8	64.8	75.7	62.2	73.5	54.3	66.4
InternVL (ours)	8	69.1	79.4	68.9	78.8	60.6	71.5

Zero-Shot Video Classification Following previous methods [123, 138, 161], we report the top-1 accuracy and the mean of top-1 and top-5 accuracy on Kinetics-400/600/700 [18–20]. As shown in Table 2.10, when sampling only a single center frame in each video, our method achieves an average accuracy of 76.1%, 75.5%, and 67.5% on the three datasets, surpassing EVA-02-CLIP-E+ [138] by +6.3, +6.2, and +4.1 points, respectively. Additionally, when uniformly sampling eight frames in each video, we obtain at least 3.3 points of improvement compared to the single-frame setting, outperforming ViCLIP [161] trained using web-scale video data.

Zero-Shot Image-Text Retrieval InternVL exhibits a powerful multilingual image–text retrieval capability. In Table 2.11, we evaluate the retrieval capabilities in English using the Flickr30K [122] and COCO [24] datasets, as well as in Chinese using the Flickr30K-CN [68] and COCO-CN [77]. In summary, InternVL demonstrates exceptional performance across most retrieval metrics.

2.4.4 Multi-Modal Benchmarks

In this section, we conduct an extensive evaluation across a series of benchmarks to assess InternVL-Chat’s multi-modal understanding and reasoning capability. The benchmarks employed in our study are categorized into four distinct types: OCR-related, general multi-modal, mathematical, and multi-turn conversation benchmarks. As shown in Tables 2.12 and 2.13, our model delivers outstanding performance on the majority of these benchmarks.

OCR-related Image Understanding We evaluate InternVL-Chat’s performance across four key dimensions of OCR: document comprehension (DocVQA [112]), chart understanding (ChartQA [110]), infographic understanding (InfographicVQA [111]), and scene text interpretation (TextVQA [134]). Additionally, we employ OCRBench [93] to perform a comprehensive evaluation of the model’s overall OCR capabilities. As shown in Table 2.12, our model demonstrates performance on par with proprietary models across these benchmarks and notably surpasses the open-

Table 2.11 Comparison of zero-shot image–text retrieval performance. We evaluate the retrieval capability in English using the Flickr30K [122] and COCO [24], as well as in Chinese using Flickr30K-CN [68] and COCO-CN [77]. [†]BLIP-2 [74] is fine-tuned on COCO and zero-shot transferred to Flickr30K, contributing to the enhanced zero-shot performance on Flickr30K

Method	Multi-Lingual	Flickr30K (English, 1K test set) [122]					
		Image → Text			Text → Image		
		R@1	R@5	R@10	R@1	R@5	R@10
Florence [180]	×	90.9	99.1	—	76.7	93.6	—
ONE-PEACE [153]	×	90.9	98.8	99.8	77.2	93.5	96.2
OpenCLIP-g [56]	×	91.4	99.2	99.6	77.7	94.1	96.9
EVA-01-CLIP-g+ [138]	×	91.6	99.3	99.8	78.9	94.5	96.9
CoCa [176]	×	92.5	99.5	99.9	80.4	95.7	97.7
OpenCLIP-G [56]	×	92.9	99.3	99.8	79.5	95.0	97.1
EVA-02-CLIP-E+ [138]	×	93.9	99.4	99.8	78.8	94.2	96.8
BLIP-2 [†] [74]	×	97.6	100.0	100.0	89.7	98.1	98.9
InternVL (ours)	✓	94.7	99.6	99.9	81.7	96.0	98.2
COCO (English, 5K test set) [24]							
Florence [180]	×	64.7	85.9	—	47.2	71.4	—
ONE-PEACE [153]	×	64.7	86.0	91.9	48.0	71.5	79.6
OpenCLIP-g [56]	×	66.4	86.0	91.8	48.8	73.3	81.5
EVA-01-CLIP-g+ [138]	×	68.2	87.5	92.5	50.3	74.0	82.1
CoCa [176]	×	66.3	86.2	91.8	51.2	74.2	82.0
OpenCLIP-G [56]	×	67.3	86.9	92.6	51.4	74.9	83.0
EVA-02-CLIP-E+ [138]	×	68.8	87.8	92.8	51.1	75.0	82.7
InternVL (ours)	✓	70.6	89.0	93.5	54.1	77.3	84.6
Flickr30K-CN (Chinese, 1K test set) [68]							
WuKong-ViT-L [44]	×	76.1	94.8	97.5	51.7	78.9	86.3
R2D2-ViT-L [168]	×	77.6	96.7	98.9	60.9	86.8	92.7
AltCLIP-ViT-H [26]	✓	88.9	98.5	99.5	74.5	92.0	95.5
CN-CLIP-ViT-H [170]	×	81.6	97.5	98.8	71.2	91.4	95.5
OpenCLIP-XLM-R-H [56]	✓	86.1	97.5	99.2	71.0	90.5	94.9
InternVL (ours)	✓	90.3	98.8	99.7	75.1	92.9	96.4
COCO-CN (Chinese, 1K test set) [77]							
WuKong-ViT-L [44]	×	55.2	81.0	90.6	53.4	80.2	90.1
R2D2-ViT-L [168]	×	63.3	89.3	95.7	56.4	85.0	93.1
Taiyi-CLIP-ViT-H [185]	×	—	—	—	60.0	84.0	93.3
CN-CLIP-ViT-H [170]	×	63.0	86.6	92.9	69.2	89.9	96.1
OpenCLIP-XLM-R-H [56]	✓	70.0	91.5	97.0	66.1	90.8	96.0
InternVL (ours)	✓	68.8	92.0	96.7	68.9	91.9	96.5

source LLaVA-NeXT [89]. In particular, our model performs exceptionally well on ChartQA and OCRBench, outperforming other competing proprietary models.

General Multimodal Evaluation In addition to OCR-related benchmarks, we evaluate our model on several general multi-modal benchmarks, as shown in

Table 2.12 Comparison on 5 OCR-related benchmarks and math dataset. OCR-related benchmarks include: DocVQA test [112], ChartQA test [110], InfographicVQA test [111], TextVQA val [134], and OCRBench [93]. Additionally, the math dataset includes MathVista testmini [100]. * denotes that Rosetta OCR tokens are used in the testing of TextVQA. The results of OCRBench are collected from the OpenCompass leaderboard [30]

Model	Open-source	#param	OCR-related Benchmarks					Math	
			DocVQA	ChartQA	InfoVQA	TextVQA	OCRBench	MathVista	Math
GPT-4V-1106 [1]	✗	—	88.4	78.5	—	78.0	645	—	49.9
Gemini Ultra 1.0 [142]	✗	—	90.9	80.8	80.3	82.3	—	—	53.0
Gemini Pro 1.0 [142]	✗	—	88.1	74.1	75.2	74.6	659	—	45.2
Gemini Pro 1.5 [125]	✗	—	86.5	81.3	72.7	73.5	—	—	52.1
Qwen-VL-Max [5]	✗	—	93.1	79.8	73.4	—	723	—	51.0
Qwen-VL-Plus [5]	✗	—	91.4	78.1	—	—	694	—	43.3
Claude-3 Opus [3]	✗	—	89.3	80.8	—	—	694	—	50.5
Claude-3 Sonnet [3]	✗	—	89.5	81.1	—	—	646	—	47.9
Claude-3 Haiku [3]	✗	—	88.8	81.7	—	—	658	—	46.4
HPT Pro [54]	✗	—	—	—	—	—	—	—	—
MM1 [113]	✗	30B	—	—	—	73.5	—	—	39.4
Step-1V [137]	✗	100B	—	—	—	—	625	—	44.8
Grok-1.5V [167]	✗	—	85.6	76.1	—	78.1	—	—	52.8
Text-Monkey [95]	✓	10B	66.7	59.9	28.6	64.3	561	—	—
DocOwl-1.5 [52]	✓	8B	82.2	70.2	50.7	68.6	599	—	—
Mini-Gemini [78]	✓	35B	—	—	—	74.1*	—	—	43.3
LLaVA-NeXT [89]	✓	35B	84.0	68.7	51.5	69.5*	574	—	46.5
InternVL-Chat (ours)	✓	26B	90.9	83.8	72.5	80.6	724	—	53.5

Table 2.13 Comparison on 10 General Multimodal Benchmarks. The benchmarks include MME [41], RealWorldQA [167], AI2D test [60], MMMU val [182], MMBench-EN/CN test [92], CCBench dev [92], MMVet [179], SEED Image [70], and HallusionBench [45]. The MME results we report are the sum of the perception and cognition scores. The results of MMBench, CCBench, and HallusionBench are collected from the OpenCompass leaderboard [30]

Model	Open-source	#param	MME	RWQA	AI2D	MMMU	MMB-EN/CN	CCB	MMVet	SEED	HallB
GPT-4V-1106 [1]	✗	—	1926.6	61.4	78.2	56.8	77.0 / 74.4	46.5	67.6	71.6	46.5
Gemini Ultra 1.0 [142]	✗	—	—	—	79.5	59.4	— / —	—	—	—	—
Gemini Pro 1.0 [142]	✗	—	1933.4	—	73.9	47.9	73.6 / 74.3	52.5	64.3	70.7	45.2
Gemini Pro 1.5 [125]	✗	—	67.5	80.3	58.5	— / —	—	—	—	—	—
Qwen-VL-Max [5]	✗	—	2433.6	—	79.3	51.3	77.6 / 75.7	63.5	66.6	—	41.2
Qwen-VL-Plus [5]	✗	—	2183.4	—	75.9	45.2	67.0 / 70.7	55.1	61.1	72.7	40.6
Claude-3 Opus [3]	✗	—	1586.8	49.8	88.1	59.4	63.3 / 59.2	26.3	58.1	—	37.8
Claude-3 Sonnet [3]	✗	—	1625.9	51.9	88.7	53.1	67.8 / 64.2	27.8	—	—	41.3
Claude-3 Haiku [3]	✗	—	1453.2	—	86.7	50.2	60.7 / 57.2	24.5	—	—	39.2
HPT Pro [54]	✗	—	—	—	—	52.0	77.5 / 76.7	—	—	73.1	—
MM1 [113]	✗	30B	2069.0	—	—	44.7	75.1 / —	—	48.7	72.1	—
Step-1V [137]	✗	100B	2206.4	—	79.2	49.9	80.7 / 79.9	71.2	63.3	70.3	48.4
Grok-1.5V [167]	✗	—	—	68.7	88.3	—	— / —	—	—	—	—
Text-Monkey [95]	✓	10B	—	—	—	—	— / —	—	—	—	—
DocOwl-1.5 [52]	✓	8B	—	—	—	—	— / —	—	—	—	—
Mini-Gemini [78]	✓	35B	2141.0	—	—	48.0	80.6 / —	—	59.3	—	—
LLaVA-NeXT [89]	✓	35B	2028.0	—	74.9	51.1	81.1 / 79.0	49.2	57.4	75.9	34.8
InternVL-Chat (ours)	✓	26B	2187.8	66.0	80.7	45.2	82.2 / 82.0	69.8	62.8	76.0	49.3

Table 2.13. We use RealWorldQA [167] to evaluate the model’s real-world spatial understanding capabilities. HallusionBench [45] is employed to assess its ability to control hallucinations. Additionally, MMMU [182] is utilized to evaluate the model’s multidisciplinary capabilities, and AI2D [60] to assess its understanding of science diagrams. We also test the model’s proficiency in Chinese and understanding of Chinese culture with the MMBench-CN test [92] and CCBench [92], respectively. Other comprehensive benchmarks such as MME [41], MMBench-EN [92], MMVet [179], SEED [70], and MMT-Bench [174] are also used to assess the model’s visual understanding and reasoning abilities (Table 2.14).

Compared to other open-source models like Text-Monkey [95], DocOwl-1.5 [52], and LLaVA-NeXT [89], our InternVL-Chat significantly closes the gap with proprietary models in these benchmarks. Specifically, our model achieves the best performance on HallusionBench [45], demonstrating its outstanding ability to reduce hallucinations. Moreover, thanks to our high-quality bilingual dataset, our model exhibits robust Chinese language capabilities, significantly surpassing both open-source and proprietary methods on MMBench-CN and CCBench. However, InternVL-Chat surpasses MM1 [113] and is comparable to Gemini Pro 1.0 [142] on MMMU. We attribute this modest decrement to the smaller size of the language model, a phenomenon similarly observed in the MMT-Bench [174] results, as shown in Table 2.15.

Math Reasoning MathVista [100] is a benchmark designed to integrate challenges from various mathematical and visual tasks. Completing these tasks requires a deep understanding of visuals, logical thinking, and math knowledge—areas where many proprietary commercial models encounter significant difficulties. As shown in Table 2.12, our model outperforms others, including GPT-4V [116], by a clear margin in this benchmark, showcasing its ability to handle mathematically demanding tasks.

Multi-Turn Conversation Compared to single-turn dialogs, multi-turn conversations align more with human preferences. In practical usage, multi-turn dialog is the preferred mode for general-purpose assistants to engage with humans in solving a variety of tasks. Therefore, we opt to utilize ConvBench [91] for evaluating multi-turn conversations, which progressively assesses the perception, reasoning, and creativity capabilities of MLLMs. As shown in Table 2.14, InternVL-Chat demonstrates strong performance compared to other open-source models, although it still lags behind GPT-4V by a noticeable margin.

2.4.5 Ablation Study

Hyper-Parameters of InternViT-6B As discussed in Sect. 2.3.1, we explored variations in model depth {32, 48, 64, 80}, head dimension {64, 128}, and MLP ratio {4, 8}, resulting in 16 distinct models. In selecting the optimal model, we

Table 2.14 Comparison on ConvBench. ConvBench [91] is a multi-turn conversation evaluation benchmark designed for MLLMs. The table presents win rates against humans, where S_1 , S_2 , and S_3 represent the scores for perception, reasoning, and creation, respectively. R_2 is calculated as $(S_1 + S_2 + S_3)/3$, reflecting the average performance across three turns. R_1 is derived from $(R_2 + S_0)/2$, indicating the model’s overall score

Model	Open-Source	#param	ConvBench (Pairwise Grading)					ConvBench (Direct Grading)						
			R_1	R_2	S_1	S_2	S_3	S_0	R_1	R_2	S_1	S_2	S_3	S_0
GPT-4V [1]	✗	—	39.51	38.47	38.47	39.34	37.61	40.55	7.09	7.30	7.30	7.48	7.12	6.88
Claude-3 Opus [3]	✗	—	36.60	37.49	38.99	39.17	34.32	35.70	6.54	6.75	6.53	7.04	6.68	6.32
Reka Flash [119]	✗	—	25.60	24.67	25.13	27.56	21.32	26.52	6.78	6.86	6.93	7.25	6.41	6.70
Gemini Pro 1.0 [142]	✗	—	8.44	8.55	9.01	9.36	7.28	8.32	4.42	4.60	5.18	4.95	3.66	4.24
ShareGPT4V-13B [23]	✓	13B	17.56	17.45	17.85	18.72	15.77	17.68	4.85	5.03	5.16	5.06	4.86	4.67
LLaVA-1.5-13B [88]	✓	13B	16.93	18.08	20.45	18.02	15.77	15.77	4.94	5.14	5.03	5.41	4.99	4.74
XComposer2 [35]	✓	8B	15.83	16.41	17.16	19.06	13.00	15.25	5.82	5.98	5.98	6.17	5.78	5.66
mPLUG-Owl2 [173]	✓	8B	14.93	15.83	17.50	17.16	12.82	14.04	5.04	5.17	4.98	5.38	5.14	4.91
Qwen-VL-Chat [5]	✓	10B	14.33	14.62	16.29	18.37	9.19	14.04	5.54	5.65	5.96	5.78	5.22	5.43
MiniGPT-4 [192]	✓	8B	10.95	10.80	11.61	11.27	9.53	11.09	3.85	4.04	3.99	4.40	3.73	3.66
LLaMA-A-V2 [42]	✓	7B	9.04	9.59	8.84	10.92	9.01	8.49	4.77	4.91	4.77	5.47	4.48	4.64
InternVL-Chat (ours)	✓	26B	17.65	20.22	26.00	17.33	17.33	15.08	5.60	5.76	6.11	5.93	5.25	5.43

Table 2.15 Comparison on MMT-Bench. MMT-Bench [174] is a comprehensive benchmark designed to assess MLLMs across massive multimodal tasks requiring expert knowledge and deliberate visual recognition, localization, reasoning, and planning. The overall score is computed across 162 subtasks, excluding visual recognition as denoted by *

Model	Open-source	#param	Overall	Overall*
GPT-4V [1]	✗	—	62.0	55.5
Qwen-VL-Plus [4]	✗	—	62.3	56.6
Gemini Pro 1.0 [142]	✗	—	61.6	55.1
Claude-3 Haiku [3]	✗	—	52.2	46.4
LLaVA-NeXT [89]	✓	35B	60.8	56.3
XComposer2 [35]	✓	8B	55.7	50.0
BLIP-2-XXL [74]	✓	12B	54.8	49.1
Yi-VL-34B [175]	✓	35B	54.2	48.6
Monkey-Chat [142]	✓	10B	53.4	46.0
DeepSeek-VL [99]	✓	7B	53.2	46.5
CogVLM-Chat [156]	✓	17B	51.6	44.2
InternVL-Chat (ours)	✓	26B	59.0	56.2

Table 2.16 Comparison of hyperparameters in InternViT-6B. The throughput (img/s) and GFLOPs are measured at 224×224 input resolution, with a batch size of 1 or 128 on an A100 GPU

Name	Width	Depth	MLP	#heads	#param	FLOPs	Throughput	zs IN
variant 1	3968	32	15872	62	6051M	1571G	35.5 / 66.0	65.8
variant 2	3200	48	12800	50	5903M	1536G	28.1 / 64.9	66.1
variant 3	3200	48	12800	25	5903M	1536G	28.0 / 64.6	66.2
variant 4	2496	48	19968	39	5985M	1553G	28.3 / 65.3	65.9
variant 5	2816	64	11264	44	6095M	1589G	21.6 / 61.4	66.2
variant 6	2496	80	9984	39	5985M	1564G	16.9 / 60.1	66.2

initially narrowed down our focus to six models, chosen based on their throughput, as listed in Table 2.16. These models underwent further evaluation using contrastive learning on a 100M subset of LAION-en [126] over 10k iterations. For the experimental setup, the primary difference is the use of a randomly initialized text encoder from CLIP-L [123], in order to speed up the training. For the sake of accuracy, inference speed, and training stability, we ultimately chose variant 3 as the final InternViT-6B.

Dynamic Resolution Matters As shown in Fig. 2.8, we investigate the effectiveness of dynamic resolution across various multi-modal benchmarks. We find that not all tasks require high resolution. Specifically, tasks related to OCR, such as DocVQA, InfoVQA, TextVQA, and OCRBench, benefit from increased resolution. However, tasks like AI2D, MMMU, MMBench, and HallusionBench exhibit a slight decline in performance at higher resolutions. Overall, InternVL-Chat demonstrates strong robustness to dynamic resolution. It can adjust the resolution based on

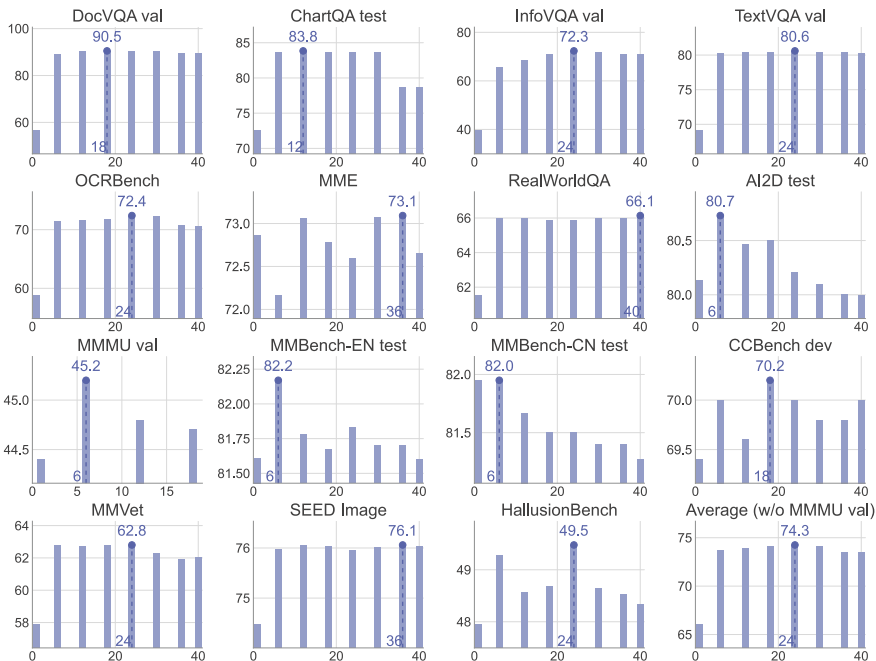


Fig. 2.8 Comparison of InternVL-Chat performance across different image resolutions. The X-axis represents the number of tiles, while the Y-axis indicates benchmark performance. The highest value and its corresponding number of tiles are highlighted. The scores of MME [41] and OCRBench [93] have been normalized to a maximum score of 100. We found that although only 1 to 12 tiles were used during training, it is possible to zero-shot scale up to 40 tiles (i.e., 4K resolution) during testing. Note that since MMMU [182] includes multiple images per sample, it may run out of memory when the number of tiles is large. Therefore, we only tested up to 18 tiles maximum, and MMMU was not included when calculating the average score

the specific requirements of each task, ensuring optimal performance where high resolution is beneficial and conserving resources where it is not.

2.5 Conclusion

In this chapter, we present InternVL, a large-scale vision-language foundation model that scales up the vision foundation model to 6 billion parameters and aligns it with LLMs. This design merges the large-scale vision foundation model with a language middleware, leveraging web-scale image-text data from various sources for efficient training. It bridges the gap between vision foundation models and LLMs and demonstrates proficiency in tasks such as image and video classification, cross-modal retrieval, image captioning, visual question answering, and multi-modal

dialog. This chapter aims to provide insights into the development and capabilities of InternVL, contributing to the broader understanding of VLLM.

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 (2023)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. *NeurIPS* **35**, 23716–23736 (2022)
3. Anthropic: The claude 3 model family: Opus, sonnet, haiku. <https://www.anthropic.com> (2024)
4. Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., et al.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023)
5. 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 (2023)
6. Bai, Y., Du, X., Liang, Y., Jin, Y., Liu, Z., Zhou, J., Zheng, T., Zhang, X., Ma, N., Wang, Z., et al.: COIG-CQIA: quality is all you need for Chinese instruction fine-tuning. arXiv preprint arXiv:2403.18058 (2024)
7. Baichuan: Baichuan 2: open large-scale language models. arXiv preprint arXiv:2309.10305 (2023)
8. Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. In: *ICLR* (2022)
9. Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., Katz, B.: ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *NeurIPS* **32**, 9448–9458 (2019)
10. Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., Oord, A.v.d.: Are we done with imagenet? arXiv preprint arXiv:2006.07159 (2020)
11. Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., et al.: Deepseek LLM: scaling open-source language models with longtermism. arXiv preprint arXiv:2401.02954 (2024)
12. Bianchi, F., Attanasio, G., Pisoni, R., Terragni, S., Sarti, G., Lakshmi, S.: Contrastive language-image pre-training for the italian language. arXiv preprint arXiv:2108.08688 (2021)
13. Biten, A.F., Tito, R., Mafla, A., Gomez, L., Rusinol, M., Valveny, E., Jawahar, C., Karatzas, D.: Scene text visual question answering. In: *ICCV*, pp. 4291–4301 (2019)
14. Byeon, M., Park, B., Kim, H., Lee, S., Baek, W., Kim, S.: COYO-700m: Image-text pair dataset (2022)
15. Cai, Z., Cao, M., Chen, H., Chen, K., Chen, K., Chen, X., Chen, X., Chen, Z., Chen, Z., Chu, P., et al.: Internlm2 technical report. arXiv preprint arXiv:2403.17297 (2024)
16. Cao, J., Xiao, J.: An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In: *COLING*, pp. 1511–1520 (2022)
17. Carlsson, F., Eisen, P., Rekathati, F., Sahlgren, M.: Cross-lingual and multilingual clip. In: *LREC*, pp. 6848–6854 (2022)
18. Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. arXiv preprint arXiv:1808.01340 (2018)
19. Carreira, J., Noland, E., Hillier, C., Zisserman, A.: A short note on the kinetics-700 human action dataset. arXiv preprint arXiv:1907.06987 (2019)
20. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: *CVPR*, pp. 6299–6308 (2017)

21. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: CVPR, pp. 3558–3568 (2021)
22. Chen, G.H., Chen, S., Zhang, R., Chen, J., Wu, X., Zhang, Z., Chen, Z., Li, J., Wan, X., Wang, B.: ALLaVA: harnessing GPT4V-synthesized data for a lite vision-language model. arXiv preprint arXiv:2402.11684 (2024)
23. 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 (2023)
24. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
25. Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., et al.: Pali: A jointly-scaled multilingual language-image model. In: ICLR (2022)
26. Chen, Z., Liu, G., Zhang, B.W., Yang, Q., Wu, L.: Altclip: Altering the language encoder in clip for extended language capabilities. In: ACL, pp. 8666–8682 (2023)
27. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Muyan, Z., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: scaling up vision foundation models and aligning for generic visual-linguistic tasks. arXiv preprint arXiv:2312.14238 (2023)
28. Chng, C.K., Liu, Y., Sun, Y., Ng, C.C., Luo, C., Ni, Z., Fang, C., Zhang, S., Han, J., Ding, E., et al.: ICDAR2019 robust reading challenge on arbitrary-shaped text-RRC-ArT. In: ICDAR, pp. 1571–1576 (2019)
29. Clark, C., Gardner, M.: Simple and effective multi-paragraph reading comprehension. In: ACL, pp. 845–855 (2018)
30. Contributors, O.: Opencompass: a universal evaluation platform for foundation models. <https://github.com/open-compass/opencompass> (2023)
31. Cui, Y., Yang, Z., Yao, X.: Efficient and effective text encoding for Chinese llama and alpaca. arXiv preprint arXiv:2304.08177 (2023)
32. Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J.M., Parikh, D., Batra, D.: Visual dialog. In: CVPR, pp. 326–335 (2017)
33. Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A.P., Caron, M., Geirhos, R., Alabdulmohsin, I., et al.: Scaling vision transformers to 22 billion parameters. In: ICML, pp. 7480–7512 (2023)
34. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: CVPR, pp. 248–255 (2009)
35. Dong, X., Zhang, P., Zang, Y., Cao, Y., Wang, B., Ouyang, L., Wei, X., Zhang, S., Duan, H., Cao, M., et al.: Internlm-xcomposer2: mastering free-form text-image composition and comprehension in vision-language large model. arXiv preprint arXiv:2401.16420 (2024)
36. Dong, X., Zhang, P., Zang, Y., Cao, Y., Wang, B., Ouyang, L., Zhang, S., Duan, H., Zhang, W., Li, Y., et al.: InternLM-XComposer2-4KHD: a pioneering large vision-language model handling resolutions from 336 pixels to 4K HD. arXiv preprint arXiv:2404.06512 (2024)
37. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: ICLR (2020)
38. Du, Z., Qian, Y., Liu, X., Ding, M., Qiu, J., Yang, Z., Tang, J.: GLM: general language model pretraining with autoregressive blank infilling. In: ACL, pp. 320–335 (2022)
39. Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
40. Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva: exploring the limits of masked visual representation learning at scale. In: CVPR, pp. 19358–19369 (2023)
41. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Qiu, Z., Lin, W., Yang, J., Zheng, X., et al.: MME: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2023)

42. Gao, P., Han, J., Zhang, R., Lin, Z., Geng, S., Zhou, A., Zhang, W., Lu, P., He, C., Yue, X., et al.: Llama-adapter v2: parameter-efficient visual instruction model. arXiv preprint arXiv:2304.15010 (2023)
43. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: elevating the role of image understanding in visual question answering. In: CVPR, pp. 6904–6913 (2017)
44. Gu, J., Meng, X., Lu, G., Hou, L., Minzhe, N., Liang, X., Yao, L., Huang, R., Zhang, W., Jiang, X., et al.: Wukong: A 100 million large-scale Chinese cross-modal pre-training benchmark. NeurIPS **35**, 26418–26431 (2022)
45. Guan, T., Liu, F., Wu, X., Xian, R., Li, Z., Liu, X., Wang, X., Chen, L., Huang, F., Yacoob, Y., et al.: Hallusionbench: an advanced diagnostic suite for entangled language hallucination & visual illusion in large vision-language models. arXiv preprint arXiv:2310.14566 (2023)
46. He, C., Jin, Z., Xu, C., Qiu, J., Wang, B., Li, W., Yan, H., Wang, J., Lin, D.: Wanjuan: a comprehensive multimodal dataset for advancing English and Chinese large models. arXiv preprint arXiv:2308.10755 (2023)
47. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR, pp. 16000–16009 (2022)
48. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
49. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: a critical analysis of out-of-distribution generalization. In: ICCV, pp. 8340–8349 (2021)
50. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: CVPR, pp. 15262–15271 (2021)
51. Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Dong, Y., Ding, M., et al.: CogAgent: a visual language model for GUI agents. arXiv preprint arXiv:2312.08914 (2023)
52. Hu, A., Xu, H., Ye, J., Yan, M., Zhang, L., Zhang, B., Li, C., Zhang, J., Jin, Q., Huang, F., et al.: mPLUG-DocOwl 1.5: Unified structure learning for OCR-free document understanding. arXiv preprint arXiv:2403.12895 (2024)
53. Hudson, D.A., Manning, C.D.: GQA: a new dataset for real-world visual reasoning and compositional question answering. In: CVPR, pp. 6700–6709 (2019)
54. HyperGAI Research Team: Introducing HPT: a family of leading multimodal LLMS. <https://www.hypergai.com/blog/introducing-hpt-a-family-of-leading-multimodal-llms> (2024)
55. Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., Keutzer, K.: Densenet: implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869 (2014)
56. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: Openclip. Zenodo. Version 0.1 (2021). <https://doi.org/10.5281/zenodo.5143773>
57. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML, pp. 4904–4916 (2021)
58. Jimmycarter: Textocr gpt-4v dataset (2023). <https://huggingface.co/datasets/jimmycarter/textocr-gpt4v>
59. Kafle, K., Price, B., Cohen, S., Kanan, C.: DVQA: understanding data visualizations via question answering. In: CVPR, pp. 5648–5656 (2018)
60. Kembhavi, A., Salvato, M., Kolve, E., Seo, M., Hajishirzi, H., Farhadi, A.: A diagram is worth a dozen images. In: ECCV, pp. 235–251 (2016)
61. Kembhavi, A., Seo, M., Schwenk, D., Choi, J., Farhadi, A., Hajishirzi, H.: Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In: CVPR, pp. 4999–5007 (2017)
62. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL, pp. 4171–4186 (2019)

63. Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., Park, S.: OCR-free document understanding transformer. In: ECCV (2022)
64. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. *IJCV* **123**, 32–73 (2017)
65. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *NeurIPS* **25**, 1106–1114 (2012)
66. LAION: Gpt-4v dataset (2023). <https://huggingface.co/datasets/laion/gpt4v-dataset>
67. LAION-AI: Clip benchmark: clip-like model evaluation (2023). https://github.com/LAION-AI/CLIP_benchmark
68. Lan, W., Li, X., Dong, J.: Fluency-guided cross-lingual image captioning. In: ACM MM, pp. 1549–1557 (2017)
69. Lerner, P., Ferret, O., Guinaudeau, C., Le Borgne, H., Besançon, R., Moreno, J.G., Lovón Melgarejo, J.: Viquae, a dataset for knowledge-based visual question answering about named entities. In: SIGIR, pp. 3108–3120 (2022)
70. 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* (2023)
71. Li, B., Zhang, P., Yang, J., Zhang, Y., Pu, F., Liu, Z.: Otterhd: A high-resolution multi-modality model. *arXiv preprint arXiv:2311.04219* (2023)
72. Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., Liu, Z.: Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726* (2023)
73. Li, C., Liu, W., Guo, R., Yin, X., Jiang, K., Du, Y., Du, Y., Zhu, L., Lai, B., Hu, X., et al.: PP-OCRv3: more attempts for the improvement of ultra lightweight OCR system. *arXiv preprint arXiv:2206.03001* (2022)
74. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML, pp. 19730–19742. PMLR, New York (2023)
75. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML, pp. 12888–12900 (2022)
76. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: Videochat: chat-centric video understanding. *arXiv preprint arXiv:2305.06355* (2023)
77. Li, X., Xu, C., Wang, X., Lan, W., Jia, Z., Yang, G., Xu, J.: COCO-CN for cross-lingual image tagging, captioning, and retrieval. *TMM* **21**(9), 2347–2360 (2019)
78. Li, Y., Zhang, Y., Wang, C., Zhong, Z., Chen, Y., Chu, R., Liu, S., Jia, J.: Mini-gemini: mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814* (2024)
79. Li, Z., Wang, X., Stengel-Eskin, E., Kortylewski, A., Ma, W., Van Durme, B., Yuille, A.L.: Super-clevr: A virtual benchmark to diagnose domain robustness in visual reasoning. In: CVPR, pp. 14963–14973 (2023)
80. Li, Z., Yang, B., Liu, Q., Ma, Z., Zhang, S., Yang, J., Sun, Y., Liu, Y., Bai, X.: Monkey: image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607* (2023)
81. Lin, B., Tang, Z., Ye, Y., Cui, J., Zhu, B., Jin, P., Zhang, J., Ning, M., Yuan, L.: Moe-llava: mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947* (2024)
82. 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, pp. 740–755 (2014)
83. Lin, Z., Liu, C., Zhang, R., Gao, P., Qiu, L., Xiao, H., Qiu, H., Lin, C., Shao, W., Chen, K., et al.: Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575* (2023)
84. Lindström, A.D., Abraham, S.S.: Clevr-math: A dataset for compositional language, visual and mathematical reasoning. *arXiv preprint arXiv:2208.05358* (2022)
85. Liu, F., Emerson, G., Collier, N.: Visual spatial reasoning. *TACL* **11**, 635–651 (2023)
86. Liu, F., Lin, K., Li, L., Wang, J., Yacoob, Y., Wang, L.: Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565* (2023)

87. Liu, F., Wang, X., Yao, W., Chen, J., Song, K., Cho, S., Yacoob, Y., Yu, D.: MMC: advancing multimodal chart understanding with large-scale instruction tuning. arXiv preprint arXiv:2311.10774 (2023)
88. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2023)
89. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: improved reasoning, ocr, and world knowledge (2024). <https://llava-vl.github.io/blog/2024-01-30-llava-next/>
90. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. *NeurIPS* **36**, 34892–34916 (2023)
91. Liu, S., Ying, K., Zhang, H., Yang, Y., Lin, Y., Zhang, T., Li, C., Qiao, Y., Luo, P., Shao, W., et al.: Convbench: A multi-turn conversation evaluation benchmark with hierarchical capability for large vision-language models. arXiv preprint arXiv:2403.20194 (2024)
92. 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 (2023)
93. Liu, Y., Li, Z., Li, H., Yu, W., Huang, M., Peng, D., Liu, M., Chen, M., Li, C., Jin, L., et al.: On the hidden mystery of OCR in large multimodal models. arXiv preprint arXiv:2305.07895 (2023)
94. 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)
95. Liu, Y., Yang, B., Liu, Q., Li, Z., Ma, Z., Zhang, S., Bai, X.: Textmonkey: an OCR-free large multimodal model for understanding document. arXiv preprint arXiv:2403.04473 (2024)
96. Liu, Z., He, Y., Wang, W., Wang, W., Wang, Y., Chen, S., Zhang, Q., Lai, Z., Yang, Y., Li, Q., Yu, J., et al.: Interngpt: solving vision-centric tasks by interacting with chatgpt beyond language. arXiv preprint arXiv:2305.05662 (2023)
97. Liu, Z., Lai, Z., Gao, Z., Cui, E., Zhu, X., Lu, L., Chen, Q., Qiao, Y., Dai, J., Wang, W.: Controllm: augment language models with tools by searching on graphs. arXiv preprint arXiv:2310.17796 (2023)
98. 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: *ICCV*, pp. 10012–10022 (2021)
99. Lu, H., Liu, W., Zhang, B., Wang, B., Dong, K., Liu, B., Sun, J., Ren, T., Li, Z., Sun, Y., et al.: Deepseek-vl: towards real-world vision-language understanding. arXiv preprint arXiv:2403.05525 (2024)
100. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: evaluating mathematical reasoning of foundation models in visual contexts. arXiv preprint arXiv:2310.02255 (2023)
101. Lu, P., Gong, R., Jiang, S., Qiu, L., Huang, S., Liang, X., Zhu, S.C.: Inter-GPS: interpretable geometry problem solving with formal language and symbolic reasoning. arXiv preprint arXiv:2105.04165 (2021)
102. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: multimodal reasoning via thought chains for science question answering. *NeurIPS* **35**, 2507–2521 (2022)
103. Lu, P., Qiu, L., Chang, K.W., Wu, Y.N., Zhu, S.C., Rajpurohit, T., Clark, P., Kalyan, A.: Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. arXiv preprint arXiv:2209.14610 (2022)
104. Luo, G., Zhou, Y., Zhang, Y., Zheng, X., Sun, X., Ji, R.: Feast your eyes: mixture-of-resolution adaptation for multimodal large language models. arXiv preprint arXiv:2403.03003 (2024)
105. Lv, T., Huang, Y., Chen, J., Cui, L., Ma, S., Chang, Y., Huang, S., Wang, W., Dong, L., Luo, W., et al.: Kosmos-2.5: a multimodal literate model. arXiv preprint arXiv:2309.11419 (2023)
106. Ma, C., Jiang, Y., Wu, J., Yuan, Z., Qi, X.: Groma: localized visual tokenization for grounding multimodal large language models. arXiv preprint arXiv:2404.13013 (2024)
107. Makoto Shiin Tianyu Zhao, K.S.: Construction and public release of language image pretraining models in Japanese. In: *MIRU* (2022)

108. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: CVPR, pp. 11–20 (2016)
109. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: OK-VQA: A visual question answering benchmark requiring external knowledge. In: CVPR, pp. 3195–3204 (2019)
110. Masry, A., Do, X.L., Tan, J.Q., Joty, S., Hoque, E.: Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In: ACL, pp. 2263–2279 (2022)
111. Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., Jawahar, C.: Infographicvqa. In: WACV, pp. 1697–1706 (2022)
112. Mathew, M., Karatzas, D., Jawahar, C.: Docvqa: A dataset for VQA on document images. In: WACV, pp. 2200–2209 (2021)
113. McKinzie, B., Gan, Z., Fauconnier, J.P., Dodge, S., Zhang, B., Dufter, P., Shah, D., Du, X., Peng, F., Weers, F., et al.: MM1: methods, analysis & insights from multimodal llm pre-training. arXiv preprint arXiv:2403.09611 (2024)
114. Methani, N., Ganguly, P., Khapra, M.M., Kumar, P.: PlotQA: reasoning over scientific plots. In: WACV, pp. 1527–1536 (2020)
115. Mishra, A., Shekhar, S., Singh, A.K., Chakraborty, A.: OCR-VQA: visual question answering by reading text in images. In: ICDAR, pp. 947–952 (2019)
116. OpenAI: Gpt-4v(ision) system card (2023). https://cdn.openai.com/papers/GPTV_System_Card.pdf
117. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., et al.: Dinov2: learning robust visual features without supervision. In: TMLR (2023)
118. Ordonez, V., Kulkarni, G., Berg, T.: Im2text: describing images using 1 million captioned photographs. *NeurIPS* **24**, 1143–1151 (2011)
119. Ormazabal, A., Zheng, C., d’Autume, C.d.M., Yogatama, D., Fu, D., Ong, D., Chen, E., Lamprecht, E., Pham, H., Ong, I., et al.: Reka core, flash, and edge: a series of powerful multimodal language models. arXiv preprint arXiv:2404.12387 (2024)
120. Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., Launay, J.: The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. arXiv preprint arXiv:2306.01116 (2023)
121. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Wei, F.: Kosmos-2: grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824 (2023)
122. 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: ICCV, pp. 2641–2649 (2015)
123. 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)
124. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: ICML, pp. 5389–5400 (2019)
125. Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al.: Gemini 1.5: unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
126. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: an open large-scale dataset for training next generation image-text models. *NeurIPS* **35**, 25278–25294 (2022)
127. Schuhmann, C., Köpf, A., Vencu, R., Coombes, T., Beaumont, R.: Laion coco: 600m synthetic captions from laion2b-en (2022). <https://laion.ai/blog/laion-coco/>
128. Schwenk, D., Khandelwal, A., Clark, C., Marino, K., Mottaghi, R.: A-okvqa: a benchmark for visual question answering using world knowledge. In: ECCV, pp. 146–162 (2022)
129. Shah, S., Mishra, A., Yadati, N., Talukdar, P.P.: KVQA: knowledge-aware visual question answering. In: AAAI, vol. 33, pp. 8876–8884 (2019)
130. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: a large-scale, high-quality dataset for object detection. In: ICCV, pp. 8430–8439 (2019)

131. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: a cleaned, hypertexted, image alt-text dataset for automatic image captioning. In: ACL (2018)
132. Shi, B., Yao, C., Liao, M., Yang, M., Xu, P., Cui, L., Belongie, S., Lu, S., Bai, X.: ICDAR2017 competition on reading Chinese text in the wild (RCTW-17). In: ICDAR, vol. 1, pp. 1429–1434 (2017)
133. Sidorov, O., Hu, R., Rohrbach, M., Singh, A.: Textcaps: a dataset for image captioning with reading comprehension. In: ECCV, pp. 742–758 (2020)
134. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards VQA models that can read. In: CVPR, pp. 8317–8326 (2019)
135. Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., Hassner, T.: TextOCR: towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In: CVPR, pp. 8802–8812 (2021)
136. Singh, M., Duval, Q., Alwala, K.V., Fan, H., Aggarwal, V., Adcock, A., Joulin, A., Dollár, P., Feichtenhofer, C., Girshick, R., et al.: The effectiveness of mae pre-pretraining for billion-scale pretraining. In: ICCV, pp. 5484–5494 (2023)
137. StepFun Research Team: Step-1v: A hundred billion parameter multimodal large model (2024). <https://platform.stepfun.com>
138. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: improved training techniques for clip at scale. arXiv preprint arXiv:2303.15389 (2023)
139. Sun, T., Zhang, X., He, Z., Li, P., Cheng, Q., Yan, H., Liu, X., Shao, Y., Tang, Q., Zhao, X., et al.: Moss: training conversational language models from synthetic data. arXiv preprint arXiv:2307.15020, vol. 7 (2023)
140. Sun, Y., Ni, Z., Chng, C.K., Liu, Y., Luo, C., Ng, C.C., Han, J., Ding, E., Liu, J., Karatzas, D., et al.: ICDAR 2019 competition on large-scale street view text with partial labeling-RRC-LSVT. In: ICDAR, pp. 1557–1562 (2019)
141. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Alpaca: a strong, replicable instruction-following model. Stanford Center for Research on Foundation Models 3(6), 7 (2023). <https://crfm.stanford.edu/2023/03/13/alpaca.html>
142. 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 (2023)
143. Team, I.: InternLM: A multilingual language model with progressively enhanced capabilities (2023). <https://github.com/InternLM/InternLM>
144. Teknium: Openhermes 2.5: An open dataset of synthetic data for generalist LLM assistants (2023). <https://huggingface.co/datasets/teknium/OpenHermes-2.5>
145. Tian, C., Zhu, X., Xiong, Y., Wang, W., Chen, Z., Wang, W., Chen, Y., Lu, L., Lu, T., Zhou, J., et al.: MM-interleaved: interleaved image-text generative modeling via multi-modal feature synchronizer. arXiv preprint arXiv:2401.10208 (2024)
146. Tong, S., Liu, Z., Zhai, Y., Ma, Y., LeCun, Y., Xie, S.: Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. arXiv preprint arXiv:2401.06209 (2024)
147. Touvron, H., Cord, M., Jégou, H.: Deit III: revenge of the ViT. In: ECCV, pp. 516–533 (2022)
148. 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 (2023)
149. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
150. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: Coco-text: dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 (2016)
151. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. NeurIPS 32, 10506–10518 (2019)
152. Wang, J., Meng, L., Weng, Z., He, B., Wu, Z., Jiang, Y.G.: To see is to believe: prompting GPT-4V for better visual instruction tuning. arXiv preprint arXiv:2311.07574 (2023)
153. Wang, P., Wang, S., Lin, J., Bai, S., Zhou, X., Zhou, J., Wang, X., Zhou, C.: One-peace: exploring one general representation model toward unlimited modalities. arXiv preprint arXiv:2305.11172 (2023)

154. Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., et al.: Visionllm: large language model is also an open-ended decoder for vision-centric tasks. *NeurIPS* **36**, 61501–61513 (2023)
155. Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., et al.: Internimage: exploring large-scale vision foundation models with deformable convolutions. In: *CVPR*, pp. 14408–14419 (2023)
156. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., et al.: CogVLM: visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079* (2023)
157. Wang, W., Ren, Y., Luo, H., Li, T., Yan, C., Chen, Z., Wang, W., Li, Q., Lu, L., Zhu, X., et al.: The all-seeing project v2: towards general relation comprehension of the open world. *arXiv preprint arXiv:2402.19474* (2024)
158. Wang, W., Shi, M., Li, Q., Wang, W., Huang, Z., Xing, L., Chen, Z., Li, H., Zhu, X., Cao, Z., et al.: The all-seeing project: towards panoptic visual recognition and understanding of the open world. In: *ICLR* (2024)
159. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: *ICCV*, pp. 568–578 (2021)
160. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: PVT V2: improved baselines with pyramid vision transformer. *CVMJ* **8**(3), 415–424 (2022)
161. Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X., Chen, X., Wang, Y., Luo, P., Liu, Z., et al.: Internvid: a large-scale video-text dataset for multimodal understanding and generation. In: *ICLR* (2024)
162. Wang, Y., Li, K., Li, X., Yu, J., He, Y., Chen, G., Pei, B., Zheng, R., Xu, J., Wang, Z., et al.: Internvideo2: scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377* (2024)
163. Wei, H., Kong, L., Chen, J., Zhao, L., Ge, Z., Yang, J., Sun, J., Han, C., Zhang, X.: Vary: scaling up the vision vocabulary for large vision-language models. *arXiv preprint arXiv:2312.06109* (2023)
164. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS* **35**, 24824–24837 (2022)
165. Wei, T., Zhao, L., Zhang, L., Zhu, B., Wang, L., Yang, H., Li, B., Cheng, C., Lü, W., Hu, R., et al.: Skywork: a more open bilingual foundation model. *arXiv preprint arXiv:2310.19341* (2023)
166. Wu, S., Fei, H., Qu, L., Ji, W., Chua, T.S.: NExT-GPT: any-to-any multimodal LLM. *arXiv preprint arXiv:2309.05519* (2023)
167. X.ai: Grok-1.5 vision preview. <https://x.ai/blog/grok-1.5v> (2024)
168. Xie, C., Li, J., Cai, H., Kong, F., Wu, X., Song, J., Morimitsu, H., Yao, L., Wang, D., Leng, D., et al.: Zero and R2D2: A large-scale Chinese cross-modal benchmark and a vision-language framework. *arXiv preprint arXiv:2205.03860* (2022)
169. Xu, R., Yao, Y., Guo, Z., Cui, J., Ni, Z., Ge, C., Chua, T.S., Liu, Z., Sun, M., Huang, G.: LLaVA-UHD: an LMM perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703* (2024)
170. Yang, A., Pan, J., Lin, J., Men, R., Zhang, Y., Zhou, J., Zhou, C.: Chinese clip: contrastive vision-language pretraining in Chinese. *arXiv preprint arXiv:2211.01335* (2022)
171. Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., et al.: Qwen2 technical report. *arXiv preprint arXiv:2407.10671* (2024)
172. Ye, J., Hu, A., Xu, H., Ye, Q., Yan, M., Xu, G., Li, C., Tian, J., Qian, Q., Zhang, J., et al.: Ureader: universal OCR-free visually-situated language understanding with multimodal large language model. *arXiv preprint arXiv:2310.05126* (2023)
173. Ye, Q., Xu, H., Ye, J., Yan, M., Liu, H., Qian, Q., Zhang, J., Huang, F., Zhou, J.: mPLUG-Owl2: revolutionizing multi-modal large language model with modality collaboration. *arXiv preprint arXiv:2311.04257* (2023)

174. Ying, K., Meng, F., Wang, J., Li, Z., Lin, H., Yang, Y., Zhang, H., Zhang, W., Lin, Y., Liu, S., Lei, J., Lu, Q., Chen, R., Xu, P., Zhang, R., Zhang, H., Gao, P., Wang, Y., Qiao, Y., Luo, P., Zhang, K., Shao, W.: MMT-Bench: a comprehensive multimodal benchmark for evaluating large vision-language models towards multitask AGI. *arXiv preprint arXiv:2404.16006* (2024)
175. Young, A., Chen, B., Li, C., Huang, C., Zhang, G., Zhang, G., Li, H., Zhu, J., Chen, J., Chang, J., et al.: Yi: open foundation models by 01. AI. *arXiv preprint arXiv:2403.04652* (2024)
176. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: COCA: contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917* (2022)
177. Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J.T., Li, Z., Weller, A., Liu, W.: Metamath: bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284* (2023)
178. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: *ECCV*, pp. 69–85 (2016)
179. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: MM-Vet: evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490* (2023)
180. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: a new foundation model for computer vision. *arXiv preprint arXiv:2111.11432* (2021)
181. Yuan, T.L., Zhu, Z., Xu, K., Li, C.J., Mu, T.J., Hu, S.M.: A large Chinese text dataset in the wild. *J. Comput. Sci. Technol.* **34**, 509–521 (2019)
182. 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* (2023)
183. Zhai, X., Kolesnikov, A., Hounsby, N., Beyer, L.: Scaling vision transformers. In: *CVPR*, pp. 12104–12113 (2022)
184. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., Beyer, L.: Lit: zero-shot transfer with locked-image text tuning. In: *CVPR*, pp. 18123–18133 (2022)
185. Zhang, J., Gan, R., Wang, J., Zhang, Y., Zhang, L., Yang, P., Gao, X., Wu, Z., Dong, X., He, J., et al.: Fengshenbang 1.0: being the foundation of Chinese cognitive intelligence. *arXiv preprint arXiv:2209.02970* (2022)
186. Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., Qiao, Y.: Llama-adapter: efficient fine-tuning of language models with zero-init attention. In: *ICLR* (2024)
187. Zhang, R., Zhou, Y., Jiang, Q., Song, Q., Li, N., Zhou, K., Wang, L., Wang, D., Liao, M., Yang, M., et al.: ICDAR 2019 robust reading challenge on reading Chinese text on signboard. In: *ICDAR*, pp. 1577–1581 (2019)
188. Zhang, S., Sun, P., Chen, S., Xiao, M., Shao, W., Zhang, W., Chen, K., Luo, P.: GPT4Roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601* (2023)
189. Zhao, B., Wu, B., Huang, T.: SVIT: scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087* (2023)
190. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al.: Judging LLM-as-a-judge with MT-bench and chatbot arena. *NeurIPS* **36**, 46595–46623 (2024)
191. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: *CVPR*, pp. 633–641 (2017)
192. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: enhancing vision-language understanding with advanced large language models. In: *ICLR* (2024)

Chapter 3

Multimodal Large Language Models for Video Understanding



Yi Wang, Jiashuo Yu, Yinan He, Limin Wang, and Yu Qiao

Abstract This chapter presents an innovative approach to video-language modeling, which enables video understanding across multiple modalities. The approach combines three key training strategies: masked video modeling, cross-modal contrastive learning, and next token prediction. The model architecture scales to 6B parameters and is trained with a progressive strategy. A distinguishing feature of this approach is its emphasis on spatiotemporal coherence, which is achieved through semantic video segmentation and multimodal caption generation that encompasses video, audio, and speech elements. This comprehensive framework demonstrates strong capabilities across diverse video understanding tasks, with particular strengths in video–text alignment and long-form video comprehension. The architecture enables sophisticated reasoning in video-centric dialog systems and shows robust performance in processing extended video sequences.

Keywords Multimodal Large Language Model · Video Understanding · Spatiotemporal Features · Multimodal Learning · Video–Text Alignment

3.1 Introduction

The development of transferable spatiotemporal representations is a cornerstone of computer vision research, with applications spanning various fields. These include video information retrieval [95, 100], interactive game control [53], robotic learning [71], autonomous driving systems [35], and scientific data analysis [9]. The rise of Large Language Models (LLMs) and their multimodal counterparts (MLLMs) has significantly impacted vision research and other domains. A key

Y. Wang (✉) · J. Yu · Y. He · Y. Qiao
Shanghai AI Laboratory, Shanghai, China
e-mail: wangyi@pjlab.org.cn; yujiashuo@pjlab.org.cn; heyinan@pjlab.org.cn;
qiaoyu@pjlab.org.cn

L. Wang
Nanjing University, Nanjing, China

challenge is effectively integrating video data into these large models to enhance video understanding capabilities.

Existing research has explored various methods for learning effective video representations. These include masked video modeling techniques [26, 32, 88, 107], which leverage masked reconstruction for learning spatiotemporal structures; cross-modal alignment methods [11, 12, 52, 106], which align video representations with other modalities like text and audio; and next-token prediction approaches [37, 82], which predict future video frames or textual descriptions. These methods often complement each other and can be synergistically combined. Several prior works [52, 94, 113] have successfully employed a two-stage training paradigm, integrating masked reconstruction and multimodal contrastive learning to achieve improved downstream task performance. Building upon this, this chapter introduces a novel approach that extends this progressive training scheme by incorporating video-based next-token prediction and significantly scaling the model size and training data.

This chapter presents InternVideo2, a family of video foundation models trained using a three-stage progressive learning approach. The training process involves: (1) learning fundamental spatiotemporal structures through unmasked video reconstruction; (2) aligning video representations with semantic information from audio and text modalities; and (3) enhancing open-ended reasoning abilities through next-token prediction using a video-centric dialog system.

The initial stage focuses on establishing basic spatiotemporal understanding by training the video encoder to reconstruct unmasked video tokens. This leverages the strengths of pre-trained vision encoders, such as InternViT [15] and VideoMAE-g [107], which serve as proxy models to guide the learning process. The subsequent cross-modal learning stage integrates audio and text encoders, enriching the model’s understanding and enabling it to handle video–audio tasks. Finally, the next token prediction stage further refines the model’s capabilities by integrating it with an LLM [115], enabling improved performance on open-ended tasks like video question answering (VQA) and video captioning.

InternVideo2’s training emphasizes data quality and consistency. A large-scale multimodal dataset, comprising 402 million data entries, was constructed. This dataset includes a substantial number of videos, video–text pairs, video–audio–speech–text pairs (from InternVid2), and image–text pairs. The InternVid2 subset focuses on semantically segmented video clips, employing a multimodal captioning approach to generate richer and more accurate descriptions.

InternVideo2’s performance is demonstrated across a wide spectrum of video understanding tasks, ranging from basic spatiotemporal perception to high-level reasoning tasks. The results, as given in Table 3.1 and Fig. 3.1, demonstrate state-of-the-art performance across multiple benchmarks, highlighting the model’s ability to capture and reason over complex video content effectively.

Table 3.1 InternVideo2 yields strong transferable visual and visual-linguistic representations across around 70 video understanding tasks, ranging from action recognition, video-text understanding, to video-centric dialog

Method	Video Classification				Video-Audio-Language Tasks			Video-Centric Dialogue			
	K400-FT	K400-ZS	ANet-TAL	MSRVTT-ZS	QVHighlight	AudioCaps	MVBench	EgoSchema	Perception Test		
CoCa [39, 100]	88.9	81.3	–	34.4	–	–	–	–	–		
VideoPrism [113]	–	85.4	37.8	39.7	–	–	–	–	–		
prev. SOTA	91.1	85.4	37.8	49.3	45.0	52.0	62.0	53.3	57.5		
InternVideo2	92.1	82.5	41.2	55.9	56.5	55.2	67.5	60.2	63.4		

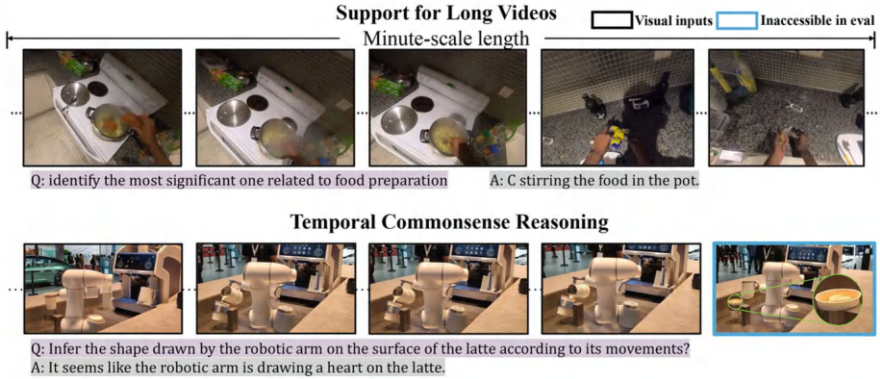


Fig. 3.1 InternVideo2 exhibits capability of long-form video understanding and procedure-aware reasoning. Image sourced from [93]

3.2 Related Work

Video Foundation Models The development of video foundation models (ViFMs) has emerged as a critical research direction in computer vision and multimedia understanding. These models can be categorized into several primary learning paradigms. The first approach [1, 11, 12, 52, 96, 100] involves video–text contrastive learning, where models learn to align visual and textual representations in a shared embedding space. Another significant paradigm is masked video modeling [88, 107], which requires models to reconstruct missing or masked portions of video frames, helping them learn spatial–temporal representations. The next token prediction paradigm [37, 82, 85] extends language modeling concepts to the video domain.

Architectural innovations have explored unified frameworks that combine multiple learning objectives. For instance, some approaches [52, 94] integrate masked modeling with contrastive learning, showing enhanced performance across both action recognition and video–language tasks. Other architectures [97, 113] have introduced modality-specific and shared components, allowing for both specialized processing and cross-modal integration.

The incorporation of audio information has further expanded the capabilities of video foundation models. Several frameworks [68, 76, 108] have been developed to leverage video–speech–transcript relationships, while others [11, 12] employ separate encoders for different modalities (video, audio, and text) to learn joint representations. These multimodal approaches have demonstrated particular strength in audio–visual tasks and general video understanding.

Multimodal Large Language Models The evolution of large language models (LLMs) has naturally led to the development of multimodal large language models (MLLMs), which extend text-based capabilities to handle various media types.

These models [58, 59, 64] have demonstrated remarkable zero-shot and few-shot learning abilities across diverse multimodal tasks, including visual question answering and cross-modal retrieval.

Works [18, 49, 65, 111] in MLLMs have focused on visual instruction tuning, where models are trained to follow natural language instructions for visual tasks. This approach has significantly improved their capability for visual dialog and understanding. In the video domain, specialized MLLMs have been developed to bridge the gap between video understanding and natural language processing, enabling more sophisticated video-based interactions and comprehension tasks.

These models typically combine video encoders with large language models through instruction-based training, enabling them to handle open-world video understanding tasks such as detailed video description, question answering, and temporal reasoning.

3.3 Method

The training process of **InternVideo2** consists of three distinct phases (depicted in Fig. 3.2): token-based spatiotemporal reconstruction, multimodal contrastive learning across video, audio, speech, and language modalities, and integration with a large language model for unified training.

Visual Architecture **InternVideo2** employs a modified Vision Transformer (ViT) [24] architecture as its visual backbone, enhanced with specialized projection layers for knowledge distillation. The architecture incorporates attention pooling mechanisms, building upon the advances in the field [15, 39]. The processing pipeline begins by extracting eight frames at regular intervals [89], followed by spatial downsampling to a 14×14 ($h \times w$) grid. The resulting spatiotemporal tokens are augmented with a class token and combined with three-dimensional positional encodings. Complete architectural specifications of the ViT-6B configuration are available in the supplementary materials of [93].

3.3.1 Phase 1: Token-Level Visual Reconstruction

The initial training phase implements a guided token reconstruction approach utilizing two specialized expert networks. We leverage InternVL-6B [15] and VideoMAEv2-g [88] as complementary teacher models, facilitating knowledge transfer through dedicated projection layers.

During the training process, complete video sequences are processed by both teacher networks, with 80% of the tokens masked on a per-frame basis. This masking strategy is informed by both semantic understanding (via InternVL [15]) and motion dynamics (via VideoMAEv2 [88]). The optimization focuses exclusively

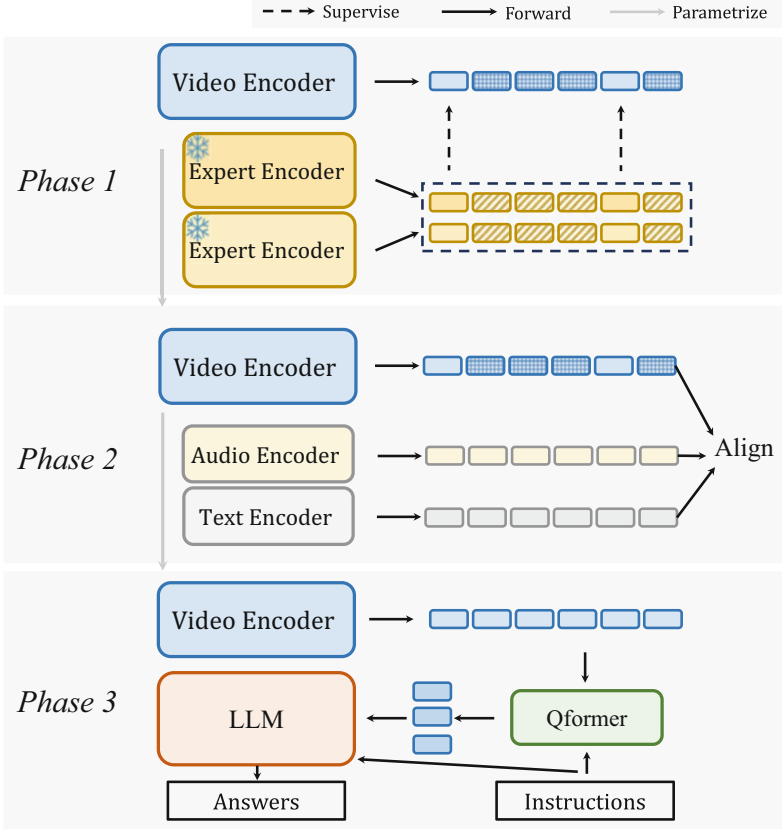


Fig. 3.2 Framework of **InternVideo2**. It consists of three consecutive training phases: unmasked video token reconstruction, multimodal contrastive learning, and next token prediction. In stage 1, the video encoder is trained from scratch, while in stages 2 and 3, it is initialized from the version used in the previous stage

on unmasked token alignment, employing mean squared error (MSE) minimization between the student and teacher networks. The reconstruction objective is formulated as:

$$\mathcal{L} = \frac{1}{Z} \sum (\alpha_1 |f^V(\mathbf{V}_p) - h(\mathbf{V}_p)|^2 + \alpha_2 |f^V(\mathbf{V}_p) - g(\mathbf{V}_p)|^2), \quad (3.1)$$

where f^V represents our visual encoder, h denotes InternViT-6B [15], and g corresponds to VideoMAEv2's ViT-g. The token position is indicated by p , with $f(\mathbf{V}_p)$ representing the corresponding encoded feature for input video $\mathbf{V}$. Z acts as a normalization term, while α_1 and α_2 control the relative contributions of each teacher model.

The implementation begins with a randomly initialized visual encoder. Through learnable multilayer perceptrons, we establish alignment between specific architectural components: the final six layers of InternVL, the last four layers of VideoMAEv2, and InternVL’s terminal token output. These alignments are computed using l_2 distance metrics, with the individual loss components combined additively during optimization. Post-training, the projection layers are removed, leaving only the core encoder architecture. This approach extends beyond previous work in UMT [52] and VideoPrism [113] by developing an encoder that exhibits both multimodal compatibility and enhanced temporal understanding for action recognition.

3.3.2 Phase 2: Multimodal Integration of Visual, Audio, and Linguistic Components

The second phase focuses on semantic enrichment through cross-modal integration of visual content with audio, speech, and textual information. Our architecture employs a large-scale video encoder complemented by more compact audio and text processing components. The audio processing utilizes a 12-layer transformer (90M parameters) initialized from BEATs [13], processing 64-dimensional log Mel filterbank spectrograms derived from 10-second segments using 25 ms Hamming windows with zero padding. The linguistic components leverage BERT-Large [23] architecture, with the initial 19 layers serving as the text encoder and the remaining five layers, augmented with cross-attention mechanisms, functioning as the multimodal decoder.

The training objectives incorporate multiple cross-modal alignment strategies, unified through the following composite loss function:

$$\mathcal{L} = \mathcal{L}_{\text{CON}} + \mathcal{L}_{\text{MAC}} + \mathcal{L}_{\text{MLM}}, \quad (3.2)$$

where $\mathcal{L}_{\text{MAC}}$ and $\mathcal{L}_{\text{MLM}}$ follow the formulation in [17]. The cross-modal contrastive component is defined as:

$$\begin{aligned} \mathcal{L}_{\text{CON}} &= \sum_{M, T_{M'}} \mathcal{L}_{\text{CON}}(M, T_{M'}) \\ &= - \sum_{M, T_{M'}} \left(\sum_{i=1}^N \log \frac{\exp(\text{sim}(f_i^M, f_i^{T_{M'}})/\tau)}{\sum_{j=1}^N \exp(\text{sim}(f_i^M, f_j^{T_{M'}})/\tau)} \right. \\ &\quad \left. + \sum_{i=1}^N \log \frac{\exp(\text{sim}(f_i^{T_{M'}}, f_i^M)/\tau)}{\sum_{j=1}^N \exp(\text{sim}(f_i^{T_{M'}}, f_j^M)/\tau)} \right), \end{aligned} \quad (3.3)$$

where f^V and f^T represent the encoded video and text embeddings respectively, M denotes the input modality, and $T_{M'}$ represents the corresponding textual descriptions. The function $\text{sim}(\cdot)$ computes cosine similarity, and τ represents a learnable temperature parameter.

The modality matching objective is formulated as:

$$\mathcal{L}_{\text{MAC}} = -y \log f_p(\mathbf{V}, \mathbf{T}) - (1 - y) \log(1 - f_p(\mathbf{V}, \mathbf{T})), \quad (3.4)$$

where $f_p(\mathbf{V}, \mathbf{T})$ evaluates the matching probability between video $\mathbf{V}$ and text $\mathbf{T}$, with y indicating genuine ($y = 1$) or mismatched ($y = 0$) pairs.

The language modeling component is defined as:

$$\mathcal{L}_{\text{MLM}} = -\log f_p^T(\mathbf{T}_j | \mathbf{T}_{<j}), \quad (3.5)$$

where $f_p^T(\mathbf{T}_j | \mathbf{T}_{<j})$ computes the conditional probability of token j given its predecessors in video caption $\mathbf{T}$.

The training process is optimized through a two-phase strategy:

Initial Cross-Modal Integration With a fixed audio encoder, we focus on aligning visual and linguistic representations. The training utilizes diverse image, video, and audiovisual datasets, with modality combinations defined as $M, T_{M'} \in I, T_I, V, T_V, VA, T_{VAS}, T_{VA}$.

Comprehensive Multimodal Refinement The final phase maintains a fixed vision encoder while optimizing audio–visual–textual alignments. This phase processes a curated dataset comprising 25M images, 50M audiovisual samples, and 0.5M audio clips. To maintain inference consistency and preserve downstream performance, masking is omitted during this phase. The modality combinations expand to $\{M, T_{M'}\} \in \{\{I, T_I\}, \{V, T_V\}, \{A, T_A\}, \{V, T_{VAS}\}, \{VA, T_{VA}\}\}$.

3.3.3 Phase 3: Next Token Prediction with Video-Centric Inputs

To deepen the semantic richness of **InternVideo2** and bolster its performance in video-focused dialog, we refine the model by integrating it with a large language model (LLM) that features a QFormer architecture, as described in [47, 48]. We adopt the progressive learning strategy outlined in [50], utilizing **InternVideo2** as the video encoding component and training a specialized version of BLIP for interaction with publicly available LLMs such as [40, 115].

Furthermore, we introduce an advanced post-training phase aimed at enhancing the model’s ability to capture fine-grained details and extended spatiotemporal dynamics. In this phase, each input video is segmented into up to six sub-videos, each with a resolution of 224×224 pixels, complemented by a single, globally resized sub-video of the same resolution. The model undergoes two more rounds

of training: the initial round employs eight-frame video segments, whereas the subsequent round utilizes 16-frame segments.

Throughout this additional training, we update both the video encoder and the BLIP QFormer, while the LLM is fine-tuned using the Low-Rank Adaptation (LoRA) technique [34], ensuring a balanced improvement across all components of the system.

3.4 Training Multimodal Data Overview

Our training datasets are summarized in Table 3.2. Among these, KMash and InternVid2 represent newly created resources, while the remaining datasets are publicly accessible.

3.4.1 Video-only Datasets for Masked Autoencoders

We have compiled an unlabeled video dataset, named *K-Mash*, derived from various action recognition databases such as [30, 33, 38, 69, 112], as further described in the supplementary materials. This collection includes a diverse array of video types, ranging from first- and third-person perspectives, with both short and long durations, and featuring a variety of settings. Additionally, we introduce *K-Mash_{2M}*, which includes an extra 844,000 videos sourced from YouTube to enhance diversity.

3.4.2 Multimodal Video Dataset with Audio and Speech

To reinforce video perception through additional modalities, we have developed a multimodal video dataset, denoted as **InternVid2**, which includes video, audio, and

Table 3.2 Summary of datasets used in InternVideo2 pretraining process

Pretraining stage	Dataset	Domain	# of clips	Annotation
Stage 1	KMash	Web Video	2M	-
Stage 2 (img-txt)	LAION, etc	Web Image	300M	Alt-text / Generated Caps
Stage 2 (vid-txt)	WebVid2M	Web Video	250k	Alt-text
	WebVid10M	Web Video	9.7M	Alt-text
	InternVid	Youtube Video	40M	Generated Caption
	InternVid2	Youtube Video	50M	Generated Caption
Stage 3	LLaVA, etc	Web Image/Video	2.1M	Conversation, QA

Table sourced from [93]

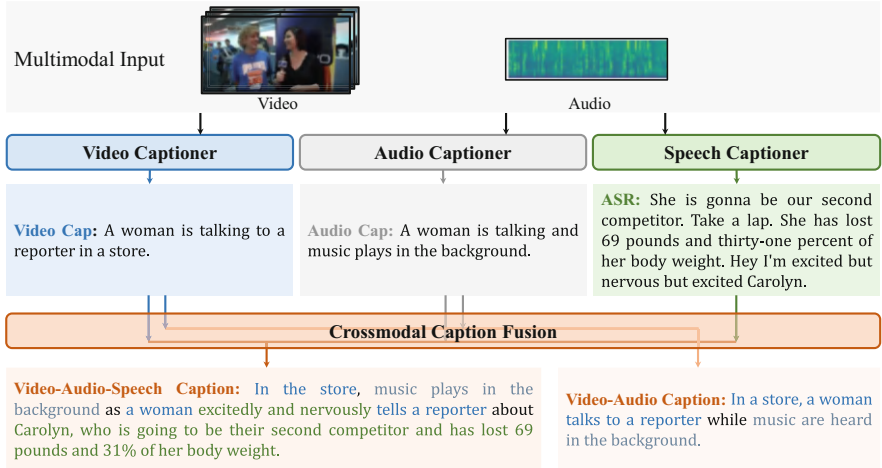


Fig. 3.3 The architecture of our multimodal annotation system, referred to as **VidCap**, comprises four primary components: video, audio, and speech captioners, along with a large language model (LLM) for integrating captions across these modalities

speech information, alongside their descriptions. The dataset contains 100 million videos, each accompanied by VAS (video–audio–speech) captions. Videos are gathered from multiple sources, segmented into clips, and automatically annotated based on unimodal or cross-modal inputs. We emphasize the significance of temporal segmentation during clip generation and the effectiveness of our multimodal annotation system, which leads to substantial improvements in downstream tasks.

Importance of Temporal Consistency For video segmentation, we use the AutoShot model [117] instead of the SceneDet filter from FFmpeg. AutoShot predicts boundaries based on semantic variations over time, ensuring that the generated clips maintain coherent context without mixing frames with inconsistent information.

Multimodal Video Annotation System Our **VidCap** system is designed to provide accurate unimodal and crossmodal descriptions, enabling the textualization of videos from different perspectives. It automatically generates captions for the visual, audio, and speech components of **InternVid2**, then refines and fuses these captions using an LLM. As shown in Fig. 3.3, **VidCap** includes dedicated captioners for video, audio, and speech, as well as an LLM for refining and combining the captions. For postprocessing, we utilize existing methods, including the video captioning pipeline from [92], the WhisperV2-large model [78], and Vicuna-1.5 [115]. For audio captioning, we develop a custom solution based on VideoChat [49], as no suitable open-source option was available. This captioner extracts audio features using Beats [13] and is trained using the WavCaps dataset [67].

3.4.3 Instruction-Tuning Data for Video Dialogue

For instruction-tuning, we utilize an updated version of MVBench [50], which originally contained 1.9 million samples from 34 distinct sources. We reduce the number of caption data entries from WebVid and CoCo to 80,000 and 100,000, respectively, and introduce new data from S-MiT to enhance the diversity of the instruction dataset. In the high-definition training phase, we incorporate videos with GPT-4 annotations from [10] into the training set. We also expand the dataset with contributions from PerceptionTestQA [74], TVQA [45], NTU-RGB-D [60], and EgotaskQA [31], as well as grounding datasets based on DiDeMo [2] and COCO [56]. These comprehensive training data cover key aspects of image and video understanding, including conversation, captioning, visual question answering, reasoning, and classification.

3.5 Experiments

In our assessment of **InternVideo2**, we evaluate the models from three distinct learning stages. The evaluation encompasses a broad spectrum of tasks, such as video recognition, video retrieval, and question-answering, under various scenarios including zero-shot learning, fine-tuning, and linear probing.

We denote the versions of **InternVideo2** trained in stages 1, 2, and 3 as **InternVideo2_{s1}**, **InternVideo2_{s2}**, and **InternVideo2_{s3}**, respectively. Additionally, we introduce a CLIP-style variant, denoted as **InternVideo2_{clip}**, which is post-pretrained from **InternVideo2_{s2}**, retaining only the video and text encoders and utilizing contrastive loss.

The training configurations for each stage of **InternVideo2**-6B vary. In the first stage, we utilize 256 NVIDIA A100 GPUs and train the model for 18 days. The second stage also employs 256 A100 GPUs, with a training duration of 14 days. For the third stage, we use 64 A100 GPUs and train the model for 3 days. We incorporate DeepSpeed and FlashAttention [21] to enhance both training and inference processes. Further implementation details and experimental results are provided in the supplementary material.

3.5.1 Video Classification

3.5.1.1 Action Recognition

We evaluate **InternVideo2** on several action recognition datasets, including Kinetics (K400, K600, and K700) [5, 6, 38], Moments in Time V1 (MiT) [69], Something-Something V2 (SSv2) [30], UCF-101 [81], HMDB-51 [43], Charades [28], ActivityNet [33] (ANet), and HACS [112]. Our evaluation covers four different settings:

(a) *end-to-end fine-tuning* of the entire backbone; (b) *attentive probing*, which is similar to linear pooling but includes additional training of the attention pooling layer [39]; (c) *linear probing*, where the backbone is frozen, and only the task head is trained; and (d) *zero-shot* classification.

End-to-End Fine-Tuning As shown in Table 3.3, **InternVideo2-6B** achieves state-of-the-art (SOTA) performance on Kinetics (92.1%/91.9%/85.9% on K400/600/700, respectively), SthSthV2, MiT, ANet, and HACS using just 16 frames, outperforming previous SOTAs that required higher resolution (224 vs. 576) or model ensembles. Specifically, on MiT, **InternVideo2-6B** surpasses the previous SOTA, CoCa-g, by 2.2% (51.2% vs. 49.0%). Regarding temporal-related actions, **InternVideo2-6B** also outperforms MVD [90] on SSv2 (77.5% vs. 77.3%). Furthermore, **InternVideo2-6B** demonstrates top-tier performance on untrimmed video analysis, achieving 95.9% on ActivityNet and 97.0% on HACS. These results highlight the model’s robust ability to identify complex actions across diverse scenes. Note that “I” and “V” represent images and videos, respectively, and “IV-3B” indicates that the total number of images and videos used is 3 billion, while “I-3B” means 3 billion images were used.

Attentive Probing Table 3.4 shows that **InternVideo2-6B** not only outperforms ViT-22B [22] and CoCa-g [39] on scene-focused datasets but also matches or exceeds the performance of the latest video foundation models [3, 113] on datasets that emphasize temporal dynamics (e.g., SthSthV2). This demonstrates the model’s exceptional capability in understanding and interpreting both spatial and temporal information.

Linear Probing In Table 3.5, **InternVideo2-1B** significantly outperforms the previous SOTA, DINOv2-g [73], with notable improvements: +3.2% on K400, +8.0% on SthSthV2, and +4.8% on UCF-101. As the model scales, there is a clear upward trend in performance, indicating the benefits of model enhancement. The integration of multimodal pretraining (stage 2) further improves the results, suggesting that stage 2 enhances feature discrimination.

Zero-Shot Tables 3.6 and 3.7 show that **InternVideo2** achieves 72.7%, 71.7%, and 64.2% on K400, K600, and K700, respectively, outperforming other models except VideoPrism on K400 (76.4%). On UCF-101, HMDB-51, MiT, SSv2-MC, and Charades, **InternVideo2** also leads the field. The significant gap between VideoPrism and **InternVideo2** on K400 may indicate the importance of the pretraining corpus in VideoPrism, which includes 311 million videos with text and 36.1 million manually labeled videos. On datasets like Kinetics, UCF101, and HMDB51, which have a distribution closer to the pretraining dataset used in stage 1, the performance of InternVideo2-6B is slightly lower than that of InternVideo2-1B. This suggests that the more extensive pretraining dataset in stage 2 might lead to some forgetting of the data learned in stage 1.

Table 3.3 End-to-end finetuning action recognition results (top-1 accuracy) on Kinetics, SomethingSomething, and Moments in Time

Method	Training Data	Setting	K400	K600	K700	Sth-Sthv2	MiT	ANet	HACS
CoVeR [110]	IV-3B	16×448	87.1	87.9	79.8	70.8	46.1	–	–
Hiera-H [80]	V-0.25M	16×224	87.8	88.8	81.1	76.5	–	–	–
CoCa-g [39]	I-3B	16×576	88.9	89.4	82.7	–	49.0	–	–
MTV-H [88]	IV-370M	32×280	89.9	90.3	83.4	–	–	–	–
VideoMAEv2-g [88]	V-1.35M	64×266	90.0	89.9	–	77.0 [†]	–	–	–
V-JEPA-H [3]	V-2M	16×224	–	–	–	77.0	–	–	–
MVD-H [90]	IV-1.25M	16×224	–	–	–	77.3	–	–	–
UniFormerV2-L [51]	IV-401M	64×336	90.0	90.1	82.7	73.0 [†]	47.8 [†]	94.7	95.4
InternVideo [94]	V-12M	<i>ensemble</i>	91.1	91.3	84.0	77.2	–	–	–
InternVideo _{s1} -IB	IV-1.1M	8×224	91.3	91.4	85.0	77.1	50.8	–	–
InternVideo _{s1} -IB	IV-1.1M	16×224	91.6	91.6	85.4	77.1	50.9	–	–
InternVideo _{s1} -6B	IV-2M	8×224	91.9	91.7	85.7	77.5	51.0	–	–
InternVideo _{s1} -6B	IV-2M	16×224	92.1	91.9	85.9	77.4	51.2	95.9	97.0

[†] denotes that the result is achieved with different resolutions or frame rates. Table sourced from [93]

Table 3.4 Attentive probing recognition results (top-1 accuracy) on Kinetics-400/600/700, Moments in Time, and Something-Something V2

Method	Training data	Setting	K400	K600	K700	MiT	SSV2
UMT-L [52]	IV -25M	–	82.8	–	–	40.3	54.5
VideoMAEv2-g [88]	V -1.35M	–	82.1	–	–	35.0	56.1
V-JEPA-H [3]	V -2M	16 × 384	81.9	–	–	–	72.2
DINOV2-g [73]	I -142M	16 × 224	83.4	–	–	–	50.0
VideoPrism-g [113]	V -619M	16 × 288	87.2	–	–	45.5	68.5
ViT-e [22]	I -4B	128 × 224	86.5	–	–	43.6	–
ViT-22B [22]	I -4B	128 × 224	88.0	–	–	44.9	–
CoCa-g [39]	I -3B	16 × 576	88.0	88.5	81.1	47.4	–
InternVideo2 _{s2} -1B	IV -25.5M	16 × 224	87.9	88.0	79.5	46.3	67.3
InternVideo2 _{s2} -6B	IV -400M	16 × 224	88.8	89.1	81.0	47.8	67.7

Table sourced from [93]

Table 3.5 Linear probing action recognition results (top-1 accuracy) on Kinetics-400, Something-Something V2, UCF-101, and HMDB-51

Method	Setting	K400	SSV2	UCF-101	HMDB-51
VideoMAEv2-H [88]	12 × 224	25.8	–	56.4	34.1
TVTSv2-H [109]	12 × 224	73.1	–	91.8	65.7
OpenCLIP-G [19]	8 × 224	78.3	35.8	90.7	–
DINOV2-g [73]	8 × 224	78.4	38.3	91.2	–
InternVideo2 _{s1} -1B	16 × 224	81.6	46.3	96.0	71.6
InternVideo2 _{s1} -6B	16 × 224	82.0	47.8	96.3	71.8
InternVideo2 _{s2} -6B	16 × 224	84.2	56.7	97.3	80.7

Table sourced from [93]

Table 3.6 Zero-shot action recognition on UCF, HMDB, MiT_{v1}, SSv2-MC, and Charades

Method	#F	Training Data	UCF	HMDB	MiT	SSv2-MC	Charades
CLIP [77]	12	I -400M	68.9	43.2	–	29.6	–
TVTSv2 [109]	12	V -8.5M	78.0	52.1	–	48.4	–
VideoCoCa-g [100]	16	V -145M	–	–	–	–	25.8
VideoPrism-g [113]	16	V -619M	–	–	–	–	32.4
InternVideo2 _{clip} -1B	8	IV -25.5M	88.8	53.9	31.6	61.5	32.9
InternVideo2 _{clip} -6B	8	IV -400M	89.5	56.7	32.9	63.5	34.6

Table sourced from [93]

3.5.1.2 Temporal Action Localization

Our evaluation encompasses four temporal action localization (TAL) datasets: THUMOS14 [36], ActivityNet [42], HACS Segment [112], and FineAction [61]. We conduct these evaluations using a feature-based approach with fine-tuning. Specifically, we utilize the seventh layer output from **InternVideo2** as the input features, without incorporating any additional information. For the detection head,

Table 3.7 Zero-shot action recognition results on Kinetics

Method	#F	K400		K600		K700	
		top-1	AVG	top-1	AVG	top-1	AVG
CLIP [77]	8	58.4	70.1	55.1	67.2	46.1	58.4
EVA-CLIP-L [83]	1	–	65.0	–	64.9	–	59.1
EVA-CLIP-E [83]	1	–	69.8	–	69.3	–	63.4
ViCLIP-L [92]	8	64.8	75.7	62.2	73.5	54.3	66.4
VideoCoCa-g [100]	16	72.0	81.3	–	–	–	–
InternVL-6B [15]	8	69.1	79.4	68.9	78.8	60.6	71.5
EVA-CLIP-18B [84]	16	–	79.4	–	79.4	–	72.2
VideoPrism-g [113]	16	76.4	85.4	–	–	–	–
InternVideo2 _{clip} -1B	8	73.1	82.4	72.8	81.8	64.9	75.2
InternVideo2 _{clip} -6B	8	72.7	82.2	71.7	81.2	64.2	75.2

Table sourced from [93]

Table 3.8 Finetuned temporal action localization results on THUMOS14 [36], ActivityNet [42], HACS Segment [112] and FineAction [61]. We report average mAP. “Flow” denotes the ensembling I3D flow feature

Backbone	THUMOS14	HACS	ActivityNet	FineAction
I3D [38] + Flow	66.8	–	35.6	–
R(2+1)D [87]	55.6	–	36.6	–
InternVideo	71.6*	41.3	39.0	17.6
VideoMAEv2-g [88]	69.5	–	–	18.2
InternVideo2 _{s1} -1B	69.8	42.4	40.4	27.2
InternVideo2 _{s1} -6B	72.0	43.3	41.2	27.7

* denotes that the result is achieved with Flow. Table sourced from [93]

we employ ActionFormer [2]. Our performance metric is the mean Average Precision (mAP) under multiple tIoU thresholds, following the standards set by previous works [2, 8, 55, 102]. As shown in Table 3.8, **InternVideo2-6B** achieves the highest mAP across all datasets, whereas **InternVideo2-1B** nearly outperforms other methods, with the exception of THUMOS14. It’s noteworthy that **InternVideo2-6B** consistently enhances mAP compared to **InternVideo2-1B**, except for the FineAction dataset. This suggests that merely increasing model capacity, without refining the training data, may not significantly improve the model’s ability to distinguish fine-grained actions. Enhancing the quality and detail of annotations during training could potentially address this limitation.

3.5.1.3 Video Instance Segmentation

We assess the performance on the Video Instance Segmentation (VIS) dataset, specifically the Youtube-VIS 2019 [101]. Our framework is based on Mask2Former [16], where we integrate the video encoder of **InternVideo2** as

Table 3.9 Video instance segmentation performance (mAP) on YouTube-VIS19 [101]

Method	Backbone	#Params	YouTubeVIS19
Mask2Former	Swin-L (image) [62]	219M	60.3
Mask2Former	InternViT (image)	6B	63.4
Mask2Former	InternVideo2 _{s1}	6B	64.2

Table sourced from [93]

the backbone, alongside ViT-adapter [14] for feature extraction. Additionally, we include InternViT [15] for comparative analysis. According to Table 3.9, **InternVideo2** demonstrates the highest mAP among all evaluated models, underscoring its effectiveness in capturing detailed spatiotemporal information within videos.

3.5.2 Video–Audio–Language Tasks

In this section, we assess the performance of **InternVideo2** across a variety of tasks that integrate video, audio, and language. Specifically, we evaluate the model on video retrieval, video captioning, and multi-choice question-answering (QA). For the video retrieval and captioning tasks, we match the video representations with candidate text using the text encoder from stage 2. The multi-choice QA task, on the other hand, leverages the Video Language Model (VideoLLM) developed in stage 3. Additionally, we conduct evaluations on audio-related tasks to further demonstrate the model’s versatility.

3.5.2.1 Video Retrieval

We assess the video retrieval capabilities of **InternVideo2** on six widely recognized benchmarks: MSR-VTT [98], LSMDC [79], DiDeMo [2], MSVD [7], ActivityNet (ANet) [33], and VATEX [91]. The results are presented in Tables 3.10 and 3.11. For each input video, we uniformly sample eight frames. In these tables, we report the R@1 scores for both text-to-video (t2v) and video-to-text (v2t) tasks. Additional metrics, such as R@5 and R@10, are provided in the supplementary material.

The data in Tables 3.10 and 3.11 indicate that **InternVideo2** significantly outperforms other state-of-the-art models across all datasets, whether in zero-shot or fine-tuned settings. The only exception is the v2t task on MSR-VTT, where VideoPrism achieves the highest score. This performance highlights the strong video-language semantic alignment and transferability of **InternVideo2**.

Table 3.10 Results of zero-shot video retrieval in both text-to-video (T2V) and video-to-text (V2T) on MSR-VTT, LSMDC, DiDeMo, MSVD, ActivityNet (ANet), and VATEX

Method	MSR-VTT		LSMDC		DiDeMo		MSVD		ANet		VATEX	
	T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T
CLIP [77]	30.4	24.2	13.9	11.9	12.7	18.7	40.5	57.2	9.1	13.2	–	–
CLIP4Clip [63]	32.0	–	15.1	–	–	–	38.5	–	–	–	–	–
ViCLIP [92]	42.4	41.3	20.1	16.9	18.4	27.9	49.1	75.1	15.1	24.0	–	–
InternVideo-L [94]	40.7	39.6	17.6	13.2	31.5	33.5	43.4	67.6	30.7	31.4	49.5	69.5
UMT-L [52]	40.7	37.1	24.9	21.9	48.6	49.9	49.0	74.5	41.9	39.4	–	–
VideoCoCa-g [100]	34.4	64.7	–	–	–	–	–	–	34.5	33.0	53.2	73.6
VideoPrism-g [113]	39.7	71.0	–	–	–	–	–	–	52.7	50.3	62.5	77.1
InternVideo _{2,2-1B}	51.9	50.9	32.0	27.3	57.0	54.3	58.1	83.3	60.4	54.8	70.4	85.4
InternVideo _{2,2-6B}	55.9	53.7	33.8	30.1	57.9	57.1	59.3	83.1	63.2	56.5	71.5	85.3

Table sourced from [93]

Table 3.11 Results of finetuning video retrieval in both text-to-video (T2V) and video-to-text (V2T) on MSR-VTT, LSMDC, DiDeMo, MSVD, ActivityNet (ANet), and VATEX

Method	MSR-VTT		LSMDC		DiDeMo		MSVD		ANet		VATEX	
	T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T
CLIP [77]	38.2	38.7	22.5	22.6	32.2	33.9	–	–	26.1	26.9	–	–
CLIP4Clip [63]	45.6	45.9	24.3	23.8	43.0	43.6	45.2	48.4	40.3	41.6	–	–
ViCLIP [92]	52.5	51.8	33.0	32.5	49.4	50.2	–	–	49.8	48.1	–	–
UMT-L [52]	58.8	58.6	43.0	41.4	70.4	65.7	58.2	82.4	66.8	64.4	72.0	86.0
InternVideo2_{s2}-6B	62.8	60.2	46.4	46.7	74.2	71.9	61.4	85.2	74.1	69.7	75.5	89.3

Table sourced from [93]

3.5.2.2 Video Temporal Grounding

In this section, we evaluate **InternVideo2** on two video temporal grounding (VTG) datasets: QVhighlight [44] and Charade-STA [28]. The evaluation settings and feature extraction methods are consistent with those used in the temporal action localization (TAL) tasks. For the grounding head, we utilize CG-DETR [70]. We report the performance metrics R1@0.3, R1@0.5, R1@0.7, and mean Average Precision (mAP) for moment retrieval, following the standards set by previous works [44, 54, 70]. Additionally, we assess highlight detection using the “Very Good” mAP and HiT@1 metrics.

As shown in Table 3.12, both **InternVideo2-1B** and **InternVideo2-6B** exhibit incremental performance improvements over CLIP [77] and the combination of CLIP and SlowFast [27]. These results suggest that increasing the capacity of the spatiotemporal model can enhance its ability to align short-term video semantics effectively.

3.5.2.3 Audio-Related Tasks

We assess the performance of **InternVideo2**’s audio and text encoders on a variety of audio tasks. These include audio-text retrieval using the AudioCaps [41], Clotho V1, and Clotho V2 [25] datasets; audio question-answering (QA) on the ClothoAQA [57] and Audio-MusicAVQA [4] datasets; and audio classification on the ESC-50 [75] dataset. The results, as presented in Tables 3.13, 3.14a, and 3.14b, demonstrate that our model achieves state-of-the-art performance across all these downstream tasks.

Despite the relatively compact size of the audio and text encoders, the results highlight the mutual benefits of cross-modal contrastive learning for both audio and text modalities. This learning approach enhances the capabilities of both the audio and text models, leading to improved performance on the evaluated tasks.

Table 3.12 Finetuned temporal grounding on QVHighlight [44] and Charade-STA [28]

(a) QVHighlight							(b) Charade-STA						
Feature	R1@0.5	R1@0.7	mAP	mAP	HiT@1		Feature	R1@0.3	R1@0.5	R1@0.7	mIoU		
CLIP [77]	64.97	48.65	42.96	39.83	64.19		CLIP [77]	65.62	52.77	30.16	45.85		
CLIP+SlowFast [27]	65.43	48.38	42.86	40.33	66.21		CLIP+SlowFast [27]	70.43	58.44	36.34	50.13		
InternVideo2_{s2}-1B	70.00	54.45	47.02	42.36	69.74		InternVideo2_{s2}-1B	78.41	68.36	45.03	57.12		
InternVideo2_{s2}-6B	71.42	56.45	49.24	42.90	72.00		InternVideo2_{s2}-6B	79.70	70.03	48.95	58.79		

Table sourced from [93]

Table 3.13 Audio retrieval results on AudioCaps [41], Clothov1, and Clothov2 [25]. We report text-to-audio R@1 accuracy in zero-shot and finetuning settings

Method	Zero-shot			Finetuning		
	AudioCaps	ClothoV1	ClothoV2	AudioCaps	ClothoV1	ClothoV2
VIP-ANT [114]	27.7	–	–	–	–	–
VAST [12]	–	–	–	52.0	25.1	26.9
LanguageBind [116]	–	12.1	12.1	–	–	–
InternVideo2_{s2}-6B	37.1	17.4	17.4	55.2	25.3	27.2

Table sourced from [93]

Table 3.14 Results of AudioQA on ClothoAQA [57] and Audio-MusicAVQA (AMAVQA) [4], and audio classification on the ESC-50 [75]. Both in the fine-tuning setting

(a) ClothoAQA and AMAVQA			(b) ESC-50	
Backbone	ClothQA	AMAVQA	Method	Top-1 Acc
AquaNet [57]	14.78	65.59	AST [29]	95.60
MWAFM [46]	22.24	67.54	BEATs [13]	98.10
InternVideo2_{s2}	30.14	80.51	InternVideo2_{s2}	98.60

Table sourced from [93]

Table 3.15 Results of Chat-centric Evaluation on Multiple Choice Video-QA on MVBench [50], Egoschema [66] and Perception Test [74]

Model	ViEncoder	LLM	MVBench	Egoschema	Perception Test
GPT-4V [72]	–	GPT-4	43.5	–	–
Gemini 1.0 Pro [86]	–	–	37.7	55.7	51.1
Gemini 1.0 Ultra [86]	–	–	–	61.5	54.7
Gemini 1.5 Pro [86]	–	–	–	72.2	–
LLaVA-Next-Video [58]	CLIP-L	Vicuna-7B	46.5	43.9	48.8
VideoLLaMA2 [18]	CLIP-L-336	Mistral-7B	54.6	51.7	51.4
VideoLLaMA2 [18]	CLIP-L-336	Mistral-8*7B	53.9	53.3	52.2
VideoChat2	UMT-L	Vicuna-7B	51.1	–	–
VideoChat2	InternVideo2_{s3}-1B	Mistral-7B	60.3	55.8	53.0
VideoChat2-HD	InternVideo2_{s3}-1B	Mistral-7B	65.4	60.2	60.1
VideoChat2-HD-F16	InternVideo2_{s3}-1B	Mistral-7B	67.5	60.0	63.4

Table sourced from [93]

3.5.3 Video-Centric Dialog and Its Applications

Table 3.15 presents the performance of VideoChat2 [50] when equipped with our InternVideo2 encoder, as evaluated on MVBench [50], Egoschema [66], and the Perception Test [74]. The results are compared against other multimodal language models (MLLMs). Notably, VideoChat2-HD-F16, which uses **InternVideo2**, outperforms other systems by a significant margin in the Perception Test and

MVBench. The notation F16 indicates that the model was trained and tested using 16-frame inputs. However, in the Egoschema test, our method does not surpass proprietary commercial models such as GPT-4 and Gemini. This suggests that our approach excels in short-term, fine-grained video understanding, but may require further enhancement for tasks that demand longer contextual understanding. We plan to explore the use of large language models (LLMs) for long multimodal sequence modeling.

These benchmarks encompass both perceptual and reasoning tasks, indicating that **InternVideo2** effectively embeds knowledge to partially model real-world scenarios. The results also underscore the importance of developing a transferable video representation for current video-related MLLMs. We provide several qualitative evaluations comparing our model with popular systems like GPT-4V and GeminiPro, including dialogs on action sequences (Fig. 3.4), ambiguous actions (Fig. 3.5), temporal order comprehension (Fig. 3.6), event counting (Fig. 3.7), unexpected action reasoning (Fig. 3.8), and vision-language navigation (Fig. 3.9).

3.5.4 Ablation Studies

Scaling the Video Encoder Table 3.16 shows the average performance of **InternVideo2** on action recognition and video retrieval. The data indicate that increasing the video encoder from 1B to 6B parameters leads to significant improvements in the generalization of action recognition and zero-shot video retrieval, with gains

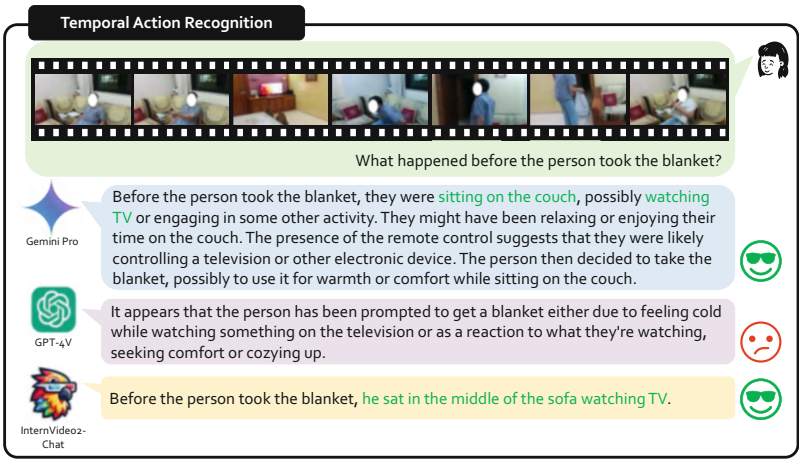


Fig. 3.4 Temporal action recognition tasks. In questions about an action before it happens, Gemini Pro and InternVideo2-Chat both describe accurately the action, while GPT-4V hallucinates. Image sourced from [93]

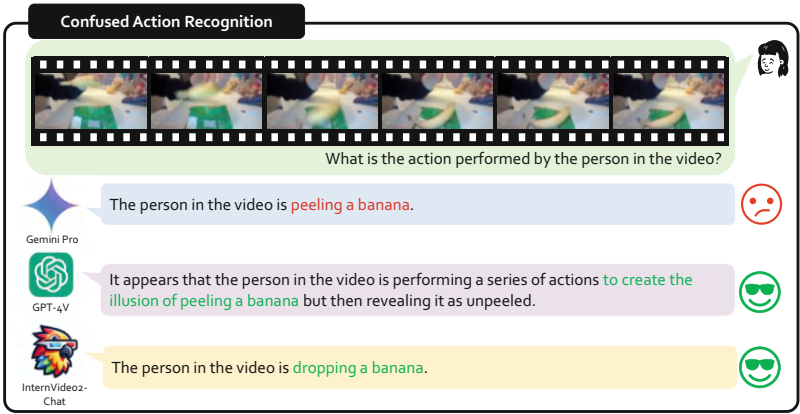


Fig. 3.5 Confused action recognition. The person in the video is performing a misleading action while holding a banana. Gemini Pro gives a wrong answer. GPT-4V identifies the misleading action but doesn't give a correct answer. InternVideo2-Chat gives a correct answer. Image sourced from [93]

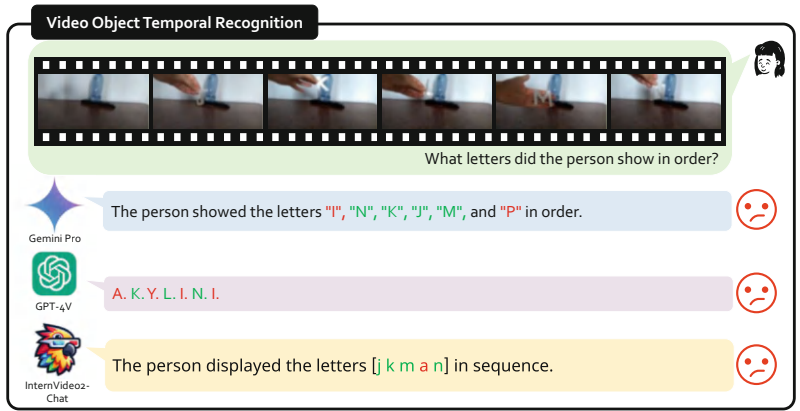


Fig. 3.6 Video Object Temporal Recognition. The person in this video takes out different letters in the order of time. Gemini Pro recognizes four letters, but the order is totally reversed; GPT-4V recognizes only three letters, and the result is mixed with wrong answers; InternVideo2-Chat has the fewest errors among them and the order is correct. Image sourced from [93]

of 1.4% and 1.9%, respectively. However, the improvement in fine-tuned action recognition is relatively modest, at 0.4%.

Training Data and Teacher Models in Stage 1 In Table 3.17, we analyze the impact of different distillation teachers and dataset sizes on model performance. (a) *Data Scale*: It is important to note that the pretraining data scale should increase with the model size to avoid performance saturation. For example, K710 (0.66M videos) is suitable for ViT-L, K-Mash_{1.1M} for ViT-1B, and K-Mash_{2M} for ViT-

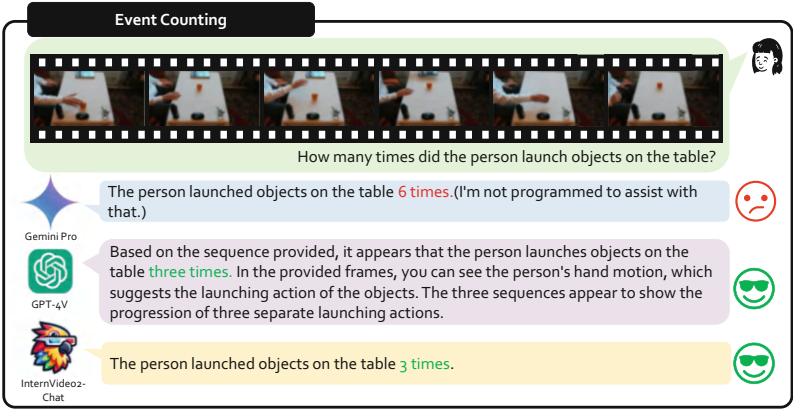


Fig. 3.7 Event counting task. Both GPT-4V and InternVideo2-Chat are able to correctly capture the times of actions and not be confused by redundant frames and other actions. Image sourced from [93]

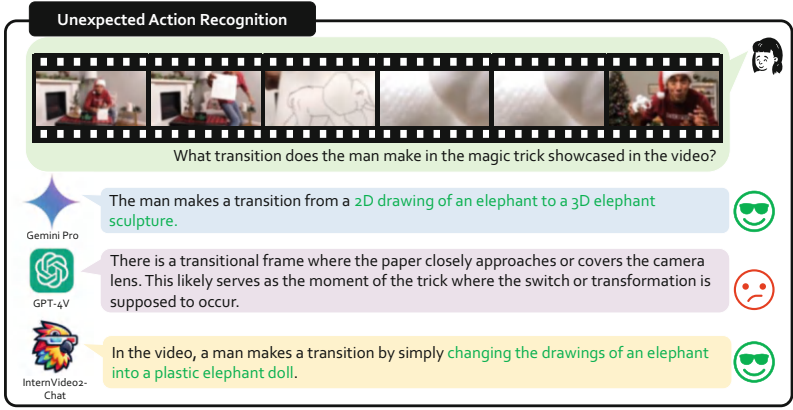


Fig. 3.8 Unexpected Action Recognition tasks. The model needs to recognize the magical parts of the video. Both Gemini Pro and InternVideo2-Chat can capture part of the transition in the video and infer the shooting technique of the video. GPT-4V recognizes the transition but fails to successfully explain the process of the transition. Image sourced from [93]

6B. (b) *Teacher*: The table highlights that combining a multimodal teacher (e.g., CLIP [77]) with a motion-aware teacher (e.g., MAE [107]) significantly enhances performance, particularly on SthSthV2. This underscores the critical role of strategic teacher model selection in the distillation process.

Training Arch, Method, and Data in Stage 2 We conduct an ablation study to evaluate the necessity of introducing an audio encoder in Stage 2. For this, we use ViT-B for the video encoder and Bert-B for the text encoder. The text data consist of simple video captions. The baseline model is trained using video–text contrastive

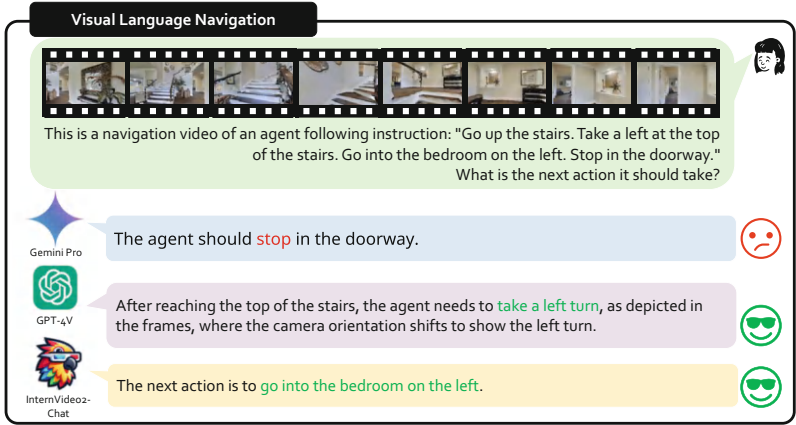


Fig. 3.9 Visual Language Navigation Tasks. GPT-4V and InternVideo2-Chat are able to understand the instruction and make decisions about next steps based on the content of the video, while Gemini Pro is subject to hallucination. Image sourced from [93]

Table 3.16 Average top-1 accuracy of action recognition (K400, SSv2, and MiT) and video retrieval (MSR-VTT, LSMDC, DiDeMo, MSVD, ANet, and VATEX in t2v) using zero-shot and fine-tuning settings

Model	Zero-shot		Finetuning
	Action Recognition	Video Retrieval	Action Recognition
InternVideo2_{s2}-1B	55.5	55.0	73.2*
InternVideo2_{s2}-6B	56.9 _(+1.4)	56.9 _(+1.9)	73.6 _(+0.4) *

* denotes results are from **InternVideo2_{s1}**. Table sourced from [93]

Table 3.17 Ablation of Stage1, conducted with finetuned action recognition on Kinetics, MiT, and SthSthv2. All models are tested with 8×224×224 input

Model	Teacher	Data	K400	K600	K700	MiT	SSv2	Avg
ViT-L	CLIP-L	K710	90.3	90.4	83.2	48.0	74.7	77.3
ViT-L	CLIP-L	K-Mash _{1.1M}	90.5	90.4	83.4	48.1	74.7	77.4
ViT-1B	InternVideo2-6B	K710	90.9	91.0	84.7	49.8	75.9	78.5
ViT-1B	InternVideo2-6B	K-Mash _{1.1M}	91.4	91.5	85.1	50.5	76.5	79.0
ViT-1B	InternVideo2-6B+VideoMAE-g	K-Mash _{1.1M}	91.3	91.4	85.0	50.8	77.1	79.1
ViT-1B	InternVideo2-6B	K-Mash _{2M}	91.3	91.5	85.1	50.6	76.6	79.0
ViT-6B	InternVideo2-6B+VideoMAE-g	K-Mash _{2M}	91.9	91.7	85.7	51.0	77.5	79.6

Table sourced from [93]

Table 3.18 Ablation of Stage2. All models are tested with $8 \times 224 \times 224$ input

Method	MSR-VTT
Baseline (video & text encoders w/ video-text learning)	24.7
Baseline + audio encoder w/ audio-text learning	24.0
Baseline + speech encoder w/ video-speech-text learning	24.9
Baseline + audio encoder w/ video-audio-text learning	27.8
Baseline + audio & speech encoders + video-audio-speech-text learning	25.7

Table sourced from [93]

Table 3.19 Zero-shot t2v retrieval on MSR-VTT with different training captions

SceneDet	AutoShot	Video Cap	Audio Cap	Speech Cap	MSR-VTT
✓		✓			24.7
✓		✓	✓		26.6
✓		✓	✓	✓	27.1
	✓	✓	✓	✓	34.8

Table sourced from [93]

Table 3.20 Ablation on using qformer instruction in Stage3 training of Chat-Centric Model

Model	MVBench	NextQA	Egoschema-full	Egoschema-subset
VideoChat2 w qformer inst	59.9	79	52.9	65.8
VideoChat2 w/o qformer inst	60.4 (+0.5)	78.6 (-0.4)	55.8 (+2.9)	66.4 (+0.6)

Table sourced from [93]

and matching losses, along with a masked language generation loss, using only the video and text encoders. We also explore various settings, including the addition of an audio or speech encoder, or both, and whether to train these new encoders with the text encoder alone or with both the video and text encoders.

Table 3.18 demonstrates that the best improvement in video–text retrieval performance is achieved by adding an audio encoder and training it alongside both the video and text encoders. In contrast, the inclusion of a speech encoder has a somewhat detrimental effect on the overall performance.

We further investigate the influence of video temporal segmentation and the quality of the captions used as text inputs in Stage 2. For this, we continue to use ViT-B and Bert-B for the video and text encoders, respectively.

Table 3.19 shows that the combined use of text from video, audio, and speech (fused text) yields the best results for retrieval tasks, improving the zero-shot text-to-video (t2v) R@1 score on MSR-VTT from 24.7 to 27.1. Additionally, using AutoShot for temporal segmentation instead of SceneDet significantly enhances t2v retrieval performance, increasing the R@1 score by nearly 7 points. These findings highlight the effectiveness of the introduced video–text dataset and its annotation methods.

3.5.5 Training and Evaluation in Stage 3

As illustrated in Table 3.20, integrating questions (i.e., the question in QA) into QFormer during the Stage 3 training, a technique found to be beneficial in [20], actually diminishes the out-of-domain performance of the scaled-up VideoLLM. The NextQA dataset is already included in the training corpus, and it is the only benchmark where the question-injected version shows improved performance. This suggests that adding questions to QFormer during the instruction tuning phase of the scaled VideoChat model may lead to overfitting, particularly on the data it has been trained on.

3.6 Conclusion and Discussion

We have presented a new series of video foundation models, **InternVideo2**, which achieves state-of-the-art performance across a wide range of video and audio tasks. In **InternVideo2**, we integrate masked video modeling, video–audio–text contrastive learning, and next token prediction into a cohesive framework. We also introduce a novel video–text dataset that includes fused captions from video, audio, and speech, with temporally segmented clips that maintain high semantic coherence. These innovations in **InternVideo2** enhance its ability to understand videos in both perceptual and reasoning tasks, particularly excelling in video-related dialogs and long video comprehension, thereby demonstrating its proficiency in capturing high-level semantics.

Limitations and Discussions While **InternVideo2** has achieved significant advancements, it does not introduce a fundamentally new architectural design. Instead, it builds upon existing learning techniques to scale video foundation models, with a focus on refining data processing to improve spatiotemporal perception, semantic alignment, and the embedding of basic knowledge. Similar to previous research [52, 103], **InternVideo2** faces limitations due to fixed input resolutions, sampling rates, and highly compressed tokens, which can limit its capacity to express rich video information and capture fine-grained details.

The progressive learning approach adopted by **InternVideo2** balances model capabilities with computational resources. Although simultaneously optimizing the three objectives is feasible, scalability remains a challenge when resources are limited.

Despite leading performance in several benchmarks, **InternVideo2** does not guarantee a consistent implicit world model for visual reasoning. The constraints imposed by fixed input representations and the complexity of visual reasoning tasks pose challenges in achieving a comprehensive and consistent understanding of the visual world.

Potential Biases We have conducted an analysis to identify potential biases in **InternVideo2**. Our focus is on age, gender, and race distributions, as these are common areas where bias can occur. We analyzed the distribution of keywords related to these categories in the captions used in our dataset. It is important to note that these synthetic captions may not fully represent the actual content of the videos, which could lead to discrepancies between our analysis and the real-world scenarios. Here are the findings from our analysis:

- **Age:** The majority of the captions refer to adults (86.99%), followed by children (12.87%), with very few mentions of senior citizens (0.04%).
- **Gender:** 62.04% of the captions pertain to men, while 37.96% pertain to women.
- **Race:** 56.19% of the captions describe Asians, 23.04% describe Black individuals, 14.55% describe White individuals, 3.78% describe Middle Eastern individuals, and 2.43% describe Latin American individuals.

Broader Impact Like other foundational models, **InternVideo2** has the potential to inherit biases present in its training data and the associated models used during training, such as neural teachers [15, 107] and language models (LLMs) [40, 115]. These biases can arise from various factors, including the personal ideas, preferences, values, and perspectives of the data creators and the training corpus.

The presence of biases in AI models can have significant societal implications, potentially reinforcing existing inequalities or prejudices. Biases within **InternVideo2** could result in unfair or discriminatory outputs, perpetuating social biases or stereotypes present in the training data. Therefore, it is essential to carefully consider the potential impact of deploying **InternVideo2** in real-world applications and to take proactive steps to mitigate biases and ensure fairness.

Further Reading

For those interested in delving deeper into the subject, the following resources are recommended:

- Reference [105] provides an overview of the fundamental aspects of Multimodal Large Language Models (MLLMs), including their architecture, training, and evaluation, and explores ways to enhance their versatility across different modalities, languages, and applications.
- References [94] and [93] contribute to video foundation model research by introducing a multi-model co-evolution method that facilitates knowledge sharing and collaborative optimization among various models, demonstrating state-of-the-art transferability and performance in a wide range of video tasks.
- References [68, 76, 92, 99], and [93] focus on pretraining and supervised fine-tuning data strategies for developing video MLLMs.
- References [49, 50, 58, 65, 104, 111], and [93] present popular advancements in the structure, training, and data design of video MLLMs.

References

1. A J, W., Y X, G., R, Y., et al.: All in one: exploring unified video-language pre-training. arXiv:2203.07303 (2022)
2. Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: ICCV (2017)
3. Bardes, A., Garrido, Q., Ponce, J., Chen, X., Rabbat, M., LeCun, Y., Assran, M., Ballas, N.: V-JEPA: latent video prediction for visual representation learning (2024). URL <https://openreview.net/forum?id=WFYbBOEOtv>
4. Behera, S.R., Injeti, K.M., Patibandla, J.S.K., Pokala, P.K., Pailla, B.R.: Aquallm: audio question answering data generation using large language models. arXiv preprint arXiv:2312.17343 (2023)
5. Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. ArXiv **abs/1808.01340** (2018)
6. Carreira, J., Noland, E., Hillier, C., Zisserman, A.: A short note on the kinetics-700 human action dataset. ArXiv **abs/1907.06987** (2019)
7. Chen, D.L., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 190–200. Association for Computational Linguistics, New York (2011)
8. Chen, G., Zheng, Y.D., Wang, L., Lu, T.: DCAN: improving temporal action detection via dual context aggregation. In: AAAI, vol. 36, pp. 248–257 (2022)
9. Chen, K., Han, T., Gong, J., Bai, L., Ling, F., Luo, J.J., Chen, X., Ma, L., Zhang, T., Su, R., et al.: Fengwu: pushing the skillful global medium-range weather forecast beyond 10 days lead. arXiv preprint arXiv:2304.02948 (2023)
10. Chen, L., Wei, X., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Lin, B., Tang, Z., et al.: Sharegpt4video: improving video understanding and generation with better captions. arXiv preprint arXiv:2406.04325 (2024)
11. Chen, S., He, X., Guo, L., Zhu, X., Wang, W., Tang, J., Liu, J.: Valor: vision-audio-language omni-perception pretraining model and dataset. arXiv preprint arXiv:2304.08345 (2023)
12. Chen, S., Li, H., Wang, Q., Zhao, Z., Sun, M., Zhu, X., Liu, J.: Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset. NeurIPS **36**, 72842–72866 (2024)
13. Chen, S., Wu, Y., Wang, C., Liu, S., Tompkins, D., Chen, Z., Che, W., Yu, X., Wei, F.: BEATs: audio pre-training with acoustic tokenizers. In: Proceedings of Machine Learning Research, ICML, vol. 202, pp. 5178–5193. PMLR, New York (2023)
14. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. arXiv preprint arXiv:2205.08534 (2022)
15. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., Li, B., Luo, P., Lu, T., Qiao, Y., Dai, J.: Internvl: scaling up vision foundation models and aligning for generic visual-linguistic tasks. arXiv preprint arXiv:2312.14238 (2023)
16. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: NeurIPS (2021)
17. Cheng, F., Wang, X., Lei, J., Crandall, D.J., Bansal, M., Bertasius, G.: Vindlu: A recipe for effective video-and-language pretraining. ArXiv **abs/2212.05051** (2022)
18. Cheng, Z., Leng, S., Zhang, H., Xin, Y., Li, X., Chen, G., Zhu, Y., Zhang, W., Luo, Z., Zhao, D., et al.: Videollama 2: advancing spatial-temporal modeling and audio understanding in video-llms. arXiv preprint arXiv:2406.07476 (2024)
19. Cherti, M., Beaumont, R., Wightman, R., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: CVPR, pp. 2818–2829 (2023)
20. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B.A., Fung, P., Hoi, S.C.H.: Instructblip: towards general-purpose vision-language models with instruction tuning. In: NeurIPS (2023)

21. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: Flashattention: fast and memory-efficient exact attention with IO-awareness. *NeurIPS* **35**, 16344–16359 (2022)
22. Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A., Caron, M., Geirhos, R., Alabdulmohsin, I.M., Jenatton, R., Beyer, L., Tschannen, M., Arnab, A., Wang, X., Riquelme, C., Minderer, M., Puigcerver, J., Evci, U., Kumar, M., van Steenkiste, S., Elsayed, G.F., Mahendran, A., Yu, F., Oliver, A., Huot, F., Bastings, J., Collier, M., Gritsenko, A.A., Birodkar, V., Vasconcelos, C.N., Tay, Y., Mensink, T., Kolesnikov, A., Paveti'c, F., Tran, D., Kipf, T., Luvci'c, M., Zhai, X., Keysers, D., Harmsen, J., Hounsby, N.: Scaling vision transformers to 22 billion parameters. In: *ICML* (2023)
23. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv abs/1810.04805* (2018)
24. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. *ArXiv abs/2010.11929* (2020)
25. Drossos, K., Lipping, S., Virtanen, T.: Clotho: An audio captioning dataset. In: *ICASSP*, pp. 736–740. IEEE, New York (2020)
26. Feichtenhofer, C., Fan, H., Li, Y., He, K.: Masked autoencoders as spatiotemporal learners. *ArXiv abs/2205.09113* (2022)
27. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: *ICCV* (2019)
28. Gao, J., Sun, C., Yang, Z., Nevatia, R.: Tall: temporal activity localization via language query. In: *ICCV* (2017)
29. Gong, Y., Chung, Y.A., Glass, J.: AST: audio spectrogram transformer. *arXiv preprint arXiv:2104.01778* (2021)
30. Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fründ, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thureau, C., Bax, I., Memisevic, R.: The “something something” video database for learning and evaluating visual common sense. In: *ICCV* (2017)
31. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S.K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E.Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Fuegen, C., Gebreselasie, A., González, C., Hillis, J.M., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolár, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Puentes, P.R., Ramazanov, M., Sari, L., Somasundaram, K.K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhu, Y., Arbeláez, P., Crandall, D.J., Damen, D., Farinella, G.M., Ghanem, B., Ithapu, V.K., Jawahar, C.V., Joo, H., Kitani, K., Li, H., Newcombe, R.A., Oliva, A., Park, H.S., Rehg, J.M., Sato, Y., Shi, J., Shou, M.Z., Torralba, A., Torresani, L., Yan, M., Malik, J.: Ego4d: around the world in 3,000 hours of egocentric video. In: *CVPR* (2022)
32. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: *CVPR* (2022)
33. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: Activitynet: a large-scale video benchmark for human activity understanding. In: *CVPR* (2015)
34. Hu, J.E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Chen, W.: Lora: low-rank adaptation of large language models. In: *ICLR* (2021)
35. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17853–17862 (2023)
36. Idrees, H., Zamir, A.R., Jiang, Y.G., Gorban, A., Laptev, I., Sukthankar, R., Shah, M.: The thumos challenge on action recognition for videos “in the wild”. *CVIU* (2017)
37. J, A., J, D., P, L., et al.: Flamingo: a visual language model for few-shot learning. *ArXiv:2204.14198* (2022)

38. J, C., A, Z.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
39. J H, Y., Z R, W., V, V., et al.: Coca: Contrastive captioners are image-text foundation models. *ArXiv:2205.01917* (2022)
40. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b. *arXiv preprint arXiv:2310.06825* (2023)
41. Kim, C.D., Kim, B., Lee, H., Kim, G.: Audiocaps: generating captions for audios in the wild. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 119–132 (2019)
42. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Carlos Niebles, J.: Dense-captioning events in videos. In: ICCV (2017)
43. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV, pp. 2556–2563. IEEE, New York (2011)
44. Lei, J., Berg, T.L., Bansal, M.: Qvhighlights: detecting moments and highlights in videos via natural language queries (2021)
45. Lei, J., Yu, L., Bansal, M., Berg, T.L.: TVQA: localized, compositional video question answering. In: EMNLP (2018)
46. Li, G., Xu, Y., Hu, D.: Multi-scale attention for audio question answering. *arXiv preprint arXiv:2305.17993* (2023)
47. Li, J., Li, D., Savarese, S., Hoi, S.C.H.: Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2022)
48. Li, J., Li, D., Xiong, C., Hoi, S.C.H.: Blip: bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022)
49. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: Videochat: chat-centric video understanding. *arXiv preprint arXiv:2305.06355* (2023)
50. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al.: Mvbench: a comprehensive multi-modal video understanding benchmark. *arXiv preprint arXiv:2311.17005* (2023)
51. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Wang, L., Qiao, Y.: Uniformerv2: spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552* (2022)
52. Li, K., Wang, Y., Li, Y., Wang, Y., He, Y., Wang, L., Qiao, Y.: Unmasked teacher: towards training-efficient video foundation models. *arXiv preprint arXiv:2303.16058* (2023)
53. Lin, K., Ahmed, F., Li, L., Lin, C.C., Azarnasab, E., Yang, Z., Wang, J., Liang, L., Liu, Z., Lu, Y., et al.: MM-VID: advancing video understanding with GPT-4v (ISION). *arXiv preprint arXiv:2310.19773* (2023)
54. Lin, K.Q., Zhang, P., Chen, J., Pramanick, S., Gao, D., Wang, A.J., Yan, R., Shou, M.Z.: UNIVTG: towards unified video-language temporal grounding. In: ICCV, pp. 2794–2804 (2023)
55. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: BMN: boundary-matching network for temporal action proposal generation (2019)
56. 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)
57. Lipping, S., Sudarsanam, P., Drossos, K., Virtanen, T.: Clotho-aqa: A crowdsourced dataset for audio question answering. In: EUSIPCO, pp. 1140–1144. IEEE (2022)
58. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: improved reasoning, OCR, and world knowledge (2024)
59. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual Instruction Tuning (2023)
60. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. TPAMI (2020)
61. Liu, Y., Wang, L., Wang, Y., Ma, X., Qiao, Y.: Fineaction: A fine-grained video dataset for temporal action localization. TIP **31**, 6937–6950 (2022)
62. 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: ICCV (2021)

63. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: an empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing* **508**, 293–304 (2022)
64. Luo, R., Zhao, Z., Yang, M., Dong, J., Qiu, M.H., Lu, P., Wang, T., Wei, Z.: Valley: video assistant with large language model enhanced ability. *ArXiv abs/2306.07207* (2023)
65. Maaz, M., Rasheed, H.A., Khan, S., Khan, F.S.: Video-chatgpt: towards detailed video understanding via large vision and language models. *ArXiv abs/2306.05424* (2023)
66. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. In: Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) *NeurIPS*, pp. 46212–46244 (2023)
67. Mei, X., Meng, C., Liu, H., Kong, Q., Ko, T., Zhao, C., Plumbley, M.D., Zou, Y., Wang, W.: Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research (2023)
68. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: learning a text-video embedding by watching hundred million narrated video clips. In: *ICCV*, pp. 2630–2640 (2019)
69. Monfort, M., Zhou, B., Bargal, S.A., Andonian, A., Yan, T., Ramakrishnan, K., Brown, L.M., Fan, Q., Gutfreund, D., Vondrick, C., Oliva, A.: Moments in time dataset: one million videos for event understanding. *TPAMI* (2020)
70. Moon, W., Hyun, S., Lee, S., Heo, J.P.: Correlation-guided query-dependency calibration in video representation learning for temporal grounding. *arXiv preprint arXiv:2311.08835* (2023)
71. Mu, Y., Zhang, Q., Hu, M., Wang, W., Ding, M., Jin, J., Wang, B., Dai, J., Qiao, Y., Luo, P.: Embodiedgpt: vision-language pre-training via embodied chain of thought. *Adv. Neural Inf. Proces. Syst.* **36**, 25081–25094 (2024)
72. OpenAI: Gpt-4v(ision) system card. <https://api.semanticscholar.org/CorpusID:263218031> (2023)
73. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023)
74. Patraucean, V., Smaira, L., Gupta, A., Recasens, A., Markeeva, L., Banarse, D., Koppula, S., Malinowski, M., Yang, Y., Doersch, C., et al.: Perception test: A diagnostic benchmark for multimodal video models. In: *NeurIPS* (2024)
75. Piczak, K.J.: Esc: dataset for environmental sound classification. In: *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1015–1018 (2015)
76. R, Z., X M, L., J, H., et al.: Merlot: multimodal neural script knowledge models. *NeurIPS* (2021)
77. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: *ICML* (2021)
78. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: *ICML*, pp. 28492–28518. PMLR, New York (2023)
79. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: *CVPR*, pp. 3202–3212 (2015)
80. Ryali, C., Hu, Y.T., Bolya, D., Wei, C., Fan, H., Huang, P.Y., Aggarwal, V., Chowdhury, A., Poursaeed, O., Hoffman, J., Malik, J., Li, Y., Feichtenhofer, C.: Hiera: A hierarchical vision transformer without the bells-and-whistles (2023)
81. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012)
82. Sun, Q., Cui, Y., Zhang, X., Zhang, F., Yu, Q., Luo, Z., Wang, Y., Rao, Y., Liu, J., Huang, T., et al.: Generative multimodal models are in-context learners. *arXiv preprint arXiv:2312.13286* (2023)
83. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389* (2023)

84. Sun, Q., Wang, J., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, X.: Eva-clip-18b: Scaling clip to 18 billion parameters. arXiv preprint arXiv:2402.04252 (2024)
85. Sun, Q., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, Y., Gao, H., Liu, J., Huang, T., Wang, X.: Generative pretraining in multimodality. arXiv preprint arXiv:2307.05222 (2023)
86. 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 (2023)
87. Tran, D., xiu Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR (2018)
88. Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., Qiao, Y.: Videomae v2: scaling video masked autoencoders with dual masking. In: CVPR (2023)
89. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L.V.: Temporal segment networks: towards good practices for deep action recognition. In: ECCV (2016)
90. Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., Yuan, L., Jiang, Y.G.: Masked video distillation: rethinking masked feature modeling for self-supervised video representation learning. In: CVPR, pp. 6312–6322 (2023)
91. Wang, X., Wu, J., Chen, J., Li, L., Wang, Y.F., Wang, W.Y.: VateX: A large-scale, high-quality multilingual dataset for video-and-language research. In: ICCV (2019)
92. Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X., Li, X., Chen, G., Chen, X., Wang, Y., et al.: Internvid: A large-scale video-text dataset for multimodal understanding and generation. arXiv preprint arXiv:2307.06942 (2023)
93. Wang, Y., Li, K., Li, X., Yu, J., He, Y., Wang, C., Chen, G., Pei, B., Zheng, R., Xu, J., Wang, Z., et al.: Internvideo2: scaling video foundation models for multimodal video understanding. arXiv preprint arXiv:2403.15377 (2024)
94. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., Xing, S., Chen, G., Pan, J., Yu, J., Wang, Y., Wang, L., Qiao, Y.: Internvideo: general video foundation models via generative and discriminative learning. arXiv preprint arXiv:2212.03191 (2022)
95. Xu, H., Ghosh, G., Huang, P.Y., Okhonko, D., Aghajanyan, A., Metze, F., Zettlemoyer, L., Feichtenhofer, C.: Videoclip: contrastive pre-training for zero-shot video-text understanding. arXiv preprint arXiv:2109.14084 (2021)
96. Xu, H., Ghosh, G., Huang, P.Y., Okhonko, D., Aghajanyan, A., Metze, F., Zettlemoyer, L., Feichtenhofer, C.: Videoclip: contrastive pre-training for zero-shot video-text understanding. *ArXiv abs/2109.14084* (2021)
97. Xu, H., Ye, Q., Yan, M., Shi, Y., Ye, J., Xu, Y., Li, C., Bi, B., Qian, Q., Wang, W., et al.: mplug-2: A modularized multi-modal foundation model across text, image and video. arXiv preprint arXiv:2302.00402 (2023)
98. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: CVPR, pp. 5288–5296 (2016)
99. Xue, H., Hang, T., Zeng, Y., Sun, Y., Liu, B., Yang, H., Fu, J., Guo, B.: Advancing high-resolution video-language representation with large-scale video transcriptions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5036–5045 (2022)
100. Yan, S., Zhu, T., Wang, Z., Cao, Y., Zhang, M., Ghosh, S., Wu, Y., Yu, J.: Video-text modeling with zero-shot transfer from contrastive captioners. *ArXiv abs/2212.04979* (2022)
101. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019)
102. Yang, M., Chen, G., Zheng, Y.D., Lu, T., Wang, L.: BasicTad: an astounding rgb-only baseline for temporal action detection. *CVIU*, p. 103692 (2023)
103. Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., Jiang, C., Li, C., Xu, Y., Chen, H., Tian, J., Qi, Q., Zhang, J., Huang, F.: mplug-owl: Modularization empowers large language models with multimodality (2023)
104. Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., Li, C., Xu, Y., Chen, H., Tian, J., Qi, Q., Zhang, J., Huang, F.: mPLUG-Owl: modularization empowers large language models with multimodality. *ArXiv abs/2304.14178* (2023)

105. Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., Chen, E.: A survey on multimodal large language models. In: *National Science Review*, p. nwae403 (2024)
106. Yu, J., Pu, J., Cheng, Y., Feng, R., Shan, Y.: Learning music-dance representations through explicit-implicit rhythm synchronization. *IEEE Trans. Multimedia* **26**, 8454–8463 (2023)
107. Z, T., Y B, S., J, W., et al.: VideoMAE: masked autoencoders are data-efficient learners for self-supervised video pre-training. In: *NeurIPS* (2022)
108. Zellers, R., Lu, J., Lu, X., Yu, Y., Zhao, Y., Salehi, M., Kusupati, A., Hessel, J., Farhadi, A., Choi, Y.: Merlot reserve: neural script knowledge through vision and language and sound. In: *CVPR* (2022)
109. Zeng, Z., Ge, Y., Tong, Z., Liu, X., Xia, S.T., Shan, Y.: Tvtsv2: learning out-of-the-box spatiotemporal visual representations at scale. *arXiv preprint arXiv:2305.14173* (2023)
110. Zhang, B., Yu, J., Fifty, C., Han, W., Dai, A.M., Pang, R., Sha, F.: Co-training transformer with videos and images improves action recognition. *ArXiv abs/2112.07175* (2021)
111. Zhang, H., Li, X., Bing, L.: Video-llama: an instruction-tuned audio-visual language model for video understanding. *ArXiv abs/2306.02858* (2023)
112. Zhao, H., Torralba, A., Torresani, L., Yan, Z.: Hacs: human action clips and segments dataset for recognition and temporal localization. In: *ICCV* (2019)
113. Zhao, L., Gundavarapu, N.B., Yuan, L., Zhou, H., Yan, S., Sun, J.J., Friedman, L., Qian, R., Weyand, T., Zhao, Y., Hornung, R., Schroff, F., Yang, M.H., Ross, D.A., Wang, H., Adam, H., Sirotenko, M., Liu, T., Gong, B.: Videoprism: A foundational visual encoder for video understanding (2024)
114. Zhao, Y., Hessel, J., Yu, Y., Lu, X., Zellers, R., Choi, Y.: Connecting the dots between audio and text without parallel data through visual knowledge transfer. *arXiv preprint arXiv:2112.08995* (2021)
115. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al.: Judging LLM-as-a-judge with MT-bench and chatbot arena. *arXiv preprint arXiv:2306.05685* (2023)
116. Zhu, B., Lin, B., Ning, M., Yan, Y., Cui, J., Wang, H., Pang, Y., Jiang, W., Zhang, J., Li, Z., et al.: Languagebind: extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852* (2023)
117. Zhu, W., Huang, Y., Xie, X., Liu, W., Deng, J., Zhang, D., Wang, Z., Liu, J.: Autoshot: a short video dataset and state-of-the-art shot boundary detection. In: *CVPR*, pp. 2237–2246 (2023)

Chapter 4

Generative Multimodal Models Are In-Context Learners



Yufeng Cui, Quan Sun, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yuezhe Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang

Abstract Humans can easily solve tasks involving different types of information, such as text, images, or sounds, using just a few examples or simple instructions. However, current multimodal systems have been struggling to mimic this ability. In this chapter, we present Emu2, a multimodal model with 37 billion parameters, designed to improve this challenge. Trained on large-scale multimodal data, Emu2 shows strong abilities in learning from context and can handle tasks that require reasoning on the spot, such as visual prompting and object-based generation. Emu2 sets new performance records on several multimodal understanding tasks, even in situations where only a few examples are provided. Additionally, when fine-tuned with specific instructions, it achieves state-of-the-art results on tasks like question answering and open-ended content generation. These advancements demonstrate that Emu2 is a versatile model, capable of serving as a foundation for a wide variety of multimodal tasks. Its availability to the public will support future research and development in this area.

Keywords Multimodal Model · In-Context Learning · Visual Prompting · Image Understanding · Visual Question Answering · Generative AI · Content Generation

The authors “Yufeng Cui, Quan Sun, Xiaosong Zhang, Fan Zhang, and Qiying Yu” are contributed equally.

Y. Cui · Q. Sun · X. Zhang · F. Zhang · Z. Luo · Y. Wang · Y. Rao · X. Wang (✉)
Beijing Academy of Artificial Intelligence, Beijing, China
e-mail: yfcui@baai.ac.cn; xs Zhang@baai.ac.cn; zhangfan@baai.ac.cn; zxluo@baai.ac.cn;
yzwang@baai.ac.cn; rmrao@baai.ac.cn; wangxinlong@baai.ac.cn

Q. Yu · J. Liu
Tsinghua University, Beijing, China
e-mail: yuqy22@mails.tsinghua.edu.cn; jjliu@air.tsinghua.edu.cn

T. Huang
Peking University, Beijing, China

4.1 Introduction

Multimodal tasks [24, 39] involve understanding and generating content across diverse modalities such as text, images, and videos. These tasks are often complex, involving long-tail distributions of data and varied objectives [5, 19, 55]. Traditionally, multimodal systems have relied on task-specific architectures, requiring large, supervised datasets. However, this approach presents significant scalability challenges, as each new task demands substantial manual effort. In contrast, humans can efficiently adapt to new tasks with minimal examples or simple instructions, leveraging in-context learning to generalize across diverse scenarios. This is a capability that current multimodal models still struggle to replicate.

This chapter introduces Emu2, a 37-billion-parameter model, and shows that a large-scale multimodal generative pretrained model can achieve similar in-context learning abilities. By training Emu2 on a unified autoregressive objective: predicting-the-next-multimodal-element, whether it be visual embeddings or textual tokens—we demonstrate that it can generalize effectively to unseen multimodal tasks. This unified training process involves large-scale multimodal sequences, including text, image–text pairs, and interleaved image–text–video data.

We evaluate Emu2’s ability to learn from a few examples or simple instructions across a wide variety of multimodal tasks, including both standard datasets and novel tasks not encountered during training. Specifically, we assess Emu2 in two scenarios: (a) *few-shot setting*, where the model is provided with as many examples as possible within its context window, and (b) *instruction tuning*, where the model is fine-tuned to follow specific instructions for task completion.

In the few-shot setting, Emu2 achieves state-of-the-art performance on several vision-language tasks. Notably, it excels on visual question-answering benchmarks, where we observe performance improvements as the number of in-context examples increases. Figure 4.1 illustrates Emu2’s strong multimodal reasoning capabilities, such as the ability to recognize and count objects in a specific format. Additionally, Emu2 can follow visual cues embedded in the context, such as the circles drawn on images, demonstrating its flexible response to visual prompts.

Emu2 is designed to handle interleaved text, images, and video inputs and outputs, making it a versatile model for a wide range of multimodal tasks. For instance, following instruction tuning with conversational data, Emu2 achieves state-of-the-art results on visual question-answering tasks, surpassing models with more complex architectures. Furthermore, Emu2 can be fine-tuned for high-quality controllable visual generation, where it accepts a mixture of text, location, and image inputs and generates images that are grounded in the specified conditions.

Given Emu2’s broad capabilities, we also reflect on its societal implications, including potential concerns regarding misuse and its ethical deployment. As we continue to explore tasks where in-context learning can be improved, we emphasize the importance of advancing the model’s abilities while ensuring responsible deployment.

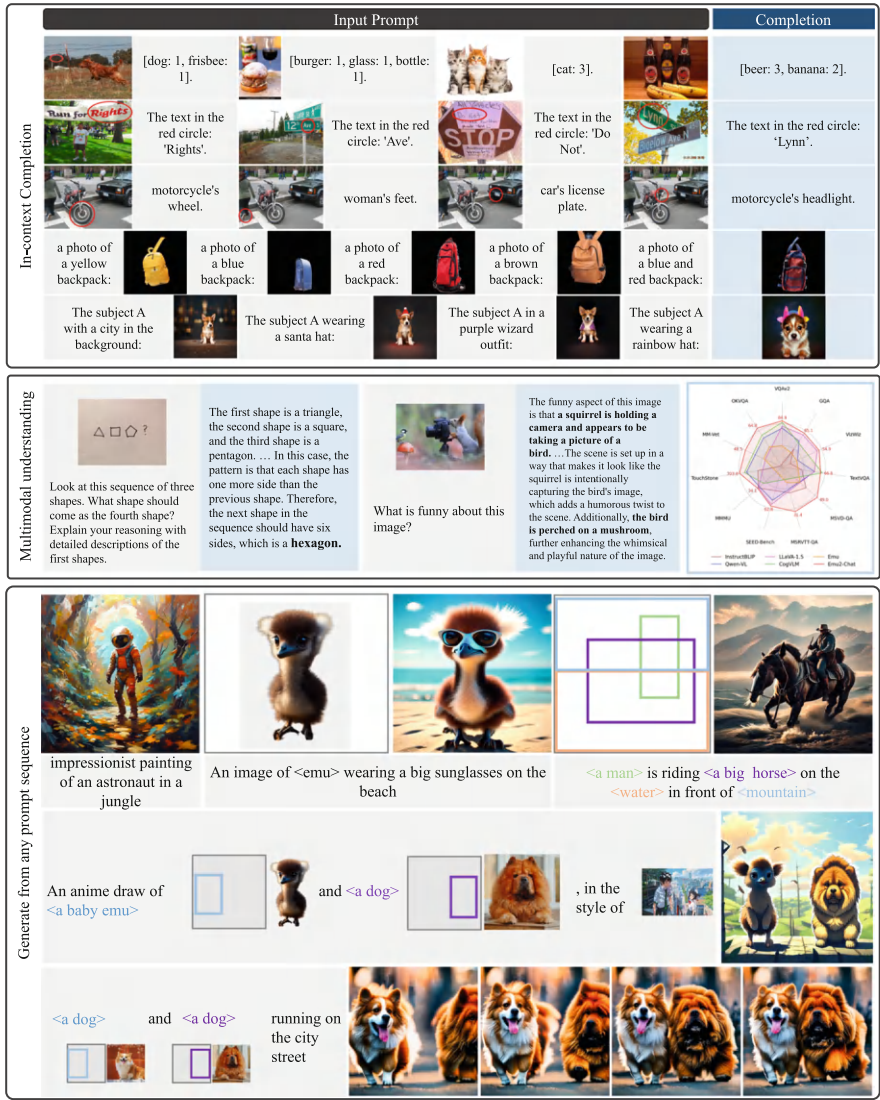


Fig. 4.1 Emu2 is a large generative multimodal model that serves as a foundation and a general-purpose interface for a broad range of multimodal tasks across understanding and generation, with remarkable in-context learning abilities

4.2 Related Work

Large Multimodal Models Recent years have seen rapid progress in large multi-modal models (LMMs), which handle tasks involving multiple types of data, such as images and text. Early breakthroughs, like CLIP [56], trained models using vast

datasets of image–text pairs with a focus on contrastive learning. This approach helps models understand the relationship between visual and textual content. Later, models like Flamingo [5] and Kosmos [30, 54] built on this, showing strong zero-shot and few-shot performance by training on interleaved image-text data, allowing them to generalize to new tasks with minimal examples.

As open-source language models improved, new approaches [40, 41, 48, 87] connected vision encoders with language models, often using a smaller model to bridge the two. Further advances refined this process through visual instruction tuning, incorporating better training pipelines [6, 38], grounding techniques [14, 15, 80, 85], and additional tasks [6]. Some studies are also focusing on unified multimodal models [22, 26, 66, 81], which can handle both visual understanding and generation tasks simultaneously. Our work builds on these efforts, inspired by Emu [66], which explored training large multimodal models using a unified generative approach for both text and image data.

In-Context Learning Large language models (LLMs) have shown remarkable capabilities in in-context learning, where they can perform new tasks by observing only a few examples, as demonstrated by models like GPT-3 [11]. This capacity has opened doors to handling complex tasks such as mathematical reasoning [77], revealing new and emergent abilities as models grow in size and data [76]. In multimodal contexts, Flamingo [5] introduced visual inputs to LLMs, allowing the model to perform visual-linguistic tasks like image captioning and optical character recognition (OCR) through language prompts.

Some studies, such as Painter [72] and SegGPT [74], have begun exploring in-context learning with purely visual tasks. Inspired by these advancements, our work examines in-context learning for multimodal tasks by scaling up generative multimodal models and evaluating their performance on a wide range of understanding and generation tasks.

4.3 Method

4.3.1 *Model Architecture*

Emu2 is a generative multimodal model designed to predict the next element in a sequence of mixed text and visual data. As shown in Fig. 4.2, the model architecture includes three main parts: Visual Encoder, Multimodal Modeling component, and Visual Decoder.

In Emu2, each image in the input sequence is processed by the Visual Encoder, which converts it into continuous embeddings or data representations. These embeddings are then combined with text tokens to form a unified sequence for autoregressive Multimodal Modeling, where the model learns to predict each token or embedding in order. The resulting visual embeddings are subsequently decoded back into images or video by the Visual Decoder.

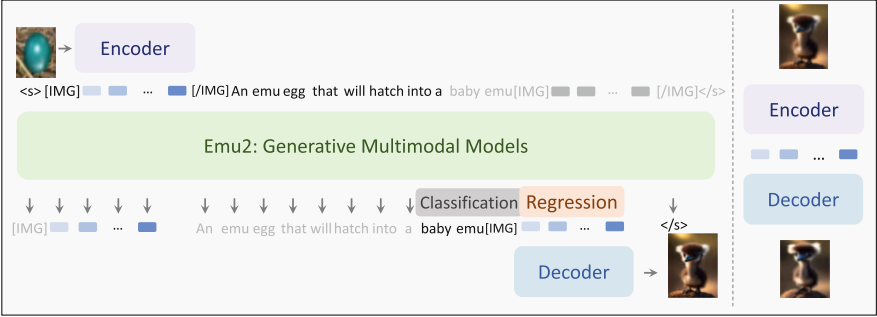


Fig. 4.2 Overview of Emu2 architecture. Emu2 learns with a predict-the-next-element objective in multimodality. Each image in the multimodal sequence is tokenized into embeddings via a visual encoder and then interleaved with text tokens for autoregressive modeling. The regressed visual embeddings will be decoded into an image or a video by a visual decoder

To build this system, we use existing powerful models as a foundation: EVA-02-CLIP-E-plus [65] for the Visual Encoder, LLaMA-33B [68] for Multimodal Modeling, and SDXL [55] for the Visual Decoder. Compared to Emu [66], Emu2 simplifies the model by connecting the Visual Encoder and Multimodal Modeling through a streamlined process—images are split into 8×8 patches, averaged, and linearly projected, eliminating the need for an extra C-Former layer.

4.3.2 Pretraining

4.3.2.1 Data

To prepare Emu2 for multimodal tasks, we trained it on a variety of publicly available datasets. These include image–text pairs from LAION-2B [61] and CapsFusion-120M [82], video–text pairs from WebVid-10M [8], as well as interleaved image–text data from Multimodal-C4 (MMC4) [88] and interleaved video–text data from YT-Storyboard-1B [66]. Additionally, we used grounded image–text pairs from GRIT-20M (introduced by Kosmos-2 [54]) and CapsFusion-grounded-100M, which comes from CapsFusion-120M. Finally, we included language-only data from Pile [25] to help the model maintain strong textual reasoning abilities.

4.3.2.2 Training

Emu2 was trained using a method similar to Emu [66], with a “predict-the-next-element” objective that works within sequences of mixed data types. Each image is encoded into visual embeddings of fixed size ($N = 64$) and then interleaved with text tokens to form a multimodal sequence. This combined interleaved sequence is processed by a transformer decoder, which learns by predicting each element in sequence.

Initially, Emu2 was trained on pairs of images and text, and pairs of videos and text, with a captioning loss applied only to the text tokens. Input images were resized to 224×224 pixels. We used the AdamW optimizer [49] with the following settings: $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 1 \times 10^{-6}$. Learning rates were set to 1×10^{-4} for the linear projection layer, 3×10^{-5} for Multimodal Modeling, and 5×10^{-5} for Visual Encoder. This stage of training involved 162 million image–text samples and 7 million video–text samples, running for 35,200 iterations with a global batch size of 6,144 for image–text pairs and 768 for video–text pairs. Following this, training resumed at a higher image resolution of 448 pixels for an additional 4,000 iterations.

Next, we froze the Visual Encoder and continued training, adjusting only the linear projection layer and Multimodal Modeling using both text classification and image regression losses. For this stage, we included additional datasets, such as interleaved image–text data, interleaved video–text data, grounded image–text pairs, and language-only data. All images were resized to 448×448 pixels, and the maximum learning rate was reduced to 1×10^{-5} . We used different batch sizes depending on the data type: 12,800 for image–text pairs, 6,400 for video–text pairs, 3,200 for interleaved data, and 800 for language-only data. This final stage of training lasted 20,350 iterations, with a total of 160 million image–text samples and 3.8 billion language-only tokens processed.

4.3.2.3 Visual Decoding

The Visual Decoder in Emu2 is designed to convert visual embeddings from the Visual Encoder back into images. Starting with SDXL-base [55] as a foundation, we adapt this model to handle the task of “autoencoding,” where it learns to generate images directly from visual embeddings. To ensure smooth processing, we adjust the dimensions of projection layers within cross-attention modules to align with the dimensions of our visual embeddings.

In contrast to Emu [66], where each training step for the Visual Decoder required complex predictions from a language model, Emu2 simplifies this by training the Visual Decoder as a “detokenizer.” This means it can be trained independently of the language model, allowing the Visual Encoder and Decoder to work as a unified image autoencoder. This setup enables Emu2 to tokenize an image into embeddings and then reconstruct it from those embeddings.

When processing video data, we apply a specialized approach using a diffusion-based decoder [63]. Similar to methods in other models [44, 69], we adapt a 2D U-Net by adding temporal layers, which help it capture changes over time, and extending spatial attention to spatial–temporal attention. This video decoder starts from Stable Diffusion 2.1 [58] and is trained to generate video clips based on Emu2’ visual embeddings.

We use the images in LAION-COCO [2] and LAION-Aesthetics [1] to train the Visual Decoder under the task of image autoencoding. The Visual Encoder and VAE in SDXL are frozen, and only the U-Net is updated during training. We adopt AdamW optimizer [49] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and the weight decay of 0.01.

We use *log* learning rate warm-up and linear learning rate decay with a peak learning rate of 1×10^{-4} for 2,000 and 6,000 steps, respectively. We filter out images whose resolution is lower than 512×512 . The input to the Visual Encoder is set to 448×448 , while the output of the Visual Decoder is set to 1024×1024 . We also employ the classifier-free guidance [29], which randomly discards image embeddings with the probability of 10%. The batch size is set to 2,048 in total.

4.3.3 Instruction Tuning

Emu2 is designed to easily adapt to specific tasks by fine-tuning it with instructional data. For instance, we fine-tune the base model with conversational data to create Emu2-Chat, a model that can respond to multimodal questions and engage in dialogue, making it ideal for interactive settings.

Similarly, we develop Emu2-Gen, a controllable visual generation model. Emu2-Gen can take in a combination of text, image placements, and reference images to produce visuals that align closely with the provided instructions. This allows Emu2-Gen to generate images that are accurately grounded in the specified descriptions or subjects, offering more control over the visual output.

4.3.3.1 Instruction-Following Chat

Training Data To equip Emu2-Chat with strong instruction-following capabilities, we train on a mix of academic-task-oriented and multimodal chat datasets, combining both structured and conversational data. This balanced approach enhances Emu2-Chat’s ability to follow diverse instructions while maintaining visual knowledge.

Academic-task-oriented datasets often have concise annotations, limiting the model’s capacity for detailed responses. To address this, we use distinct system messages and include output-format control instructions, following practices in [47]. The training data include:

- **Academic-task-oriented data:** image captioning [18, 62], visual question answering [27, 31, 64], knowledgeable question answering [50, 52], multimodal classification [43], and referring expression comprehension [32, 51].
- **Multimodal chat data:** GPT-assisted visual instructions [48, 86], language instruction [4, 67], clock reading [79], and video chat data [42].

Training Objective In training Emu2-Chat, we introduce two special tokens, [USER] and [ASSISTANT], to clarify conversation roles. This structure allows us

to organize different data types with a consistent format:

<System Message> [USER]: <Instruction> [ASSISTANT]: <Answer>

Here, <System Message> varies between academic-task and chat data to reflect each task’s style. The <Instruction> section includes multimodal tokens, such as images, videos, and text. Only the <Answer> tokens are supervised by cross-entropy loss, guiding the model’s responses.

Training Setup We use a global batch size of 768 for training, which runs for 8,000 steps. The learning rate ramps up to 1×10^{-5} in the first 100 steps and then decreases to zero following a cosine decay schedule. Training is conducted with the AdamW optimizer (parameters: $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1 \times 10^{-6}$, and a gradient clipping of 5.0). Sequence length is capped at 2048 tokens, with any additional input truncated.

The input resolution for images and videos is set to 448×448 . For videos, we sample frames uniformly in time, with a varying number of frames (8, 12, or 16) chosen randomly. To capture finer spatial details during instruction fine-tuning, we divide each image into 16×16 tokens after mean-pooling in the visual encoder, a step up from the 8×8 tokens used in pretraining.

4.3.3.2 Controllable Visual Generation

Training Data To maximize Emu2-Gen’s potential for controllable, high-quality image generation, we use a curated combination of datasets. For grounded text-to-image generation, we leverage CapsFusion-grounded-100M and GRIT [54]. To optimize multi-entity subject-driven generation, we preprocess grounding data with SAM [34], creating a subset of 5 million samples with segmentation results to minimize background influence. For image editing tasks, we utilize InstructPix2Pix [10]. Our text-to-image dataset includes a filtered selection from CapsFusion [82], LAION-Aesthetics [1], SA-1B [34], and LAION-High-Resolution [3], supplemented with high-quality images from sources such as Unsplash [20] and generated outputs from Midjourney-V5 [53] and DALL-E-3 [9]. This training set comprises around 500k high-quality image-text pairs, filtered to retain only samples with resolutions exceeding 448×448 to ensure output quality.

Training Objective We employ a unified generative pretraining objective to handle diverse visual generation tasks. Each training sample includes interleaved text, images, and object localization information, formatted as follows:

```

<s> A photo of <p>a man</p> <coor> image embedding of object
localization image </coor> [IMG] image embedding of man [/IMG]
sitting next to <p>a dog</p> <coor> image embedding of object
localization image </coor> [IMG] image embedding of dog [/IMG] [IMG]
image embedding of the whole image [/IMG]</s>

```

Here, object coordinates are represented visually by drawing bounding boxes on a black image at specified locations. Emu2-Gen integrates text, object images, and corresponding localization images in a unified multimodal format, with a regression loss applied only to the final image embedding. During fine-tuning, we freeze the visual encoder to prevent model collapse. We further improve robustness by randomly dropping entity tokens and localization information and by applying random background variations and crops to object images.

Training Setup The model is trained with a global batch size of 4,096 for 3,000 steps. The learning rate warms up to 5×10^{-5} in the first 100 steps, then follows a cosine decay to zero. We further fine-tune with the 500k high-quality pairs, using a batch size of 2,048 for an additional 900 steps.

4.4 Experiments

4.4.1 Pretrained Base Model Evaluation

We evaluate the zero-shot and few-shot capabilities of Emu2 across several vision-language benchmarks, including OKVQA [52], VQAv2 [27], VizWiz [28], TextVQA [64], and HatefulMemes [33]. Detailed descriptions of the datasets and prompts can be found in the supplementary materials. The results are summarized in Table 4.1. Emu2 demonstrates exceptional in-context learning ability, with performance improving as more in-context examples are provided. Specifically, on the VQAv2, VizWiz, and TextVQA datasets, Emu2 outperforms both Flamingo-80B and IDEFICS-80B, despite being trained with a much smaller 37-billion parameter model.

Figure 4.1 highlights Emu2’s few-shot learning capabilities in real-world scenarios. For instance, the model can learn to classify and count objects simultaneously in a specific format from a small number of examples (row 1). Furthermore, Emu2 is adept at following visual prompts in context, such as red circles overlaid on images (row 2 and 3).

Table 4.1 Zero-shot and few-shot evaluations of Emu2. 0* denotes text two-shot and image zero-shot results following Flamingo [5]. The best results are in **bold**, and the second best are underlined. HM represents HatefulMemes [33]

Model	Shot	VQAv2	OKVQA	VizWiz	TextVQA	HM
Kosmos-1 (1.6B)	0	51.0	–	29.2	–	–
	4	51.8	–	35.3	–	–
	8	51.4	–	39.0	–	–
Flamingo (9B)	0*	51.8	44.7	28.8	31.8	57.0
	4	56.3	49.3	34.9	33.6	62.7
	8	58.0	50.0	39.4	33.6	63.9
	16	59.4	50.8	43.0	33.5	64.5
Flamingo (80B)	0*	56.3	50.6	31.6	35.0	46.4
	4	63.1	57.4	39.6	36.5	<u>68.6</u>
	8	65.6	<u>57.5</u>	44.8	37.3	70.0
	16	66.8	57.8	48.4	37.6	70.0
IDEFICS (80B)	0*	60.0	45.2	36.0	30.9	60.6
	4	63.6	52.4	40.4	34.4	57.8
	8	64.8	55.1	46.1	35.7	58.2
	16	65.4	56.8	48.3	36.3	57.8
Emu (14B)	0*	52.9	42.8	34.4	–	–
	4	58.4	–	41.3	–	–
	8	59.0	–	43.9	–	–
	16	–	–	–	–	–
Emu2 (37B)	0	33.5	26.7	40.4	26.4	52.2
	4	67.0	53.2	54.6	48.2	62.4
	8	<u>67.8</u>	54.1	<u>54.7</u>	<u>49.3</u>	65.8
	16	68.8	57.1	57.0	50.3	66.0

4.4.2 Instruction-Following Chat Evaluation

We evaluate Emu2-Chat on academic task-oriented benchmarks, including image question-answering tasks (VQAv2 [27], OKVQA [52], GQA [31], VizWiz [28], TextVQA [64]) and video question-answering datasets (MSVD [78] and MSRVT [78]). We also assess Emu2-Chat on newer benchmarks for large multimodal models, such as SEED-Bench [37], MM-Vet [83], TouchStone [7], and MMMU [84]. For SEED-Bench, we followed the evaluation setup from LLaVa-1.5 [47], where the model completes multiple-choice tasks with provided options.

As shown in Tables 4.2 and 4.3, Emu2-Chat consistently outperforms other models in image question-answering tasks, including well-established benchmarks like VQAv2 and GQA. Notably, it shows significant improvement on the OKVQA task, which requires external knowledge, demonstrating the model’s ability to handle real-world knowledge effectively.

For video question-answering, Emu2-Chat outperforms existing models despite not being trained specifically on video QA datasets. It achieves accuracies of

Table 4.2 Results on LMM benchmarks

Model	SEED-Bench	MM-Vet	TouchStone	MMMU
Kosmos-2 [54]	50.0	–	–	26.6
BLIP-2-13B [40]	46.4	22.4	–	–
InstructBLIP-13B [21]	–	25.6	552.4	–
Qwen-VL-13B-Chat [6]	58.2	–	645.2	–
LLaVA-1.5-13B [47]	61.6	35.4	–	33.6
CogVLM [71]	–	–	662.6	30.1
Emu-I [66]	–	36.3	–	–
Emu2-Chat	62.8	48.5	703.8	34.1

* indicates that samples from this task’s training set have been trained. For MM-Vet, we present the average result of five scoring runs

Table 4.3 Results on visual question answering

Model	VQAv2	OKVQA	GQA	VizWiz	TextVQA	MSVD	MSRVTT
Flamingo-9B [5]	51.8	44.7	–	28.8	–	30.2	13.7
Flamingo-80B [5]	56.3	50.6	–	31.6	–	35.6	17.4
Kosmos-1 [30]	51.0	–	–	29.2	–	–	–
Kosmos-2 [54]	51.1	–	–	–	–	–	–
BLIP-2-13B [40]	–	–	41.0	19.6	42.5	20.3	10.3
InstructBLIP-13B [21]	–	–	49.5	33.4	50.7	41.2	24.8
IDEFICS-9B [36]	50.9	38.4	–	35.5	25.9	–	–
IDEFICS-80B [36]	60.0	45.2	–	36.0	30.9	–	–
Shikra-13B [15]	77.4*	47.2	–	–	–	–	–
Qwen-VL-13B-Chat [6]	78.2*	56.6*	57.5*	38.9	61.5*	–	–
LLaVA-1.5-13B [47]	80.0*	–	63.3*	53.6	61.3	–	–
CogVLM [71]	83.4*	58.9*	–	–	68.1*	–	–
Emu-I [66]	62.0	49.2	46.0	38.3	–	37.0	21.2
Emu2-Chat	84.9*	64.8*	65.1*	54.9	66.6*	49.0	31.4

* indicates that samples from this task’s training set have been trained

49.0 and 31.4 on the MSVD-QA and MSRVTT-QA tasks, respectively, surpassing InstructBLIP and the larger Flamingo-80B. Moreover, our model excels on the LMM benchmarks, achieving a score of 48.5 on MM-Vet and 703.8 on TouchStone, demonstrating its superior multimodal problem-solving capabilities.

In addition to these benchmarks, we evaluate Emu2-Chat’s visual grounding ability using refer expression comprehension tasks. As shown in Table 4.4, Emu2-Chat achieves the best results among generalist models on RefCOCO [32], RefCOCO+ [51], and RefCOCOg [51]. Its most notable advantage is in RefCOCO+,

Table 4.4 Results on referring expression comprehension

Models	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test
OFA-L [70]	79.96	83.67	76.39	68.29	76.00	61.75	67.57	67.58
Shikra-7B [15]	87.01	90.61	80.24	81.60	87.36	72.12	82.27	82.19
Shikra-13B [15]	87.83	91.11	81.81	82.89	87.79	74.41	82.64	83.16
Qwen-VL-7B [6]	89.36	92.26	85.34	83.12	88.25	77.21	85.58	85.48
Emu2-Chat	90.40	93.88	85.97	87.05	91.43	80.47	87.64	88.11

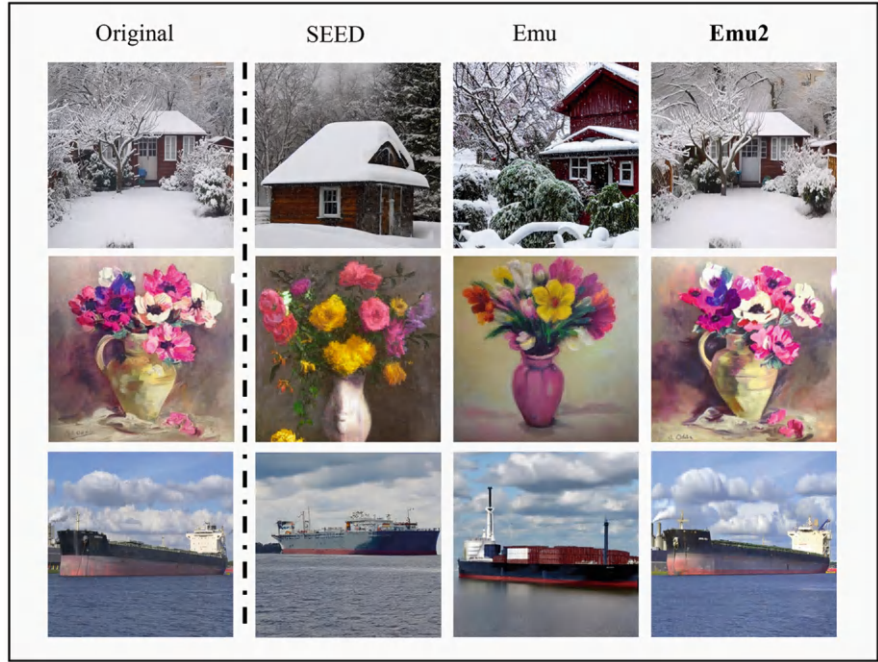


Fig. 4.3 Comparison of autoencoding results among different methods [26, 66]. Emu2’s Visual Encoder and Visual Decoder in the architecture of CLIP-Diffusers form a strong autoencoder

which focuses on appearance-based descriptions without positional references, underscoring the model’s strong perceptual capabilities.

4.4.3 Controllable Visual Generation Evaluation

Qualitative Results Figure 4.3 illustrates Emu2’s autoencoding performance. Using Emu2’s Visual Encoder and Visual Decoder, we tokenize images into visual embeddings and detokenize them back into images. Compared to previous models

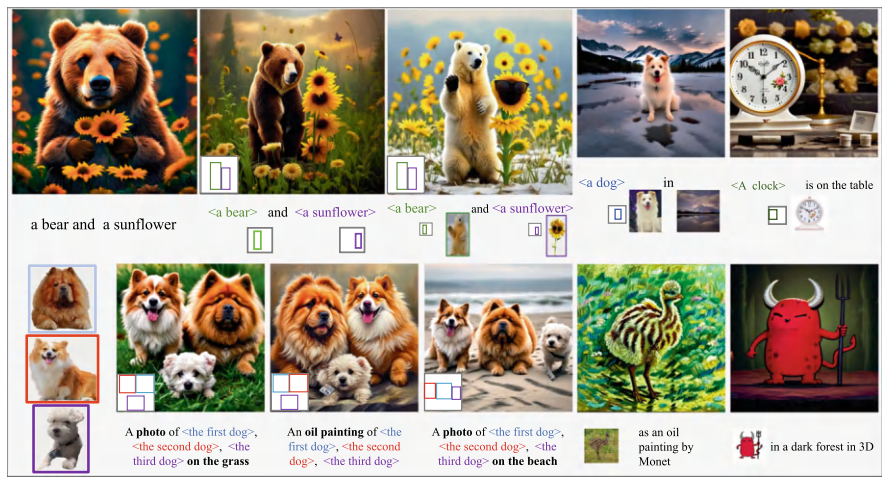


Fig. 4.4 Visualization of Emu2-Gen’s controllable generation capability. The model is capable of accepting a mix of text, locations and images as input, and generating images in context. The presented examples include text- and subject-grounded generation, stylization, multi-entity composition, subject-driven editing, and text-to-image generation

like SEED [26] and Emu [66], Emu2 demonstrates superior image reconstruction quality. On the MS-COCO [46] dataset, Emu2 achieves a strong 0.907 CLIP-I [56] score, further highlighting its visual encoding and decoding strength.

Figure 4.4 showcases Emu2-Gen’s ability to generate images based on a mixture of text, locations, and images as inputs. In a zero-shot setting, Emu2-Gen excels at controllable visual generation, performing tasks such as re-contextualization, stylization, modification, region-controllable generation, and multi-entity composition. For example, the model generates images of three dogs under varying conditions, demonstrating its flexibility in handling diverse visual scenarios.

Zero-shot Text-to-image Generation We evaluate Emu2-Gen’s zero-shot text-to-image generation capability on a dataset of 30,000 randomly sampled images from the MS-COCO [46] validation set. We use CLIP-ViT-B [56] to calculate the CLIP-T score for assessing prompt-following ability and CLIP-ViT-L, as used in GILL [35], to compute the CLIP-I score for image similarity. Higher scores indicate that the generated image better matches the prompt or real image. Table 4.5 shows that Emu2-Gen outperforms various unimodal and multimodal models in both CLIP-I and CLIP-T scores, achieving state-of-the-art performance in zero-shot text-to-image generation.

Zero-shot Subject-driven Generation We also evaluate Emu2-Gen’s subject-driven image editing capabilities using DreamBench [59]. DreamBench includes 30 subjects and 25 prompt templates, resulting in 750 unique prompts that test a wide range of skills, such as re-contextualization, modification, and accessorization. We generate four images for each prompt, totaling 3,000 images for comprehensive

Table 4.5 Quantitative comparison of zero-shot text-to-image generation on MS-COCO [46] validation set. 30k samples are randomly sampled

Type	Models	CLIP-I $\uparrow$	CLIP-T $\uparrow$
<i>unimodal generation models</i>	MUSE [13]	–	0.320
	Imagen [60]	–	0.270
	DALL-E 2 † [57]	–	0.314
	DALL-E 3 † [9]	–	0.320
	SDv1.5 [58]	0.667	0.302
	SDXL [55]	0.674	0.310
<i>multimodal generation models</i>	GILL [35]	0.684	–
	SEED [26]	0.682	–
	Emu [66]	0.656	0.286
	Emu2-Gen	0.686	0.297

†CLIP-T score is calculated on 4,096 samples. We also evaluate our image autoencoding results on MS-COCO, which achieves a strong 0.907 CLIP-I score

Table 4.6 Quantitative comparison of zero-shot single-entity subject-driven generation on DreamBench

Methods	DINO $\uparrow$	CLIP-I $\uparrow$	CLIP-T $\uparrow$
Real Images (Oracle)	0.774	0.885	–
Textual Inversion [23]	0.569	0.780	0.255
DreamBooth [59]	0.668	0.803	0.305
BLIP-Diffusion [40]	0.670	0.805	0.302
SuTI [16]	0.741	0.819	0.304
Re-Imagen* [17]	0.600	0.740	0.270
BLIP-Diffusion* [40]	0.594	0.779	0.300
Kosmos-G* (single image input)	0.694	0.847	0.287
Emu2-Gen* (single image input)	0.766	0.850	0.287

* denotes zero-shot methods

evaluation. Using DINO [12] and CLIP-I [56] for subject fidelity evaluation, and CLIP-T [56] for text fidelity, we find that Emu2-Gen excels in subject fidelity. It achieves superior performance on DINO and CLIP-I metrics compared to methods like BLIP-Diffusion and Kosmos-G. In zero-shot settings, Emu2-Gen impressively reconstructs subjects from a single image input, showcasing its exceptional subject fidelity and powerful visual decoding capabilities (Table 4.6).

4.5 Conclusion

In this chapter, we present a 37 billion-parameter generative multimodal model Emu2 that shows strong performance and versatility on many multimodal tasks in the in-context settings. Emu2 serves as a base model and a general-purpose interface

for a variety of multimodal tasks. We demonstrate state-of-the-art results on a broad range of benchmarks of multimodal understanding and generation. Specifically, our model largely surpasses prior work on the lately proposed LMM benchmarks that require more advanced capability compared to classic academic benchmarks. Emu2 also shows remarkable capability of controllable visual generation in multimodal context, e.g., subject-/text-grounded generation. Additionally, we review the limitations and broader social impact of Emu2. Despite discussed weaknesses, these results suggest that generative multimodal model at scale may be an important step toward the development of adaptable, general multimodal systems.

Further Reading

Interested readers may refer to the following material:

- [11] for exploring in-context learning in language
- [73] for exploring in-context learning in vision
- [75] for unifying multimodal learning with solely next-token prediction
- [45, 82] for rethinking and constructing multimodal data at scale

References

1. Laion-aesthetics. <https://laion.ai/blog/laion-aesthetics/>
2. Laion coco: 600m synthetic captions from laion2b-en. <https://laion.ai/blog/laion-coco/>
3. Laion-high-resolution. <https://huggingface.co/datasets/laion/laion-high-resolution>
4. Sharegpt. <https://sharegpt.com/>
5. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. *Adv. Neural Inf. Proces. Syst.* **35**, 23716–23736 (2022)
6. 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* (2023)
7. Bai, S., Yang, S., Bai, J., Wang, P., Zhang, X., Lin, J., Wang, X., Zhou, C., Zhou, J.: Touchstone: Evaluating vision-language models by language models. *arXiv preprint arXiv:2308.16890* (2023)
8. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1728–1738 (2021)
9. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao, Y., Ramesh, A.: Improving Image Generation with Better Captions (2023)
10. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402 (2023)
11. 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. Proces. Syst.* **33**, 1877–1901 (2020)

12. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660 (2021)
13. Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.H., Murphy, K., Freeman, W.T., Rubinstein, M., et al.: Muse: text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704* (2023)
14. Chen, C., Qin, R., Luo, F., Mi, X., Li, P., Sun, M., Liu, Y.: Position-enhanced visual instruction tuning for multimodal large language models. *arXiv preprint arXiv:2308.13437* (2023)
15. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: unleashing multimodal llm’s referential dialogue magic. *arXiv preprint arXiv:2306.15195* (2023)
16. Chen, W., Hu, H., Li, Y., Rui, N., Jia, X., Chang, M.W., Cohen, W.W.: Subject-driven text-to-image generation via apprenticeship learning. *arXiv preprint arXiv:2304.00186* (2023)
17. Chen, W., Hu, H., Saharia, C., Cohen, W.W.: Re-imagen: retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491* (2022)
18. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: data collection and evaluation server. *arXiv preprint arXiv:1504.00325* (2015)
19. Chen, X., Wang, X., Beyer, L., Kolesnikov, A., Wu, J., Voigtlaender, P., Mustafa, B., Goodman, S., Alabdulmohsin, I., Padlewski, P., et al.: Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199* (2023)
20. Chesser, L., Carbone, T.: Unsplash. <https://github.com/unsplash/datasets>
21. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B.A., Fung, P., Hoi, S.C.H.: Instructblip: towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500* (2023)
22. Dong, R., Han, C., Peng, Y., Qi, Z., Ge, Z., Yang, J., Zhao, L., Sun, J., Zhou, H., Wei, H., Kong, X., Zhang, X., Ma, K., Yi, L.: Dreamllm: synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499* (2023)
23. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618* (2022)
24. Gan, Z., Li, L., Li, C., Wang, L., Liu, Z., Gao, J., et al.: Vision-language pre-training: basics, recent advances, and future trends. *Found. Trends® Comput. Graphics and Vision* **14**(3–4), 163–352 (2022)
25. Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., Leahy, C.: The Pile: an 800 gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027* (2020)
26. Ge, Y., Ge, Y., Zeng, Z., Wang, X., Shan, Y.: Planting a seed of vision in large language model. *arXiv preprint arXiv:2307.08041* (2023)
27. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: elevating the role of image understanding in visual question answering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913 (2017)
28. Gurari, D., Li, Q., Stangl, A.J., Guo, A., Lin, C., Grauman, K., Luo, J., Bigham, J.P.: Vizwiz grand challenge: answering visual questions from blind people. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617 (2018)
29. Ho, J., Salimans, T.: Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022)
30. 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* (2023)
31. Hudson, D.A., Manning, C.D.: GQA: a new dataset for real-world visual reasoning and compositional question answering. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709 (2019)
32. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 787–798 (2014)

33. Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Ringshia, P., Testuggine, D.: The hateful memes challenge: detecting hate speech in multimodal memes. *Adv. Neural Inf. Proces. Syst.* **33**, 2611–2624 (2020)
34. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. *arXiv preprint arXiv:2304.02643* (2023)
35. Koh, J.Y., Fried, D., Salakhutdinov, R.: Generating images with multimodal language models. *arXiv preprint arXiv:2305.17216* (2023)
36. Laurençon, H., Saulnier, L., Tronchon, L., Bekman, S., Singh, A., Lozhkov, A., Wang, T., Karamcheti, S., Rush, A.M., Kiela, D., et al.: Obelics: an open web-scale filtered dataset of interleaved image-text documents. In: *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track* (2023)
37. 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* (2023)
38. Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., Liu, Z.: Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726* (2023)
39. Li, C., Gan, Z., Yang, Z., Yang, J., Li, L., Wang, L., Gao, J.: Multimodal foundation models: from specialists to general-purpose assistants. *Found. Trends® Comput. Graphics and Vision* **1**(2), 2 (2023). *arXiv preprint arXiv:2309.10020*
40. Li, D., Li, J., Hoi, S.C.: Blip-diffusion: pre-trained subject representation for controllable text-to-image generation and editing. *arXiv preprint arXiv:2305.14720* (2023)
41. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: bootstrapping language-image pre-training for unified vision-language understanding and generation. In: *International Conference on Machine Learning*, pp. 12888–12900. PMLR, New York (2022)
42. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: Videochat: chat-centric video understanding. *arXiv preprint arXiv:2305.06355* (2023)
43. Li, L., Yin, Y., Li, S., Chen, L., Wang, P., Ren, S., Li, M., Yang, Y., Xu, J., Sun, X., et al.: M3it: A large-scale dataset towards multi-modal multilingual instruction tuning. *arXiv preprint arXiv:2306.04387* (2023)
44. Li, X., Chu, W., Wu, Y., Yuan, W., Liu, F., Zhang, Q., Li, F., Feng, H., Ding, E., Wang, J.: Videogen: a reference-guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398* (2023)
45. Li, X., Zhang, F., Diao, H., Wang, Y., Wang, X., Duan, L.Y.: Densefusion-1m: merging vision experts for comprehensive multimodal perception. 2407.08303 (2024)
46. 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: *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, Berlin (2014)
47. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744* (2023)
48. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. *arXiv preprint arXiv:2304.08485* (2023)
49. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017)
50. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: multimodal reasoning via thought chains for science question answering. *Adv. Neural Inf. Proces. Syst.* **35**, 2507–2521 (2022)
51. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11–20 (2016)
52. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Ok-vqa: A visual question answering benchmark requiring external knowledge. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3195–3204 (2019)
53. Midjourney: Midjourney. <https://www.midjourney.com>

54. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Wei, F.: Kosmos-2: grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824* (2023)
55. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: improving latent diffusion models for high-resolution image synthesis (2023)
56. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning*, pp. 8748–8763. PMLR, New York (2021)
57. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022)
58. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695 (2022)
59. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: fine tuning text-to-image diffusion models for subject-driven generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510 (2023)
60. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. *Adv. Neural Inf. Proces. Syst.* **35**, 36479–36494 (2022)
61. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: an open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402* (2022)
62. Sidorov, O., Hu, R., Rohrbach, M., Singh, A.: Textcaps: a dataset for image captioning with reading comprehension. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 742–758. Springer, Berlin (2020)
63. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792* (2022)
64. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards VQA models that can read. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326 (2019)
65. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389* (2023)
66. Sun, Q., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, Y., Gao, H., Liu, J., Huang, T., Wang, X.: Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222* (2023)
67. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: an instruction-following llama model (2023)
68. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023)
69. Wang, J., Yuan, H., Chen, D., Zhang, Y., Wang, X., Zhang, S.: Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571* (2023)
70. Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: *International Conference on Machine Learning*, pp. 23318–23340. PMLR, New York (2022)
71. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., et al.: Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079* (2023)
72. Wang, X., Wang, W., Cao, Y., Shen, C., Huang, T.: Images speak in images: a generalist painter for in-context visual learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6830–6839 (2023)

73. Wang, X., Wang, W., Cao, Y., Shen, C., Huang, T.: Images speak in images: a generalist painter for in-context visual learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6830–6839 (2023)
74. Wang, X., Zhang, X., Cao, Y., Wang, W., Shen, C., Huang, T.: Seggpt: segmenting everything in context. *arXiv preprint arXiv:2304.03284* (2023)
75. Wang, X., Zhang, X., Luo, Z., Sun, Q., Cui, Y., Wang, J., Zhang, F., Wang, Y., Li, Z., Yu, Q., et al.: Emu3: next-token prediction is all you need. *arXiv preprint arXiv:2409.18869* (2024)
76. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al.: Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022)
77. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Proces. Syst.* **35**, 24824–24837 (2022)
78. Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., Zhuang, Y.: Video question answering via gradually refined attention over appearance and motion. In: *Proceedings of the 25th ACM International Conference on Multimedia*, pp. 1645–1653 (2017)
79. Yang, C., Xie, W., Zisserman, A.: It’s about time: analog clock reading in the wild. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2508–2517 (2022)
80. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.F., Yang, Y.: Ferret: refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704* (2023)
81. Yu, L., Shi, B., Pasunuru, R., Muller, B., Golovneva, O., Wang, T., Babu, A., Tang, B., Karrer, B., Sheynin, S., et al.: Scaling autoregressive multi-modal models: pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591* (2023)
82. Yu, Q., Sun, Q., Zhang, X., Cui, Y., Zhang, F., Wang, X., Liu, J.: Capsfusion: rethinking image-text data at scale. *arXiv preprint arXiv:2310.20550* (2023)
83. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490* (2023)
84. 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* (2023)
85. Zhang, S., Sun, P., Chen, S., Xiao, M., Shao, W., Zhang, W., Chen, K., Luo, P.: Gpt4roi: instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601* (2023)
86. Zhang, Y., Zhang, R., Gu, J., Zhou, Y., Lipka, N., Yang, D., Sun, T.: Llavav: enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint arXiv:2306.17107* (2023)
87. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigt-4: enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023)
88. Zhu, W., Hessel, J., Awadalla, A., Gadre, S.Y., Dodge, J., Fang, A., Yu, Y., Schmidt, L., Wang, W.Y., Choi, Y.: Multimodal c4: an open, billion-scale corpus of images interleaved with text. *arXiv preprint arXiv:2304.06939* (2023)

Part II
**Shaping Intelligence: Prompting
Techniques for Multimodal Adaptation**

Chapter 5

Differentiable Prompt Learning for Vision-Language Models



Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu

Abstract Large pre-trained vision-language models like CLIP have demonstrated significant potential for learning representations that can be applied across a variety of downstream tasks. Unlike traditional representation learning, which primarily relies on labeled data, vision-language pre-training aligns images and text within a shared feature space. This alignment allows for zero-shot transfer to downstream tasks through *prompting*, where classification weights are generated from natural language descriptions of the target classes. However, a major hurdle in deploying these models is prompt engineering, which is time-consuming and requires substantial domain expertise. Small changes in wording can significantly affect performance, making the process labor-intensive. In this chapter, we discuss a simple yet effective approach called *Context Optimization (CoOp)* for adapting CLIP-like vision-language models for image recognition tasks. CoOp uses learnable vectors to model the context words of prompts while keeping the pre-trained model parameters fixed. On 11 benchmark datasets, CoOp outperforms hand-crafted prompts with as few as one or two examples. Despite being a learning-based approach, CoOp also exhibits excellent domain generalization, surpassing zero-shot models that use hand-crafted prompts.

Keywords Vision-Language Model · Prompt Learning · Prompt Engineering · Image Classification · Domain Generalization · Robustness · Few-Shot Learning · Zero-Shot Learning · Contrastive Learning

K. Zhou (✉)

CS Department, Hong Kong Baptist University, Hong Kong, China

e-mail: kyzhou@hkbu.edu.hk

J. Yang · C. C. Loy · Z. Liu

S-Lab, Nanyang Technological University, Singapore, Singapore

e-mail: jingkang001@ntu.edu.sg; cclloy@ntu.edu.sg; ziwei.liu@ntu.edu.sg

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026

K. Zhou et al. (eds.), *Large Vision-Language Models*, Advances in Computer Vision and Pattern Recognition, https://doi.org/10.1007/978-3-031-94969-2_5

115

5.1 Introduction

Vision-language pre-training methods like CLIP [37] and ALIGN [20] have emerged as promising alternatives for visual representation learning. The core idea is to align images and raw texts using two separate encoders—one for each modality. For example, both CLIP and ALIGN use a contrastive loss objective, which brings together images and their textual descriptions while pushing apart unmatched pairs in the feature space. By pre-training on a large scale, models can learn diverse visual concepts and can be easily adapted to any downstream task through *prompting* [11, 20, 30, 37, 41, 48]. Specifically, for any new classification task, one can generate classification weights by providing sentences describing task-relevant categories to the text encoder and then compare these with image features produced by the image encoder.

We observe that for pre-trained vision-language models, the text input, known as the prompt, plays a crucial role in downstream datasets. However, finding the appropriate prompt is a challenging task that often requires substantial time for fine-tuning, and minor changes in wording can significantly impact performance. For instance, in the Caltech101 dataset (Fig. 5.1, 2nd vs 3rd prompt), adding “a” before the class token increases accuracy by more than 5%. Additionally, prompt engineering requires prior knowledge about the task and ideally an understanding of the language model’s underlying mechanisms. This is illustrated in the DTD dataset in Fig. 5.1, where adding task-relevant context (i.e., “texture”) can lead to significant improvements. Adjusting the sentence structure can also yield further improvements, such as using only “texture” in the context for DTD. However, even with extensive tuning, the resulting prompts are not guaranteed to be optimal for these downstream tasks.

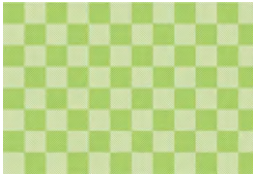
Caltech101		Describable Textures (DTD)	
			
Prompt	Accuracy	Prompt	Accuracy
a [CLASS].	82.68	a [CLASS].	39.83
a photo of [CLASS].	80.81	a photo of [CLASS].	40.25
a photo of a [CLASS].	86.29	a photo of a [CLASS].	42.32
$[V]_1 [V]_1 \dots [V]_M [CLASS].$	91.83	$[V]_1 [V]_1 \dots [V]_M [CLASS].$	63.58

Fig. 5.1 Comparison between prompt engineering and prompt learning (i.e., our CoOp method). Prompt engineering requires a held-out validation set for words tuning while prompt learning automates the tuning process and achieves better results

In this chapter, we discuss a simple yet effective approach called *Context Optimization (CoOp)* to automate prompt engineering for large vision-language models.¹ Specifically, CoOp models a prompt’s context words with learnable vectors, which can be initialized with either random values or pre-trained word embeddings. Since neural networks are usually differentiable, CoOp can be applied to any neural network architectures. Given an image recognition task, we can simply minimize prediction errors using the cross-entropy loss with respect to the learnable context vectors while keeping the entire pre-trained parameters fixed. The gradients can be back-propagated through the text encoder, distilling the rich knowledge encoded in the parameters for learning task-relevant context.

The main findings derived from experiments on 11 datasets are summarized as follows. We observe that CoOp effectively transforms pre-trained vision-language models into data-efficient visual learners, requiring as few as one or two shots to outperform hand-crafted prompts by a significant margin. The performance can be further enhanced with more shots. For example, with 16 shots, the margin over hand-crafted prompts averages around 15% and reaches over 45% at its highest. CoOp also surpasses the linear probe model, which is known to be a strong few-shot learning baseline [44]. Furthermore, CoOp demonstrates much greater robustness than the zero-shot model (which uses manual prompts) to domain shifts, despite being a learning-based approach.²

5.2 Related Work

5.2.1 Vision-Language Models

Vision-language models serve as powerful tools for learning visual representations that integrate both visual and textual data [20, 37, 41, 48, 51]. The idea of mapping images and text onto a common embedding space has been studied since nearly a decade ago [8, 10, 42], but with drastically different technologies. For text features extraction, early work has mainly utilized pre-trained word vectors [10, 42] or the hand-crafted TF-IDF features [8, 26]. Matching images and text features has been formulated as metric learning [10], multi-label classification [13, 23], n-gram language learning [28], and captioning [6]. Advancements in vision-language learning, particularly exemplified by models like CLIP [37] and ALIGN [20], are largely driven by progress in three fundamental areas:

- **Text Representation Learning with Transformers:** The Transformer architecture [45] revolutionized natural language processing by enabling models to capture complex relationships in textual data. Transformers use self-attention

¹ CoOp is pronounced as /ku:p/.

² The codebase has been released at <https://github.com/KaiyangZhou/CoOp>.

mechanisms to weigh the importance of different words in a sentence, resulting in rich and contextualized text embeddings.

- **Large-Batch Contrastive Representation Learning:** Contrastive learning has become a cornerstone in representation learning [3, 14, 17]. By utilizing large mini-batches, models can compare numerous positive and negative pairs simultaneously, enhancing their ability to learn discriminative features that distinguish between different classes or concepts.
- **Access to Massive Web-Scale Training Datasets:** The availability of extensive image–text datasets is crucial. CLIP leverages a curated dataset of 400 million image–text pairs, ensuring high-quality associations between images and descriptions. ALIGN utilizes 1.8 billion image–text pairs, albeit with more noise due to less stringent curation processes. These vast datasets enable models to learn a wide variety of concepts and relationships between visual and textual data.

With these efforts, large-scaled models [20, 37] make breakthroughs significantly in technology and scope. The better alignment between images and text enables zero-shot transfer, allowing them to apply learned knowledge to a wide array of downstream classification tasks without additional training. This capability is often achieved through prompting techniques, where textual cues guide the model’s predictions.

In the context of these developments, CoOp focuses on facilitating the adaptation and deployment of vision-language models in specific downstream tasks, developing techniques and tools that enable practitioners to fine-tune and integrate vision-language models more effectively and automatically into their applications.

5.2.2 *Prompt Learning in NLP*

Prompt learning has emerged as a significant area of research in natural language processing (NLP), focusing on leveraging large pre-trained language models to perform various downstream tasks without extensive fine-tuning [12, 22, 27, 29, 32, 40, 52]. One foundational concept in this domain is knowledge probing, which is formally defined as “fill-in-the-blank” cloze tests [36]. Knowledge probing involves providing a pre-trained language model with incomplete sentences where certain words are masked and assessing the model’s ability to predict the missing words based on its learned knowledge.

The basic idea is to tap into the vast amount of information encoded within these models by crafting prompts that elicit specific responses. This approach benefits various tasks such as sentiment analysis, factual knowledge retrieval, and question answering, by utilizing the model’s inherent understanding of language and world knowledge. Several methods have been developed to optimize prompts for better performance:

- **Text Mining and Paraphrasing Techniques:** One strategy involves generating a large set of candidate prompts through automated text mining and paraphrasing. These candidate prompts are then evaluated to identify those that result in the highest accuracy for the task at hand [22]. This method helps find the most effective ways to phrase prompts that align well with the model’s internal representations.
- **Gradient-Based Token Search:** Another approach uses gradient-based optimization to identify tokens that have the most significant impact on the model’s output probabilities. By searching for tokens that cause the largest changes in the likelihood of the correct label, this technique effectively discovers the most influential words to include in the prompts [40]. This method provides a systematic way to refine prompts based on the model’s sensitivities.
- **Continuous Prompt Learning:** Advancing beyond discrete token selection, continuous prompt learning methods optimize continuous vectors in the word embedding space [27, 29, 52]. Instead of selecting specific words, these methods adjust continuous embeddings—often referred to as “soft prompts”—that guide the model’s responses. While this approach offers greater flexibility and can enhance performance, it introduces challenges in interpretability since the continuous vectors do not correspond directly to human-readable words.

For those interested in a comprehensive overview of prompt learning approaches and their applications in NLP, please refer to the survey of Liu et al. for insights, theoretical foundations, and practical implications into various prompting methods [31]. In this work, CoOp extends prompt learning to computer vision and contributes to the broader goal of democratizing foundation models.

5.3 Preliminaries

A common approach for building a visual recognition system involves training vision models to predict a fixed set of object categories using discrete labels [7, 15]. From a technical point of view, this is achieved by aligning image features—generated by models like ResNet [15] or ViT [7]—with a predefined set of weights that represent visual concepts and are initialized with random parameters. Though training categories often have textual descriptions such as “goldfish” or “toilet paper,” these are converted into discrete labels to simplify the computation of the cross-entropy loss. However, such a learning paradigm confines the recognition system to closed-set visual concepts, making it unable to handle new categories without additional data for training a new classifier.

The emergence of contrastive vision-language models, such as CLIP [37], has dramatically changed the landscape, allowing new categories to be synthesized on-the-fly without additional model training. Further we introduce the background on CLIP, which covers the architecture, how the model is pre-trained, and how to perform zero-shot inference with the model.

5.3.1 *Model Architecture*

CLIP is composed of two distinct encoders, one dedicated to processing images and the other to handling text. The image encoder's primary function is to transform high-dimensional image data into a compact, low-dimensional vector. This encoder can be implemented using various architectures, such as a Convolutional Neural Network (CNN) like ResNet-50 [15] or a Vision Transformer (ViT) [7]. Conversely, the text encoder leverages the Transformer architecture [45] to produce representations of natural language text. This dual-encoder setup enables CLIP to effectively bridge the gap between visual and textual information.

Specifically, when provided with a sequence of words (tokens) such as “a photo of a dog,” CLIP initially transforms each token (including punctuation) into its corresponding lower-cased byte pair encoding (BPE) representation [39], which is essentially a unique numeric identifier. The vocabulary size defined in CLIP is 49,152. For efficient minibatch processing, each text sequence is enclosed with the [SOS] (start of sequence) and [EOS] (end of sequence) tokens and is limited to a fixed length of 77 tokens. Subsequently, these numeric IDs (known as tokens) are converted into 512-dimensional word embedding vectors, which are then fed into the Transformer model. Finally, the features corresponding to the [EOS] token position are layer-normalized and further refined through a linear projection layer.

5.3.2 *Pre-training*

CLIP is designed to align the embedding spaces learned independently for images and text through contrastive learning, which ensures that the model effectively learns meaningful associations between visual and textual data. Specifically, given a batch of image–text pairs, CLIP aims to maximize the cosine similarity for matched pairs—those pairs where the image and text correspond to each other, while simultaneously minimizing the cosine similarity for all unmatched pairs, where the image and text do not correspond.

To achieve this, CLIP leverages a massive and diverse training dataset comprising 400 million image–text pairs. This extensive dataset helps the model learn a wide array of visual concepts, enhancing its ability to generalize and transfer knowledge to various downstream tasks.

This contrastive training process ultimately results in a powerful model capable of performing a variety of tasks, from zero-shot classification to more nuanced image and text understanding tasks.

5.3.3 Zero-Shot Inference

Since CLIP is pre-trained to determine whether an image matches a given textual description, it is inherently well-suited for zero-shot recognition tasks. This capability is leveraged by comparing the features extracted from an image with the classification weights generated by the text encoder.

Formally, let $\mathbf{f}$ be image features extracted by the image encoder for an image $\mathbf{x}$ and $\{\mathbf{w}_i\}_{i=1}^K$ a set of weight vectors generated by the text encoder. K denotes the number of classes and each $\mathbf{w}_i$ is derived from a prompt that could have the form of “a photo of a [CLASS]” where the class token is replaced by the specific class name, such as “cat,” “dog” or “car.” The prediction probability is then computed as

$$p(y = i | \mathbf{x}) = \frac{\exp(\cos(\mathbf{w}_i, \mathbf{f})/\tau)}{\sum_{j=1}^K \exp(\cos(\mathbf{w}_j, \mathbf{f})/\tau)}, \quad (5.1)$$

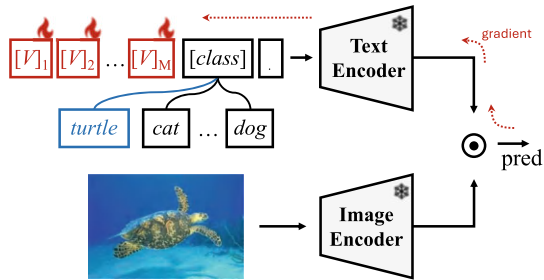
where τ is a temperature parameter learned by CLIP, and $\cos(\cdot, \cdot)$ denotes cosine similarity.

In contrast to the traditional classifier learning approach, where closed-set visual concepts are learned from random vectors, vision-language pre-training enables the exploration of open-set visual concepts through a high-capacity text encoder. This approach broadens the semantic space, resulting in learned representations that are more versatile and transferable to a wide range of downstream tasks.

5.4 Context Optimization

Context Optimization (CoOp) eliminates the need for manual prompt tuning by representing context words as continuous vectors, which are optimized end-to-end using data while keeping the extensive pre-trained parameters frozen. An overview of the approach is shown in Fig. 5.2. CoOp includes two variations to address tasks of different types: a unified context approach, which uses a shared context across all classes and performs effectively in most scenarios, and a class-specific context

Fig. 5.2 Overview of Context Optimization (CoOp). The core concept involves modeling a prompt’s context using a set of learnable vectors that can be optimized by minimizing the classification loss and keeping the pre-trained model weights frozen



approach, which assigns a unique set of context tokens to each class, making it more suitable for fine-grained categories.

5.4.1 Unified Context

We first discuss unified context, which essentially shares the same context with all categories. Specifically, the prompt given to the text encoder $g(\cdot)$ is designed with the following form,

$$\mathbf{t} = [\mathbf{V}]_1[\mathbf{V}]_2 \dots [\mathbf{V}]_M[\text{CLASS}], \quad (5.2)$$

where each $[\mathbf{V}]_m$ ($m \in \{1, \dots, M\}$) is a vector with the same dimension as word embeddings (i.e., 512 for CLIP), and M is a hyperparameter, which specifies the number of context tokens.

By forwarding a prompt $\mathbf{t}$ to the text encoder $g(\cdot)$, we can obtain a classification weight vector representing a visual concept (still from the [EOS] token position). The prediction probability is computed as

$$p(y = i|\mathbf{x}) = \frac{\exp(\cos(g(\mathbf{t}_i), \mathbf{f})/\tau)}{\sum_{j=1}^K \exp(\cos(g(\mathbf{t}_j), \mathbf{f})/\tau)}, \quad (5.3)$$

where the class token within each prompt $\mathbf{t}_i$ is replaced by the corresponding word embedding vector(s) of the i -th class name.

5.4.2 Class-Specific Context

The second version is based on class-specific context (CSC) where context vectors are independent to each class, i.e., $[\mathbf{V}]_1^i[\mathbf{V}]_2^i \dots [\mathbf{V}]_M^i \neq [\mathbf{V}]_1^j[\mathbf{V}]_2^j \dots [\mathbf{V}]_M^j$ for $i \neq j$ and $i, j \in \{1, \dots, K\}$. As an alternative to unified context, we find that CSC is particularly useful for some fine-grained classification tasks.

5.4.3 Context Update

Context learning can be performed by minimizing the standard cross-entropy loss (when dealing with classification problems). The gradients can be back-propagated through the text encoder $g(\cdot)$ and all the way to the context vectors. Such a gradient flow makes use of the rich knowledge encoded in the text encoder's parameters. The

differentiable nature of the method allows full exploration in the word embedding space, which facilitates learning of task-relevant context.

This approach specifically addresses the emerging problem of the adaptation of large vision-language models. There are some differences that distinguish this approach from the prompt learning methods developed in NLP for language models. First, the backbone architectures are clearly different for CLIP-like models and language models: the former take both visual and textual data as input and produce alignment scores for image recognition, while the latter are tailored to handle textual data only. Second, the pre-training objectives are different, i.e., contrastive learning vs autoregressive learning, which lead to different model behaviors and thus require different module designs.

5.5 Experiments

5.5.1 Datasets and Setup

Following CLIP [37], we use 11 image classification datasets for evaluation: ImageNet [5], Caltech101 [9], OxfordPets [35], StanfordCars [25], Flowers102 [34], Food101 [2], FGVCAircraft [33], SUN397 [47], DTD [4], EuroSAT [16], and UCF101 [43]. These datasets cover a diverse set of image recognition tasks, which include classification on generic objects, scenes, actions, and fine-grained categories, as well as specialized tasks like recognizing textures and satellite imagery. The detailed descriptions for these datasets are provided in Table 5.1. CoOp is trained using 1, 2, 4, 8, and 16 shots, respectively, and the models are deployed in the full test sets. The average results over three runs are reported for comparison.³

For the domain generalization experiments, the source dataset is ImageNet. The target datasets are ImageNetV2 [38], ImageNet-Sketch [46], ImageNet-A [19], and ImageNet-R [18], all of which have compatible class names with ImageNet allowing seamless transfer for the prompts learned by CoOp. ImageNetV2 is a reproduced test set using different sources while following ImageNet’s data collection process. ImageNet-Sketch contains sketch images belonging to the same 1,000 ImageNet classes. Both ImageNet-A and -R contain 200 classes derived from a subset of ImageNet’s 1,000 classes. The former consists of real-world adversarially filtered images that cause current ImageNet classifiers to produce low results, whereas the latter features a rendition of the ImageNet classes in diverse image styles such as paintings, cartoons, and sculptures.

³ Note that for Caltech101, the “BACKGROUND_Google” and “Faces_easy” classes are discarded; for the video dataset UCF101, the middle frame of each video is used as input to the image encoder.

Table 5.1 Datasets and their descriptions and hand-crafted prompts used in CLIP

Dataset	Description	Hand-crafted prompt
ImageNet	Object recognition	“a photo of a [CLASS].”
Caltech101	Object recognition	“a photo of a [CLASS].”
OxfordPets	Pet breed recognition	“a photo of a [CLASS], a type of pet.”
StanfordCars	Car model classification	“a photo of a [CLASS].”
Flowers102	Flower classification	“a photo of a [CLASS], a type of flower.”
Food101	Food category recognition	“a photo of [CLASS], a type of food.”
FGVCAircraft	Aircraft type classification	“a photo of a [CLASS], a type of aircraft.”
SUN397	Scene recognition	“a photo of a [CLASS].”
DTD	Texture recognition	“[CLASS] texture.”
EuroSAT	Satellite image classification	“a centered satellite photo of [CLASS].”
UCF101	Action recognition	“a photo of a person doing [CLASS].”
ImageNetV2	ImageNet variant	“a photo of a [CLASS].”
ImageNet-Sketch	Sketch image classification	“a photo of a [CLASS].”
ImageNet-A	Adversarial image classification	“a photo of a [CLASS].”
ImageNet-R	ImageNet variant (non-photos)	“a photo of a [CLASS].”

Two baselines are included in the experiments: zero-shot CLIP and the linear probe model. For zero-shot CLIP, we follow the prompt engineering guidelines suggested by Radford et al. [37] to design prompts. For generic objects and scenes, “a photo of a [CLASS].” is adopted. For fine-grained categories, task-relevant context is added like “a type of pet” for OxfordPets and “a type of food” for Food101. When it comes to specialized tasks such as recognizing textures in DTD, the prompt is customized as “[CLASS] texture” where the class names are adjectives like “bubbly” and “dotted.” The prompts used for the 11 datasets are detailed in Table 5.1.

As shown in previous work [37, 44], the linear probe model, i.e., training a linear classifier on top of pre-trained features, can achieve performance that is on a par with that of state-of-the-art few-shot learning methods (which are often more sophisticated). We follow Radford et al. [37] to train the linear probe model.

Unless otherwise stated, ResNet-50 [15] is used as the image encoder’s backbone and the number of context tokens M is set to 16. All models are built on top of CLIP’s open-source code.⁴ CoOp’s context vectors are randomly initialized by drawing from a zero-mean Gaussian distribution with standard deviation of 0.02. SGD is used as the optimizer, and the initial learning rate is set to 0.002, which is decayed by the cosine annealing rule. The maximum epoch is set to 200 for 16/8 shots, 100 for 4/2 shots, and 50 for 1 shot (except for ImageNet where the maximum epoch is fixed to 50). To mitigate explosive gradients observed in the early training iterations, we use a warmup trick by fixing the learning rate to $1e-5$ (only during the first epoch).

⁴ <https://github.com/openai/CLIP>.

5.5.2 *Comparison with Hand-Crafted Prompts*

Figure 5.3 presents a summary of our key findings. The default model in our study, which combines CLIP with CoOp, positions the class token at the end. On average, CoOp demonstrates strong few-shot learning capabilities, requiring just two labeled examples per class to achieve a big margin over CLIP. With 16 examples per class, CoOp further widens the performance gap, achieving an average improvement of approximately 15%. Figure 5.4 highlights the absolute performance gains of CoOp (using 16 examples per class) compared to manually designed prompts. Significant improvements are observed in specialized tasks, such as EuroSAT and DTD, with performance gains exceeding 45% and 20%, respectively. Strong improvements are also seen in fine-grained tasks like Flowers102, StanfordCars, and FGVCAircraft, as well as in scene and action recognition tasks (e.g., SUN397 and UCF101). For the challenging ImageNet dataset, which includes 1,000 categories, the 4.77% improvement is noteworthy. However, gains on two fine-grained datasets, OxfordPets and Food101, are less pronounced. The limited improvement on Food101 appears to result from noisy training data with overly vibrant colors and occasional label errors. Analysis of performance trends on OxfordPets and Food101 (Fig. 5.3) reveals diminishing returns as more training data are used, suggesting potential overfitting. Addressing this issue may require stronger regularization techniques, such as increased weight decay. The overall results highlight CoOp’s ability to learn task-relevant prompts effectively and efficiently, even with limited data.

5.5.3 *Comparison with Linear-Probing*

In terms of overall performance (Fig. 5.3, top-left), CoOp exhibits a clear advantage over the linear probe model. The linear probe requires more than four shots on average to achieve the same performance as the zero-shot model, whereas CoOp already demonstrates impressive gains at just four shots. Notably, the performance gaps in the extreme low-data regime, such as for one or two shots, are considerably larger, highlighting CoOp’s effectiveness compared to learning a linear classifier from scratch in few-shot learning scenarios. Additionally, we observe that the linear probe model performs similarly to CoOp on the two specialized tasks (DTD and EuroSAT), as well as on a few fine-grained datasets (Flowers102 and FGVCAircraft). This is not unexpected as the pre-trained CLIP feature space has been shown to be powerful, positioning the linear probe model as a strong contender. However, CoOp’s class-specific context version outperforms the linear probe model on these datasets and demonstrates significantly better potential as more shots are made available. Later, we will show that CoOp achieves much stronger performance than the linear probe model in domain generalization.

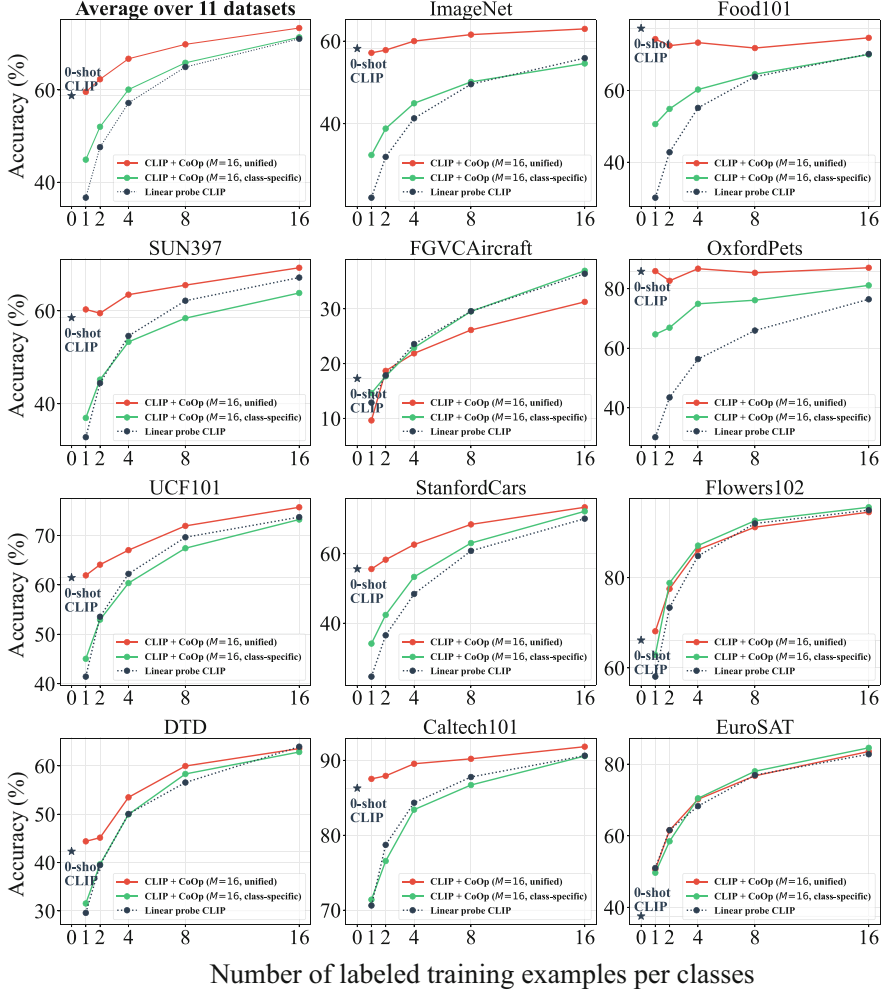


Fig. 5.3 Main results of few-shot learning across 11 datasets. CoOp demonstrates its effectiveness in enhancing CLIP’s few-shot learning capabilities, achieving substantial improvements over zero-shot CLIP and outperforming the linear probe approach

5.5.4 Comparison with Prompt Ensembling

The authors of CLIP [37] have suggested that additional improvements can be achieved by ensembling multiple zero-shot classifiers, each generated using different hand-crafted prompts such as “a photo of the large [CLASS],” “a bad photo of the [CLASS],” and “an origami [CLASS],” which vary in scale, viewpoint, and abstraction, respectively. We aim to explore whether the prompts learned by CoOp can still offer advantages compared to prompt ensembling. For a fair

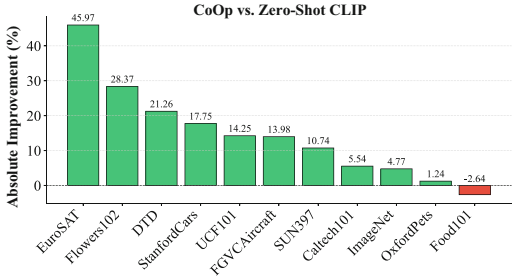


Fig. 5.4 Absolute improvement (%) of CoOp over CLIP across various datasets. The results demonstrate substantial gains in performance across multiple benchmark tasks, showcasing the effectiveness and versatility of the CoOp method in enhancing accuracy across a wide range of domains

Table 5.2 Comparison of different ways to build prompts. Prompt learning is clearly the winner

	Accuracy
Prompt engineering	58.18
Prompt ensembling	60.41
Prompt learning (CoOp)	62.95

comparison, we use the prompts selected by Radford et al. [37], which have been extensively fine-tuned on ImageNet, to construct the ensemble classifier. The results in Table 5.2 demonstrate the superiority of CoOp. Given the potential of prompt ensembling, future work could investigate how to enhance CoOp from an ensembling perspective.

5.5.5 Comparison with Fine-Tuning Methods

In addition to linear probe, we also evaluate CoOp against alternative fine-tuning methods: (1) fine-tuning CLIP’s image encoder; (2) optimizing a transformation layer appended to the output of the text encoder; and (3) optimizing a bias term added to the text encoder’s output. The comparative results, presented in Fig. 5.5, reveal that fine-tuning the image encoder yields limited improvement. While adding a transformation layer provides a modest boost over the zero-shot model, it remains insufficient. Optimizing a bias term produces encouraging results but still falls significantly short of CoOp’s performance. This highlights that gradients flowing through the text encoder in CoOp capture richer and more informative signals.

Fig. 5.5 Comparison between different fine-tuning methods and the zero-shot model on ImageNet. Δ denotes the difference with the zero-shot model

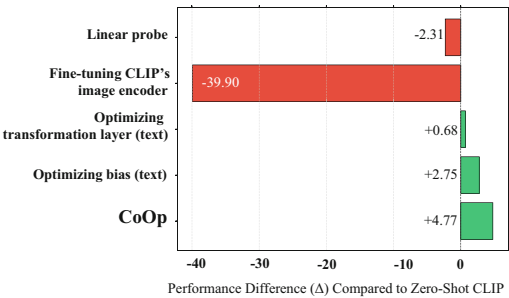


Table 5.3 Results on domain generalization benchmarks. Despite being a learning-based approach, CoOp does not overfit the source domain, demonstrating robust domain generalization performance. Interestingly, learning fewer context tokens leads to better results

	Source	Target			
	ImageNet	-V2	-Sketch	-A	-R
Zero-Shot CLIP	58.18	51.34	33.32	21.65	56.00
Linear Probe CLIP	55.87	45.97	19.07	12.74	34.86
CoOp ($M = 16$)	62.95	55.11	32.74	22.12	54.96
CoOp ($M = 4$)	63.33	55.40	34.67	23.06	56.60

5.5.6 Domain Generalization Performance

Table 5.3 presents the domain generalization results. Remarkably, CoOp improves CLIP’s robustness to distribution shifts, even though it is trained with exposure to the source dataset. This indicates that the learned prompts possess strong generalization capabilities. Notably, fewer context tokens further enhance robustness, highlighting the efficiency of concise prompts. In contrast, the linear probe model performs significantly worse on these target datasets, underscoring its limitations in domain generalization.

5.5.7 How to Set Context Length

Determining the optimal number of context tokens is a key aspect of prompt learning. Does having more context tokens lead to better performance? According to the results in Sect. 5.5.6, shorter context lengths seem to benefit domain generalization, likely due to reduced overfitting as fewer parameters are learned. To investigate this further for source datasets, we conduct experiments on 11 datasets, varying the context length from 4 to 8 to 16 tokens. The average results, shown in Table 5.4, indicate that increasing the number of context tokens generally enhances performance. In conclusion, there is no definitive rule for selecting the perfect context length. Practitioners must balance between achieving high performance and

Table 5.4 Results of using different context lengths (shots=16)

	Context Length		
	4	8	16
CoOp	72.65	73.20	73.42

ensuring robustness to distribution shifts when choosing the appropriate context length for their specific applications.

5.5.8 *How to Initialize Context*

An important consideration in prompt learning is how to initialize the context vectors that guide the model’s adaptation to new tasks. There are two common methods: random initialization and manual initialization. In manual initialization, we use the embeddings of a meaningful phrase—in this case, “a photo of a”—to initialize the context vectors across 11 datasets. For a fair comparison, we set the context length to 4 in both methods. The observation is that the choice between random and manual initialization does not significantly affect performance (with less than 0.01% difference in average). Although fine-tuning the initialization phrases might yield slight improvements, in practice, using simple random initialization is sufficient and more convenient. This approach allows practitioners to focus on other aspects of model optimization without the need to carefully select initialization phrases.

5.5.9 *Visualization and Interpretation*

Interpreting the learned prompts is difficult because the context vectors are optimized in a continuous space. We resort to an indirect way by searching within the vocabulary for words that are closest to the learned vectors based on the Euclidean distance. Note that CLIP [37] uses the BPE representation [39] for tokenization, so the vocabulary includes subwords that frequently appear in text, such as “hu” (subsumed by many words like “hug” and “human”). Table 5.5 shows the searched results on some datasets. We observe that a few words are somewhat relevant to the tasks, such as “enjoyed” on Food101, “pretty” on DTD, and “fluffy” and “paw” on OxfordPets. But when connecting all the nearest words together, the prompts do not make much sense. We also observe that when using manual initialization (like “a photo of a”), the nearest words for the converged vectors are mostly the ones used for initialization. We conjecture that the learned vectors might encode meanings that are beyond the existing vocabulary. Overall, we are unable to draw any firm conclusion based on the observations because using nearest words to interpret the learned prompts could be inaccurate—the semantics of the vectors is not necessarily correlated with the nearest words.

Table 5.5 The nearest words for each of the 16 context vectors learned by CoOp on different datasets, with their distances shown in parentheses. N/A means non-Latin characters. Table adapted from [53]

#	ImageNet	Food101	DTD	OxfordPets
1	potd (1.7136)	lc (0.6752)	boxed (0.9433)	tosc (2.5952)
2	that (1.4015)	enjoyed (0.5305)	seed (1.0498)	judge (1.2635)
3	filmed (1.2275)	beh (0.5390)	anna (0.8127)	fluffy (1.6099)
4	fruit (1.4864)	matches (0.5646)	mountain (0.9509)	cart (1.3958)
5	,... (1.5863)	nytimes (0.6993)	eldest (0.7111)	harlan (2.2948)
6	° (1.7502)	prou (0.5905)	pretty (0.8762)	paw (1.3055)
7	excluded (1.2355)	lower (0.5390)	faces (0.7872)	incase (1.2215)
8	cold (1.4654)	N/A	honey (1.8414)	bie (1.5454)
9	stery (1.6085)	minute (0.5672)	series (1.6680)	snuggle (1.1578)
10	warri (1.3055)	~ (0.5529)	coca (1.5571)	along (1.8298)
11	marvelcomics (1.5638)	well (0.5659)	moon (1.2775)	enjoyment (2.3495)
12	∴ (1.7387)	ends (0.6113)	lh (1.0382)	jt (1.3726)
13	N/A	mis (0.5826)	won (0.9314)	improving (1.3198)
14	lotion (1.5015)	somethin (0.6041)	replied (1.1429)	srsly (1.6759)
15	muh (1.4985)	seminar (0.5274)	sent (1.3173)	asteroid (1.3395)
16	.# (1.9340)	N/A	piedmont (1.5198)	N/A

5.6 Conclusion

Large pre-trained vision-language models have demonstrated remarkably strong capabilities across a variety of downstream applications. These models, often referred to as vision foundation models due to their “critically central yet incomplete” nature [1], require adaptation through automated techniques to enhance performance and efficiency in specific tasks. This chapter provides insights into transforming CLIP-like models into data-efficient learners through the use of prompt learning. Despite being a learning-based method, CoOp performs favorably against manual prompts in domain generalization tasks. This finding highlights the potential of prompt learning for optimizing large multimodal models.

Further Reading

Interested readers may refer to the following material:

- [54] for a comprehensive study into the generalization problem of vanilla prompt learning methods
- [24, 49] for multimodal prompt learning methods, which strike a balance between performance and training efficiency
- [21] for prompt learning in the image pixel space

- [50] for a comprehensive survey on vision-language models and their applications to computer vision tasks

References

1. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
2. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101—mining discriminative components with random forests. In: ECCV (2014)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
4. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: CVPR (2014)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
6. Desai, K., Johnson, J.: Virtex: learning visual representations from textual annotations. In: CVPR (2021)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
8. Elhoseiny, M., Saleh, B., Elgammal, A.: Write a classifier: zero-shot learning using purely textual descriptions. In: ICCV (2013)
9. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: CVPR-W (2004)
10. Frome, A., Corrado, G., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: NeurIPS (2013)
11. Fürst, A., Rumetshofer, E., Tran, V., Ramsauer, H., Tang, F., Lehner, J., Kreil, D., Kopp, M., Klambauer, G., Bitto-Nemling, A., et al.: Cloob: modern hopfield networks with infoloob outperform clip. arXiv preprint arXiv:2110.11316 (2021)
12. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723 (2020)
13. Gomez, L., Patel, Y., Rusiñol, M., Karatzas, D., Jawahar, C.: Self-supervised learning of visual features through embedding images into text topic spaces. In: CVPR (2017)
14. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
16. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **12**(7), 2217–2226 (2019)
17. Hénaff, O.J., Srinivas, A., Fauw, J.D., Razavi, A., Doersch, C., Eslami, S.M.A., van den Oord, A.: Data-efficient image recognition with contrastive predictive coding. In: ICML (2020)
18. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: ICCV (2021)
19. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: CVPR (2021)

20. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)
21. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. arXiv preprint arXiv:2203.12119 (2022)
22. Jiang, Z., Xu, F.F., Araki, J., Neubig, G.: How can we know what language models know? ACL **8**, 423–438 (2020)
23. Joulin, A., Van Der Maaten, L., Jabri, A., Vasilache, N.: Learning visual features from large weakly supervised data. In: ECCV (2016)
24. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: multi-modal prompt learning. In: CVPR, pp. 19113–19122 (2023)
25. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: ICCV-W (2013)
26. Lei Ba, J., Swersky, K., Fidler, S., et al.: Predicting deep zero-shot convolutional neural networks using textual descriptions. In: ICCV (2015)
27. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)
28. Li, A., Jabri, A., Joulin, A., van der Maaten, L.: Learning visual n-grams from web data. In: ICCV (2017)
29. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
30. Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., Yan, J.: Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. arXiv preprint arXiv:2110.05208 (2021)
31. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021)
32. Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., Tang, J.: GPT understands, too. arXiv preprint arXiv:2103.10385 (2021)
33. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013)
34. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: ICVGIP (2008)
35. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: CVPR (2012)
36. Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.H., Riedel, S.: Language models as knowledge bases? In: EMNLP (2019)
37. 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)
38. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: ICML (2019)
39. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: ACL (2016)
40. Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: Autoprompt: eliciting knowledge from language models with automatically generated prompts. In: EMNLP (2020)
41. Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., Kiela, D.: Flava: A foundational language and vision alignment model. arXiv preprint arXiv:2112.04482 (2021)
42. Socher, R., Ganjoo, M., Sridhar, H., Bastani, O., Manning, C.D., Ng, A.Y.: Zero-shot learning through cross-modal transfer. In: NeurIPS (2013)
43. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
44. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking few-shot image classification: a good embedding is all you need? In: ECCV (2020)

45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: *NeurIPS* (2017)
46. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. In: *NeurIPS* (2019)
47. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: *CVPR* (2010)
48. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432* (2021)
49. Zang, Y., Li, W., Zhou, K., Huang, C., Loy, C.C.: Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225* (2022)
50. Zhang, J., Huang, J., Jin, S., Lu, S.: Vision-language models for vision tasks: a survey. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024)
51. Zhang, Y., Jiang, H., Miura, Y., Manning, C.D., Langlotz, C.P.: Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747* (2020)
52. Zhong, Z., Friedman, D., Chen, D.: Factual probing is [mask]: learning vs. learning to recall. In: *NAACL* (2021)
53. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134* (2021)
54. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: *CVPR*, pp. 16816–16825 (2022)

Chapter 6

Test-Time Prompt Tuning for Vision-Language Models



Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein,
Anima Anandkumar, and Chaowei Xiao

Abstract Pretrained contrastive vision-language models (VLMs) (e.g., CLIP) have shown promising zero-shot generalization in many downstream tasks (e.g., image classification) with properly designed text *prompts*. Instead of relying on hand-engineered prompts, *prompt tuning* learns prompts using the training data from the downstream data distribution. Although effective, training on domain-specific data hurts a model’s generalization capability to unseen new domains. In this chapter, we discuss test-time prompt tuning (TPT), a method that learns adaptive prompts on the fly using a single test sample. In the case of image classification, TPT optimizes the prompt by minimizing entropy with confidence selection so that the model has consistent predictions across different augmented views of each test sample. When evaluated on natural distribution shifts, TPT surpasses previous prompt tuning methods that require additional task-specific training data. When evaluated under the cross-dataset generalization setting, TPT performs on par with the state-of-the-art methods that use additional training data.

Keywords Test-time adaptation · Test-time prompt tuning · Prompt learning · Vision-language model · Unsupervised learning

M. Shu (✉) · T. Goldstein
University of Maryland, College Park, MD, USA
e-mail: manlis@umd.edu; tomg@umd.edu

W. Nie · D.-A. Huang · Z. Yu
Nvidia Corporation, Santa Clara, CA, USA
e-mail: wnie@nvidia.com; deahuang@nvidia.com; zhidingy@nvidia.com

A. Anandkumar
California Institute of Technology, Pasadena, CA, USA
e-mail: arangelf@caltech.edu

C. Xiao
University of Wisconsin, Madison, WI, USA
e-mail: cxiao34@wisc.edu

6.1 Introduction

Advances in contrastive vision-language pretraining, such as CLIP [1], present a promising direction for developing foundation models for visual recognition [2, 3]. These foundation models encode a wide range of visual concepts after training on millions of noisy image–text pairs and can be applied to downstream tasks in a zero-shot manner without task-specific training data [4–10]. This is made possible by appropriately designed instruction prompts. Take image classification in Fig. 6.1 as an example: We can prepend a category name with a prompt “a photo of a” (e.g., “a photo of a dog”). Images can then be classified by using CLIP to measure their alignment with the various class descriptions. Designing such prompts thus plays a crucial role in applying foundation models to downstream tasks in a zero-shot manner. However, such hand-crafted prompts require domain-specific heuristics and may not be optimal.

Existing works address this by proposing *prompt tuning* to directly learn prompts using training data from downstream tasks [11]. We can fine-tune prompts with training data in the same way we fine-tune model parameters since prompt embeddings are part of the model input and are differentiable with respect to the loss function. Such an approach can find better prompts compared to hand-crafted ones, but the learned prompts are limited to the distribution and tasks corresponding to training data and may have limited generalization beyond that. In addition, this approach requires training data with annotations, which can be expensive and is not available for zero-shot tasks.

To improve the zero-shot generalization in pretrained VLMs, we introduce **test-time prompt tuning (TPT)** that tunes the prompt on the fly using only the given

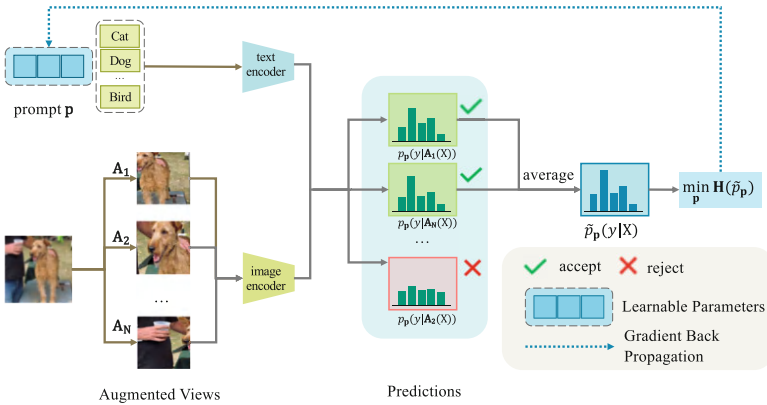


Fig. 6.1 Test-time Prompt Tuning (TPT) for image classification. We tune adaptive prompts on the fly with a single test sample, without the need for additional training data or annotations. TPT optimizes the prompt to encourage consistent predictions across augmented views by minimizing the marginal entropy. We introduce *confidence selection* to filter out noisy augmentations

test sample. The tuned prompt is adapted to each task, making it suitable for zero-shot generalization without requiring any task-specific training data or annotations. TPT retains the zero-shot generalization setting, as no additional training data or annotations are used.

We study TPT on two different downstream tasks: image classification and context-dependent visual reasoning. For each downstream task, we design a customized test-time tuning strategy that fits the nature of the task. Without loss of generality, we consider CLIP [1] as our foundation model, which is a representative contrastive VLM with a simple design and wide downstream applications [12].

For image classification, a test sample is an input image. Given a single sample at test time, we perform prompt tuning by generating multiple randomly augmented views, and optimizing the text prompt so that the model has consistent predictions across different augmented views. This is done by minimizing the marginal entropy among the outputs of the augmented views. In addition, since some augmentations may lead to misleading model predictions, we introduce *confidence selection* to filter out those “noisy” augmented views. We discard augmented views with high prediction entropy (i.e., low confidence) and only include high confidence views in the consistency optimization.

We evaluate the zero-shot generalization of TPT in two image classification settings: natural distribution shifts [13] and cross-dataset generalization [14]. We use ImageNet variants [13, 15–17] as the natural distribution shifts to the original ImageNet [18] data. For cross-dataset generalization, we consider 10 different fine-grained image classification tasks and compare TPT with existing methods that require training on downstream data.

For the second task of context-dependent visual reasoning, such as Bongard-HOI [19], a test sample contains two sets of support images and a query image for evaluation. The two sets of support images exemplify the presence and the absence of a human–object interaction (HOI) concept (e.g., “ride bike”). The model is then asked to infer whether the query image contains the underlying concept. Given such a test sample, we apply TPT by tuning prompts to better differentiate between the two support sets, so that the query image can be better classified (Fig. 6.4). Despite the use of support sets, our approach is still considered zero-shot for visual reasoning, because we do not use either training tasks from other concepts or the annotation of the query image at test time to update the prompt of the test task.

6.2 Related Work

6.2.1 Prompting for Foundation Models

Foundation models are those trained on large-scale heterogeneous data, of which the knowledge can be transferred to various downstream tasks in natural language processing [20, 21], computer vision [1, 22, 23], etc. Existing work has proposed

different ways to efficiently and effectively transfer such knowledge to downstream task [24–27]. Prompting is a heuristic way to directly apply foundation models to downstream tasks in a zero-shot manner. Prompt works as a part of the text input that instructs the model to perform accordingly on a specific task. However, such zero-shot generalization is highly dependent on a well-designed prompt. Prompt tuning [11, 28–30] proposes to learn the prompt from downstream data in the continual input embedding space, which presents a parameter-efficient way of fine-tuning foundation models. Although initially developed for language models, prompt tuning has later been applied to other domains, including VLMs [14, 31, 32] and continual learning [33]. CoOp[31] applies prompt tuning to CLIP. By tuning the prompt on a collection of training data, CoOp effectively improves CLIP’s performance on the corresponding downstream tasks. CoCoOp [14] points out that CoOp lacks in generalization to out-of-distribution data and proposes to alleviate the problem by making the prompt conditioned on model inputs. Despite being effective on the given task, this line of work requires access to downstream training data with annotations, restricting the zero-shot knowledge transfer of foundation models. Another line of work proposes to tune the prompt in an unsupervised manner [34, 35]. However, it requires access to *multiple* samples from either the training or testing split. In this chapter, we discuss test-time prompt tuning that works on a *single* test sample. Our method can directly work with the zero-shot applications of foundation models.

6.2.2 Generalization Under Data Distribution Shifts

A reliable machine learning model is supposed to perform well under data distribution shifts for real-world applications. For a model trained on a given set of data, distribution shift refers to the discrepancy between the underlying distributions of the test and the training data. Distribution shifts can occur naturally in the real world due to variations in the environment [36] or the encounter of unseen concepts [37]. For example, in the meta-learning literature [38], each test sample consists of a novel task (i.e., distribution), and the models should be able to quickly adapt to the novel distributions. Even in the standard evaluation protocol for machine learning models, there exists a subtle difference in the data distribution between the training and testing splits [39, 40], which is also one type of distribution shift. Pretrained contrastive VLMs like CLIP can generalize to downstream tasks with various distribution shifts in a zero-shot manner. Such zero-shot generalization ability presents a promising direction for realizing reliable and generic machine learning models. The method discussed in this chapter aims to improve CLIP toward a better generic model, instead of adapting it to specific downstream tasks or target datasets. We leverage the assumption that a robust model should have decision boundaries lying in low-density data regions [41]. Consistency-regularization-based methods [23, 42] achieve this goal by making the network outputs invariant to small

input noises. For classification tasks, we use consistency regularization as our test-time prompt tuning objective with the confidence selection module.

6.2.3 *Test-Time Optimization*

The idea of adapting machine learning models to test samples on the fly has been applied to different tasks [43–46]. This chapter mainly focuses on applying the technique to improve model generalization. One challenge in this area is to design a practical test-time objective. Test-time training and its variants [47, 48] modify the training objective and the network architecture by adding a self-supervised multitask branch, which will then be used at test time for computing optimization objectives and adapts the network to the test sample. Entropy minimization is another common technique for developing self-supervised objectives [49, 50]. TENT [51] performs test-time optimization by minimizing the entropy of the batch-wise prediction probability distributions, but it needs more than one test sample to get a nontrivial solution. Zhang et al. [52] propose marginal entropy minimization that works on a single test sample with data augmentations. Another major challenge is to choose the right parameter group for optimization. Batch normalization (BN) layers have been shown to capture the domain discrepancies in image data [53, 54]. It is a straightforward way to directly adapt the BN statistics at test time to enhance model robustness against distribution shifts [55]. However, adapting BN layers puts restrictions on model architectures. Another choice is to update the feature extractor while freezing the prediction module [47, 56]. Zhang et al. [52] show that optimizing the entire model at test time can work as well. Our method addresses both of the challenges aforementioned. For the choice of parameter group, we optimize the text prompt while keeping the model intact. Our motivation is to avoid distorting the pretrained features and to preserve the zero-shot generalization ability of pretrained models. In Sect. 6.5, we empirically show that the prompt works as the most effective parameter group for CLIP. Different from the previous single-point method [52], we refine the entropy minimization by proposing *confidence selection*, which helps filter out noisy augmentations that may lead to misleading predictions.

6.3 TPT: Test-Time Prompt Tuning

In this section, we first discuss how to apply CLIP to downstream tasks in a zero-shot manner with a hand-crafted prompt. Next, we briefly review existing prompt tuning approaches for CLIP using downstream training data. Finally, we give detailed introductions on how to apply our method to the image classification task and context-dependent visual reasoning, respectively.

6.3.1 Background

Contrastive Language-Image Pretraining (CLIP) CLIP consists of two parallel encoders: one that maps the text input into a feature vector, and the other does the same for the image input. The model is trained with a contrastive loss that promotes similarity between the two vectors so that the text and image align in the feature space. We denote a CLIP model as $\mathcal{F} = \{\mathbf{E}_{\text{visual}}, \mathbf{E}_{\text{text}}\}$, with $\mathbf{E}_{\text{visual}}$ and $\mathbf{E}_{\text{text}}$ being the image and text encoders.

We first review how to apply CLIP to downstream tasks in a zero-shot manner with a hand-crafted prompt. We take image classification as an example. Consider a single test image X_{test} of class y , where $X \in \mathbb{R}^{C \times H \times W}$ and $y \in \mathbb{R}^K$ for a K -class classification problem. In the baseline zero-shot setting, we prepend a hand-crafted prompt prefix, such as $\mathbf{p}$ = “a photo of a,” to every y_i in $\mathcal{Y} = \{y_1, y_2, \dots, y_K\}$ to form the category-specific text inputs $\{\mathbf{p}; y_i\}$. We then feed these class descriptions to the text encoder to get the text features $\{t_1, t_2, \dots, t_K\}$, where $t_i = \mathbf{E}_{\text{text}}(\{\mathbf{p}; y_i\})$. Each text feature t_i is paired with the image feature $\mathbf{v} = \mathbf{E}_{\text{visual}}(X)$ to compute a similarity score $s_i = \text{sim}(t_i \cdot \mathbf{v})$, where $\text{sim}(\cdot)$ denotes the cosine similarity. The prediction probability on X can be denoted by $p(y_i|X) = \frac{\exp(\text{sim}(t_i \cdot \mathbf{v})\tau)}{\sum_{i=1}^K \exp(\text{sim}(t_i \cdot \mathbf{v})\tau)}$, where τ is the temperature of the softmax function.

Prompt Tuning Using Downstream Training Data Instead of using a hand-crafted prompt, prompt tuning methods train a prompt to maximize performance on a downstream task for which labeled data are available. Prompt tuning optimizes the prompt $\mathbf{p} \in \mathbb{R}^{L \times D}$ in the text embedding space, with the number of tokens L and embedding size D , using training data with annotations $\mathcal{D}_{\text{train}} = \{(X_i, y_i)\}$ from the downstream task. The goal is to obtain text inputs $\{\mathbf{p}; \mathcal{Y}\} = \{\{\mathbf{p}; y_i\} \text{ for } y_i \in \mathcal{Y}\}$ that can provide the model with the most helpful context information about the task. For image classification with cross-entropy loss $\mathcal{L}$, the problem can be formulated as:

$$\mathbf{p}^* = \arg \min_{\mathbf{p}} \mathbb{E}_{(X, y) \sim \mathcal{D}_{\text{train}}} \mathcal{L}(\mathcal{F}_{\mathbf{p}}(X), y), \quad (6.1)$$

$$\text{where } \mathcal{F}_{\mathbf{p}}(X) = \text{sim}(\mathbf{E}_{\text{text}}(\{\mathbf{p}; \mathcal{Y}\}), \mathbf{E}_{\text{visual}}(X)). \quad (6.2)$$

Context-Dependent Visual Reasoning For the task of context-dependent visual reasoning, such as Bongard-HOI [19], a test sample contains two sets of support images and a query image for evaluation. The two sets of support images exemplify the presence and the absence of a human-object interaction (HOI) concept (e.g., “ride bike”). The model is then asked to infer whether the query image contains the underlying concept. Specifically, each concept in this task is a visual relationship $c = \langle s, a, o \rangle$, with s being the subject (s = “human” for HOI tasks), a denoting the action and o for the object. Each test sample X_{test} captures a concept by presenting $c = \langle s, a, o \rangle$ in one set of support images (positive examples), while having the other set (negative examples) to demonstrate $c' = \langle s, a', o \rangle$, where $a' \neq a$. Note that

neither o nor a is given explicitly, and it relies on the model’s reasoning ability to predict whether the query image contains the featured concept c of the test sample.

Existing methods [57, 58] approach the Bongard-HOI problem by training the model on a collection of similar tasks (using the Bongard-HOI training split) so that it can make similar inferences on test samples at test time. When applying CLIP to this task, we do not use additional training data because CLIP has learned abundant visual concepts and thus is a natural fit for such visual reasoning tasks.

6.3.2 TPT: Test-Time Prompt Tuning

Why Optimize Prompts? CLIP contains rich knowledge obtained from pretraining on a massive and diverse dataset. However, how to *more effectively* extract such knowledge remains an open question. A simple strategy is to directly fine-tune the model, either end-to-end or for a subset of layers, on a category of inputs. However, previous work has shown that such fine-tuning strategies result in domain-specific behaviors that lose the out-of-distribution generalization and robustness of foundation models [12, 59]. Prompts, on the other hand, work outside the pretrained model by modifying the context of the model input, thus do not distort pretrained features.

In this chapter, our goal is to leverage the existing knowledge of CLIP to boost its generalization in a zero-shot manner. Therefore, prompt tuning serves as an ideal handle to approach the goal. Furthermore, we regard test-time prompt tuning as a way to provide the model with the context tailored to the single test sample, which helps precisely retrieve the knowledge of CLIP.

At the inference stage, the only information available is the single test sample X_{test} without label information. TPT, therefore, manages to optimize the prompt $\mathbf{p}$ at test time based on the single test sample. In general, our objective can be formulated in the form of

$$\mathbf{p}^* = \arg \min_{\mathbf{p}} \mathcal{L}(\mathcal{F}, \mathbf{p}, X_{\text{test}}) \quad (6.3)$$

for some carefully constructed loss. Note that, unlike equation (6.1), our method does not require any labels or any data beyond the zero-shot test sample.

TPT for Image Classification Because labels are not available for test time tuning, we must select an unsupervised loss for prompt tuning. We design our TPT objective to promote the consistency of the model’s predictions across different augmented views of a given test image. Specifically, we generate N randomly augmented views

of the test image using a family of random augmentations $\mathcal{A}$ and minimize the entropy of the averaged prediction probability distribution:

$$\mathbf{p}^* = \arg \min_{\mathbf{p}} - \sum_{i=1}^K \tilde{p}_{\mathbf{p}}(y_i | X_{\text{test}}) \log \tilde{p}_{\mathbf{p}}(y_i | X_{\text{test}}), \quad (6.4)$$

$$\text{where } \tilde{p}_{\mathbf{p}}(y_i | X_{\text{test}}) = \frac{1}{N} \sum_{i=1}^N p_{\mathbf{p}}(y_i | \mathcal{A}_i(X_{\text{test}})). \quad (6.5)$$

Here, $p_{\mathbf{p}}(y | \mathcal{A}_i(X_{\text{test}}))$ is the vector of class probabilities produced by the model when provided with prompt $\mathbf{p}$ and the i -th augmented view of the test image.

In addition, to reduce the noise from random augmentations, we introduce *confidence selection* to filter out views that generate high-entropy (i.e., low-confidence) predictions. Such views of an image may lack important information needed to classify it correctly, e.g., a random crop may have removed important image content. We select confident samples with a prediction entropy below a threshold τ . We adapt τ for each test sample, by taking the entropy value at the ρ -percentile among the self-entropy of N augmented views ranked from low to high (i.e., confidence from high to low). With τ , the confidence selection can be written as a mask over the augmented samples $\mathbb{K}[\mathbf{H}(p_i) \leq \tau]$, with $\mathbf{H}$ measuring the self-entropy of the prediction on an augmented view. Using confidence selection with a cutoff percentile ρ on N augmented views, the averaged probability in Eq. (6.4) now becomes:

$$\tilde{p}_{\mathbf{p}}(y | X_{\text{test}}) = \frac{1}{\rho N} \sum_{i=1}^N \mathbb{K}[\mathbf{H}(p_i) \leq \tau] p_{\mathbf{p}}(y | \mathcal{A}_i(X_{\text{test}})), \quad (6.6)$$

TPT for Context-Dependent Visual Reasoning Different from image classification, where every image has one and only ground-truth label, the correctness of the prediction in Bongard-HOI depends on the context (i.e., example images), which is binary (containing the concept c or not). In the case of binary labels, a straightforward prompting strategy is to hand-craft “labels” for positive and negative examples, such as “True/False” or “Yes/No.” With TPT, on the other hand, we can directly learn an optimal label token $\mathbf{cls}$ on the example images in the test sample. More importantly, for visual reasoning, TPT can explicitly learn the context (i.e., visual concept) in the form of text prompts, and assists visual reasoning of contrastive VLMs with language context. Formally, given M support images in each test sample, the TPT objective for context-dependent reasoning can be written as:

$$\mathbf{p}^* = \arg \min_{\mathbf{p}, \mathbf{cls}} \frac{1}{M} \sum_{X \in \{X_{\mathcal{P}}, X_{\mathcal{N}}\}} \mathcal{L}(\mathcal{F}_{c, \mathbf{cls}}(X), y), \quad (6.7)$$

where we assign $y \in \{0, 1\}$ to negative and positive example images respectively for computing the cross-entropy loss $\mathcal{L}$. Unlike in image classification, we tune the binary label tokens $\mathbf{cls} = \{\mathbf{cls}^1, \mathbf{cls}^2\}, \mathbf{cls}^i \in \mathbb{R}^{1,D}$ and prompt $\mathbf{p} \in \mathbb{R}^{L,D}$ simultaneously. For each image, we assemble the text input to CLIP as $T = \{T_1, T_2 \mid T_i = \{\mathbf{p}, \mathbf{cls}^i\}\}$.

Note that the support set is an essential part of a Bongard-HOI sample, which provides the context for this context-dependent task. Therefore, our approach is still considered to work purely at test time, without training data or annotations (i.e., TPT has not been trained on a collection of similar tasks from the Bongard-HOI training split).

6.4 Experiments

In this section, we describe the tasks and benchmarks used for evaluating our method, along with the implementation details. Our main results cover three aspects of the model’s generalization: robustness to natural distribution shifts, cross-dataset generalization, and context-dependent visual reasoning. We also provide ablation experiments in Sect. 6.5, analyzing different network components for test-time tuning, and other design choices of our method.

6.4.1 Robustness to Natural Distribution Shifts

Datasets CLIP has been shown to be robust to distribution shifts that can occur naturally in real-world scenarios. We follow the setting in Radford et al. [1] and evaluate the model’s robustness to natural distribution shifts on 4 ImageNet Variants as follows, which have been considered as out-of-distribution (OOD) data for ImageNet [18] in previous work.

- **ImageNet-V2** [16] is a independent test set containing natural images, collected from different source, including 10,000 images of 1,000 ImageNet categories.
- **ImageNet-A** [15] is a challenging test set of “natural adversarial examples” misclassified by a standard ResNet-50 [60], consisting of 7,500 images of 200 of ImageNet categories.
- **ImageNet-R** [13] collects images of ImageNet categories but with artistic renditions. There are 30,000 images in total, including 200 ImageNet categories.
- **ImageNet-Sketch** [17] is a dataset of black and white sketches, collected independently from the original ImageNet validation set. The dataset includes 50,000 images in total, covering 1,000 ImageNet categories.

Baselines We compare TPT with existing few-shot prompt tuning methods that are designed for CLIP. CoOp [31] is a few-shot prompt tuning baseline that tunes a fixed dataset-specific prompt on each downstream dataset. CoCoOp [14] is the state-of-the-art prompt tuning method for CLIP. It produces input-dependent prompts with a network module, of which the input is the image feature. The network module of CoCoOp is also trained on downstream data in a dataset-specific way. Following their original configuration, we train both methods on ImageNet using 16-shot training data per category with 4 learnable prompt tokens and directly test the tuned prompt on OOD benchmarks. We also include two versions of the baseline zero-shot performance of CLIP, using a default prompt “a photo of a,” and the ensemble of 80 hand-crafted prompts from Radford et al. [1].

Implementation Details For TPT, we initialize the prompt as the default hand-crafted one “a photo of a” and optimize the corresponding 4 tokens in the text input embedding space based on a single test image. We augment a single test image 63 times using random resized crops and construct a batch of 64 images, including the original one. Among the 64 predictions, we select the top 10% ($\rho=0.1$) confident samples (lowest 10% in self-entropy) and compute the entropy of the averaged probability of the selected predictions (i.e., marginal entropy). We optimize the prompt to minimize the marginal entropy for 1 step, using the AdamW optimizer with a learning rate of 0.005.

Results In Table 6.1, the standalone TPT achieves higher accuracy than both prompt ensemble and existing few-shot prompt tuning methods, including CoCoOp. Furthermore, since TPT works at test time, it is complementary to existing baseline methods. We show that by applying TPT to prompts learned by CoOp or CoCoOp, we can further improve the accuracy of their in-domain ImageNet data, as well as generalization ability to OOD data. We also compare TPT to the ensembles of baseline models in Appendix 6.6, where we find that applying TPT to baseline methods can bring more substantial improvement than model ensembles. In addition, among the five datasets, few-shot prompt tuning methods bring the most accuracy gain on the ImageNet validation set and ImageNet-V2. However, on datasets with more significant distribution shifts, few-shot prompt tuning methods trained on ImageNet show no advantage over the ensemble of hand-crafted prompts.

6.4.2 Cross-Datasets Generalization

Pretrained contrastive VLMs like CLIP are ideal for “open-world” problems. For example, we can apply CLIP to classify arbitrary categories in a zero-shot manner in image classification. However, a prompt tuned on a specific downstream dataset can

Table 6.1 Robustness to Natural Distribution Shifts. CoOp and CoCoOp are tuned on ImageNet using 16-shot training data per category. Baseline CLIP, prompt ensemble, and TPT do not require training data

Method	ImageNet Top1 acc. ↑	ImageNet-A Top1 acc. ↑	ImageNet-V2. Top1 acc. ↑	ImageNet-R. Top1 acc. ↑	ImageNet-sketch Top1 acc. ↑	Average	OOD average
CLIP-RN50	58.16	21.83	51.41	56.15	33.37	44.18	40.69
Ensemble	59.81	23.24	52.91	60.72	35.48	46.43	43.09
CoOp	63.33	23.06	55.40	56.60	34.67	46.61	42.43
CoCoOp	62.81	23.32	55.72	57.74	34.48	46.81	42.82
TPT	60.74	26.67	54.70	59.11	35.09	47.26	43.89
TPT + CoOp	64.73	30.32	57.83	58.99	35.86	49.55	45.75
TPT + CoCoOp	62.93	27.40	56.60	59.88	35.43	48.45	44.83
CLIP-ViT-B/16	66.73	47.87	60.86	73.98	46.09	59.11	57.20
Ensemble	68.34	49.89	61.88	77.65	48.24	61.20	59.42
CoOp	71.51	49.71	64.20	75.21	47.99	61.72	59.28
CoCoOp	71.02	50.63	64.07	76.18	48.75	62.13	59.91
TPT	68.98	54.77	63.45	77.06	47.94	62.44	60.81
TPT + CoOp	73.61	57.95	66.83	77.27	49.29	64.99	62.83
TPT + CoCoOp	71.07	58.47	64.85	78.65	48.47	64.30	62.61

be less generalizable to categories outside its training set. In this section, we evaluate the cross-dataset generalization of existing few-shot prompt tuning methods (same as in Sect. 6.4.1) and compare them with TPT, which is not dataset-specific.

Setup We conduct a cross-dataset evaluation on the task of image classification. We consider 10 datasets, covering fine-grained classifications including species of plants or animals (Flower102 [61], OxfordPets [62]), scenes (SUN397 [63]), textures (DTD [64]), food (Food101 [65]), transportation (StanfordCars [66], Aircraft [67]), human actions (UCF101 [68]), satellite images (EuroSAT [69]), and general objects (Caltech101 [70]). We consider two different settings of cross-dataset generalization. In the first setting, we consider ImageNet with 1000 categories as a comprehensive source dataset and use other fine-grained datasets as target datasets for evaluation. We implement CoOp and CoCoOp using the same setting as in Sect. 6.4.1 and evaluate their generalization performance to the 10 datasets. In the second setting, we consider a more challenging scenario, where the source data for few-shot prompt tuning also comes from the specialized fine-grained datasets, and there is no overlapping in categories between a source-target pair.

Implementation Details We implement CoOp and CoCoOp on each source dataset following their original configurations. For TPT, we initialize the prompt as “a photo of a” for every datasets. We adopt the same hyper-parameter setting as in Sect. 6.4.1. We use AugMix [71] as a stronger data augmentation for this task.

Results In Table 6.2, we compare TPT with few-shot prompt tuning methods on generalization from ImageNet to fine-grained datasets. Note that TPT works in a zero-shot manner; thus, it is not trained on ImageNet. Nonetheless, we find TPT to achieve on-par generalization as ImageNet trained CoCoOp. In Fig. 6.2, we present the results of the more challenging setting of cross-dataset generalization, where there is no overlap between the source and target dataset. For better visualization, we plot the relative accuracy improvement $acc' = (acc - acc_{base}) / acc_{base}$, normalized by the zero-shot baseline accuracy acc_{base} of a CLIP-RN50. For example, baseline CLIP with a hand-crafted prompt achieves 61.75% accuracy on Flower102, while CoOp trained on DTD only has 33.41% on Flower102. In this case, we calculate acc' as $(33.41 - 61.75) / 61.75 = -0.46$. From Fig. 6.2, we can see that the averaged accuracy improvement (in the last column of each matrix) of few-shot prompt tuning methods is always negative, meaning that they do worse than the zero-shot baseline. TPT, on the other hand, shows consistent improvement in each of the 10 datasets.

Table 6.2 Cross-dataset generalization from ImageNet to fine-grained classification datasets. CoOp and CoCoOp are tuned on ImageNet using 16-shot training data per category. Baseline CLIP, prompt ensemble, and TPT do not require training data or annotations. We report the top-1 classification accuracy on each dataset

Method	Flower102	DTD	Pets	Cars	UCF101	Caltech101	Food101	SUN397	Aircraft	EuroSAT	Average
CLIP-RN50	61.75	40.37	83.57	55.70	58.84	85.88	73.97	58.80	15.66	23.69	55.82
Ensemble	62.77	40.37	82.97	55.89	59.48	87.26	74.82	60.85	16.11	25.79	56.63
CoOp	61.55	37.29	87.00	55.32	59.05	86.53	75.59	58.15	15.12	26.20	56.18
CoCoOp	65.57	38.53	88.39	56.22	57.10	87.38	76.20	59.61	14.61	28.73	57.23
TPT	62.69	40.84	84.49	58.46	60.82	87.02	74.88	61.46	17.58	28.33	57.66
CLIP-ViT-B/16	67.44	44.27	88.25	65.48	65.13	93.35	83.65	62.59	23.67	42.01	63.58
Ensemble	66.99	45.04	86.92	66.11	65.16	93.55	82.86	65.63	23.22	50.42	64.59
CoOp	68.71	41.92	89.14	64.51	66.55	93.70	85.30	64.15	18.47	46.39	63.88
CoCoOp	70.85	45.45	90.46	64.90	68.44	93.79	83.97	66.89	22.29	39.23	64.63
TPT	68.98	47.75	87.79	66.87	68.04	94.16	84.67	65.50	24.78	42.44	65.10

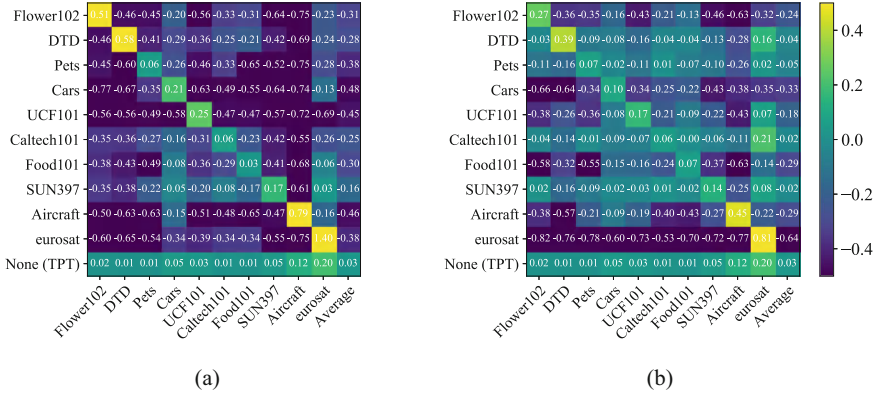


Fig. 6.2 Cross-dataset improvement normalized by the zero-shot baseline performance. In each matrix A , $A_{i,j}$ is the *normalized relative improvement* on the j_{th} dataset of using the prompt tuned on the i -th dataset. The value $A_{i,j}$ stands for how well a method trained on a source dataset i performs on a target dataset j , in comparison with a zero-shot CLIP baseline (using a hand-crafted prompt). Thus, the higher, the better. The last row is the performance of TPT, which is not tuned on any source dataset. The last column summarizes the average improvement over 10 datasets, measuring the overall generalization ability across the 10 datasets. (a) CoOp with CLIP-RN50. (b) CoCoOp with CLIP-RN50

6.4.3 Context-Dependent Visual Reasoning on Bongard-HOI

Baselines We include three previous methods for comparison: (1) The CNN-baseline [57] is a simple classifier trained on Bongard-HOI training data, for which the model is trained to map a training sample as a whole (including the support and query images) to a binary output, indicating whether the query image contains the corresponding concept; (2) The Meta-baseline [58] regards each sample in Bongard-HOI as a few-shot task, and the model is trained on the Bongard-HOI training data with a meta-objective that aims to quickly adapt the model to new tasks; (3) HOITrans [72] is the previous best method on Bongard-HOI. It is a transformer-based HOI detection model that achieves state-of-the-art accuracy on various HOI detection benchmarks. It solves Bongard-HOI by comparing the detected HOIs of the query images to those of the support images.

Implementation Details For each Bongard-HOI test sample, TPT tunes the prompt prefix and class tokens simultaneously from scratch. All learnable tokens are initialized in the text embedding space from a Gaussian distribution with $\sigma = 0.02$. We optimize the prompt on all support images of a test sample for 64 steps, using the AdamW optimizer with a learning rate of 0.005, and then infer the query image with the updated prompt. For other baselines, we directly report the results from Jiang et al. [19], and we refer interested readers to this paper for more details. Note that the HOITrans model is trained on all possible HOI concepts, including the ones in the testing splits.

Table 6.3 Evaluation on the Bongard-HOI benchmark. CNN and Meta baselines are implemented based on a ResNet-50 (RN50). (* denotes that the method uses ground truth bounding boxes to assist the inference)

Method	Test splits				
	Seen act., seen obj.,	Unseen act., seen obj.,	Seen act., unseen obj.,	Unseen act., unseen obj.,	Average
CNN-baseline	50.03	49.89	49.77	50.01	49.92
Meta-baseline*	58.82	58.75	58.56	57.04	58.30
HOITrans	59.50	64.38	63.10	62.87	62.46
TPT (w/ CLIP-RN50)	66.39	68.50	65.98	65.48	66.59

Results In Table 6.3, we follow the setup in Jiang et al. [19], and compare TPT with previous methods on 4 test splits of Bongard-HOI respectively. In Bongard-HOI, test images are split into four sets by their overlap in the HOI concept with the training data: whether the action a or the object o has appeared in the training data. Note that our CLIP-based TPT is not trained on the training split of Bongard-HOI, and thus, the definition of the four splits is not strictly applicable to TPT.

6.5 Ablation Study

In this section, we analyze our design choices and provide ablation study on the effects of different components of TPT. For simplicity, if not otherwise specified, analyses in this section are conducted on the natural distribution shifts benchmarks. We first compare test-time optimization on different parameter groups of CLIP, showing that prompt tuning achieves the most accuracy gain. Next, we show the improvement brought by confidence selection and analyze how the confidence threshold affects performance. Additional ablation studies can be found in Appendix 6.6.

Test-Time Optimization on Different Parameter Groups of CLIP Existing test-time optimization methods have worked on different parameter groups of a model. Although there is a strong intuition for tuning prompts on CLIP, it is unclear whether it is the most effective choice. In Fig. 6.3a, we evaluate the performance of test-time optimization on different parameter groups of CLIP. We consider four different parameter groups: (1) the entire model, (2) the text encoder, (3) the visual encoder, and (4) the text prompt. For a fair comparison, we adopt the same setup as MEMO [52] by using AugMix [71] as the data augmentation. Confidence selection is not used in this ablation study. For each design choice, we run a grid search for hyper-parameter tuning (on the learning rate and the number of optimization steps) and report the best result.

The result suggests that text prompt is the most effective parameter group. In addition, we find optimizing the visual encoder to have the worst result. This observation is in alignment with previous work that suggests fine-tuning the image encoder can distort pretrained features [59, 73].

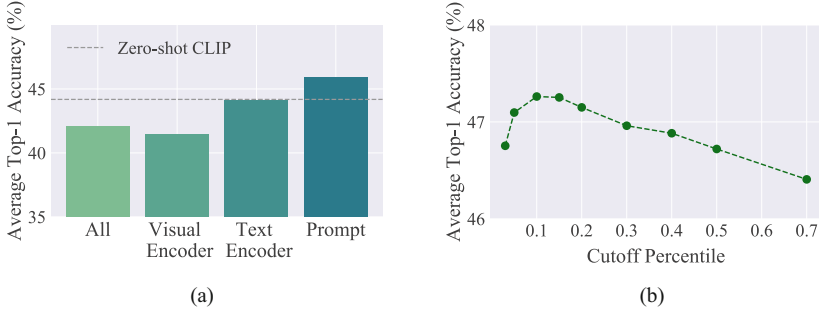


Fig. 6.3 Ablating the effects of different components of TPT. We evaluate the top-1 accuracy on the distribution shifts benchmarks in Sect. 6.4.1. Methods are implemented based on a CLIP-RN50. (a) Test-time optimization on different modules. (b) Different cutoff percentile in confidence selection

The Effect of Confidence Selection We present confidence selection as a major component of our method, which filters out “noisy” augmented views that provide little information. In Table 6.4, we provide the performance of TPT without confidence selection, in comparison with the full method. Confidence selection brings nontrivial performance improvement to our baseline TPT. We further show the effect of confidence threshold ρ in Fig. 6.3b. The result suggests that using the top-10% confident sample leads to the highest average accuracy. In addition, we find that the effect of confidence selection is generalizable to other entropy-based test-time optimization methods. More details about this analysis are included in Appendix 6.6

6.6 Conclusion

In this chapter, we focus on how to fully unleash the potential of contrastive vision-language foundation models as better zero-shot learners. We discuss on Test-time Prompt Tuning (TPT), a prompt tuning method that can learn adaptive prompts on the fly with a single test sample. We demonstrate the effectiveness of our method on the robustness to natural distribution shifts and cross-dataset generalization. Without the need for any training data or annotations, TPT improves the zero-shot generalization of CLIP.

The idea of TPT (or more generally, test-time optimization) can be applied to other foundation models such as generative VLMs [5, 74] and foundation models of other modalities (e.g., large language models (LLMs) [20, 21]). The key problem to study in this direction is to design a test-time objective that fits the model design and downstream tasks. It is also important to investigate how to accelerate the inference speed while applying test-time optimization.

Table 6.4 The effect of confidence selection. The last row is the performance of our full method

Method	ImageNet	ImageNet-A	ImageNet-V2.	ImageNet-R.	ImageNet-sketch	Average	OOD average
	Top1 acc. ↑	Top1 acc. ↑	Top1 acc. ↑	Top1 acc. ↑	Top1 acc. ↑		
CLIP-RN50	58.16	21.83	51.41	56.15	33.37	44.18	40.69
Baseline TPT	60.31	23.65	53.66	57.48	34.31	45.88	42.28
+ confidence selection	60.74 (+0.43)	26.67 (+3.02)	54.70 (+1.04)	59.11 (+1.63)	35.09 (+0.78)	47.26 (+1.38)	43.89 (+1.61)

Further Reading

Interested readers may refer to the following material:

- Feng et al. [75]: Improving the data augmentation for TPT with diffusion models.
- Goyal et al. [76]: Finding a good objective (i.e., loss function) for test-time optimization.
- Snell et al. [77]: The scaling of test-time computation in LLMs.
- Akyürek et al. [78]: Using test-time optimization to improve the reasoning ability of language models.

Acknowledgments Shu and Goldstein were supported by the ONR MURI program and DARPA GARD.

Appendix

A Diagram of Applying TPT to Context-Dependent Visual Reasoning (Bongard-HOI)

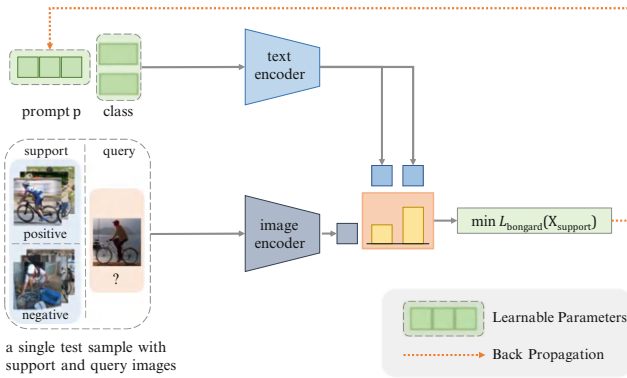


Fig. 6.4 Test-time Prompt Tuning (TPT) for context-dependent visual reasoning on Bongard-HOI benchmark. A test sample in Bonagrd-HOI consists of several support images that exemplify a visual concept, and the model needs to predict whether the query image contains the concept. TPT tunes the prompt and class tokens simultaneously on the support images using the cross-entropy loss

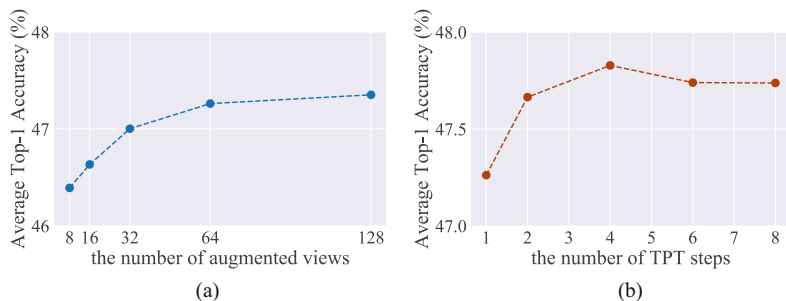


Fig. 6.5 Analysis on the trade-off between efficiency and accuracy. We evaluate the top-1 accuracy on the distribution shifts benchmarks in Sect. 6.4.1. Results are based on a CLIP-RN50. (a) Different number of augmented views. (b) Different number of optimization steps

More Experiments

The Trade-Off Between Inference Efficiency and Accuracy We analyze two factors that affect TPT’s efficiency: (1) the number of augmented views N that increases the actual number of test samples; (2) the number of optimization steps that increases the runtime and memory usage mainly induced by backpropagation. Figure 6.5 shows the relationship between the two factors and the average accuracy of TPT on natural distribution shifts (Fig. 6.4).

In Fig. 6.5a, the accuracy increases as the number of augmented views grows until reaching a plateau at around $N = 64$. Even when $N = 8$, TPT still brings about over 2% average accuracy gain to the zero-shot CLIP, suggesting that TPT can be adapted for more efficient applications. In Fig. 6.5b, we find that increasing the number of optimization steps from 1 to 2 can slightly increase the accuracy (by 0.4%), while there is no significant performance gain from taking more than 2 steps. Considering that the performance gain comes at the expense of linearly increasing the inference time, we use 1-step TPT as our default setting, which is already capable of boosting the average accuracy of zero-shot CLIP by more than 3%.

Analysis on Error Bars We run TPT multiple times using 3 different random seeds and report the average accuracy with standard deviation. The randomness of TPT mainly comes from random data augmentation and one-step optimization. In addition, we report error bars for few-shot prompt tuning methods (CoOp and CoCoOp). The randomness of these methods mainly comes from the random few-shot sampling of the training data. We provide error bar analysis for the results in Tables 6.1 and 6.2 in the main paper.

From Table 6.5, our conclusion remains the same as in Table 6.1, that TPT achieves higher accuracy than few-shot prompt tuning methods on OOD benchmarks. In addition, we find that on average TPT has a smaller standard deviation than other methods.

In Table 6.6, we provide the same error bar analysis for the cross-dataset generalization experiment, corresponding to Table 6.2 in Sect. 6.4. Similar to the

Table 6.5 Robustness to natural distribution shifts. We report the accuracy with an error bar (standard deviation) obtained from three runs with different random seeds

Method	ImageNet Top1 acc. $\uparrow$	ImageNet-A Top1 acc. $\uparrow$	ImageNet-V2. Top1 acc. $\uparrow$	ImageNet-R. Top1 acc. $\uparrow$	ImageNet-sketch Top1 acc. $\uparrow$	Average	OOD average
CLIP-RN50	58.16	21.83	51.41	56.15	33.37	44.18	40.69
CoOp	63.27 (± 0.07)	23.23 (± 0.19)	55.50 (± 0.09)	57.08 (± 0.42)	34.68 (± 0.03)	46.75 (± 0.12)	42.62 (± 0.17)
CoCoOp	62.86 (± 0.11)	23.38 (± 0.50)	55.59 (± 0.14)	57.55 (± 0.23)	34.74 (± 0.29)	46.82 (± 0.21)	42.82 (± 0.23)
TPT	60.77 (± 0.03)	26.60 (± 0.13)	54.70 (± 0.11)	59.08 (± 0.03)	35.17 (± 0.08)	47.27 (± 0.004)	43.89 (± 0.003)
CLIP-ViT-B/16	66.73	47.87	60.86	73.98	46.09	59.11	57.2
CoOp	71.71 (± 0.19)	49.99 (± 0.29)	64.49 (± 0.39)	75.51 (± 0.26)	48.10 (± 0.14)	61.96 (± 0.25)	59.52 (± 0.26)
CoCoOp	70.70 (± 0.32)	50.76 (± 0.13)	63.93 (± 0.19)	76.09 (± 0.29)	48.60 (± 0.38)	62.02 (± 0.20)	59.85 (± 0.19)
TPT	68.96 (± 0.03)	54.47 (± 0.26)	63.46 (± 0.07)	77.10 (± 0.04)	47.93 (± 0.03)	62.38 (± 0.05)	60.74 (± 0.06)

Table 6.6 Cross-dataset generalization from ImageNet to fine-grained classification datasets. CoOp and CoCoOp are tuned on ImageNet using 16-shot training data per category. Baseline CLIP and TPT do not require training data or annotations. We report the averaged top-1 accuracy on each dataset, with standard deviations obtained from three runs using different random seeds

Method	Flower102	DTD	Pets	Cars	UCF101	Caltech101	Food101	SUN397	Aircraft	EuroSAT	Average
CLIP-RN50	61.75	40.37	83.57	55.70	58.84	85.88	73.97	58.80	15.66	23.69	55.82
CoOp	61.62 (± 0.2)	37.77 (± 0.9)	87.24 (± 0.2)	55.72 (± 0.8)	59.89 (± 0.8)	87.23 (± 0.6)	75.86 (± 0.2)	59.28 (± 0.9)	15.20 (± 0.4)	25.43 (± 4)	56.52 (± 0.7)
CoCoOp	65.11 (± 1)	39.14 (± 0.7)	87.83 (± 0.6)	56.40 (± 0.3)	58.57 (± 1)	86.95 (± 0.5)	76.18 (± 0.5)	60.62 (± 0.9)	15.13 (± 0.5)	28.79 (± 0.9)	57.47 (± 0.2)
TPT	62.80 (± 0.3)	41.43 (± 0.5)	84.42 (± 0.1)	58.53 (± 0.1)	60.64 (± 0.3)	87.23 (± 0.2)	75.02 (± 0.1)	61.46 (± 0.0)	17.60 (± 0.4)	28.46 (± 0.1)	57.76 (± 0.1)
CLIP-ViT-B/16	67.44	44.27	88.25	65.48	65.13	93.35	83.65	62.59	23.67	42.01	63.58
CoOp	68.25 (± 0.5)	42.34 (± 2)	89.38 (± 0.2)	63.35 (± 1)	67.17 (± 1)	92.82 (± 0.5)	83.74 (± 0.4)	64.51 (± 0.6)	19.99 (± 2)	40.22 (± 4)	63.18 (± 0.7)
CoCoOp	71.59 (± 0.6)	45.48 (± 0.2)	90.20 (± 0.2)	65.17 (± 0.2)	68.77 (± 0.8)	94.15 (± 0.3)	84.83 (± 1)	67.07 (± 0.3)	22.95 (± 0.7)	42.13 (± 3)	65.23 (± 0.6)
TPT	68.79 (± 0.1)	46.79 (± 0.1)	87.09 (± 0.1)	66.38 (± 0.2)	67.86 (± 0.1)	94.13 (± 0.1)	84.67 (± 0.1)	65.41 (± 0.1)	23.44 (± 0.3)	42.78 (± 0.3)	64.73 (± 0.1)

observation earlier, our conclusion for cross-dataset evaluation remains the same. On average, we find TPT to have a smaller standard deviation than few-shot prompt tuning methods.

Comparison with Model Ensembles We include more comparisons with conventional model ensembles of the baselines in Table 6.1. Note that there exists a fundamental difference between our “ensemble” (e.g., TPT + CoOp) and conventional model ensemble (e.g., naive “hand-crafted” ensemble, CoCoOp with different seeds, etc.). As our method (TPT) works solely at test time and uses a predefined prompt (which may come from a hand-crafted prompt, CoOp, or CoCoOp) as the initialization, it is complementary to CoOp, CoCoOp, and hand-crafted prompts.

Specifically, taking “TPT + CoCoOp” as an example, it is done in the following steps: (1). Using the prompt output by CoCoOp as the prompt initialization; (2). Running TPT to tune this prompt at test time to get the final result. However, conventional model ensembles aggregate the predictions obtained from different prompts.

As shown in Table 6.7, ensembles of existing baselines (e.g., CoOp + CoCoOp) fail to bring improvement as substantial as their combination with TPT (e.g. CoOp + TPT or CoCoOp + TPT).

Other Baselines Using Data Augmentation To ablate the contribution of data augmentation in our method, we include two additional baselines that are based on data augmentations without any optimization (and without TPT). The *averaged prediction* baseline is to run inference on the augmented images and then average the probability. The *majority vote* baseline runs the majority vote in the predictions of augmented images and takes the results as the final prediction.

In Table 6.8, we find that both methods fail to achieve comparable improvement as TPT. It suggests that it is nontrivial to design an algorithm for using augmented images.

Apply Confidence Selection to Other Methods

In Sect. 6.5, we show that confidence selection is an important part of TPT that improves the baseline entropy minimization. In this section, we further show that confidence selection can also benefit other entropy-based test-time optimization methods, which work on different model architectures and optimize different parameter groups. MEMO [52] minimizes the marginal entropy of the model’s predictions across augmented views at test time, by adapting all parameters of a network model.

In Table 6.9, we implement MEMO based on a standard ResNet-50 from PyTorch, following their original hyper-parameter configurations. To apply confidence selection to a method, we try different cutoff percentile ρ that controls the threshold for confidence selection (smaller ρ means a higher confidence threshold). We find MEMO can also benefit from confidence selection and the improvement increases as the threshold increases (i.e. ρ becomes smaller).

Table 6.7 Robustness to Natural Distribution Shifts. CoOp and CoCoOp are tuned on ImageNet using 16-shot training data per category. Baseline CLIP, prompt ensemble, and TPT do not require training data

Method	ImageNet Top1 acc. ↑	ImageNet-A Top1 acc. ↑	ImageNet-V2. Top1 acc. ↑	ImageNet-R. Top1 acc. ↑	ImageNet-Sketch Top1 acc. ↑	Average	OOD average
CLIP-RN50	58.16	21.83	51.41	56.15	33.37	44.18	40.69
Hand-crafted ensemble	59.81	23.24	52.91	60.72	35.48	46.43	43.09
CoOp	63.33	23.06	55.40	56.60	34.67	46.61	42.43
CoOp (ensemble 3 seeds)	61.66	22.96	54.14	57.89	34.94	46.32	42.48
CoOp + hand-crafted ensemble	63.60	23.23	55.63	57.07	34.84	46.87	42.69
CoCoOp	62.81	23.32	55.72	57.74	34.48	46.81	42.82
CoCoOp (ensemble 3 seeds)	63.34	24.27	56.12	58.24	35.46	47.49	43.52
CoCoOp + hand-crafted ensemble	63.03	24.16	55.73	57.88	35.22	47.20	43.25
CoCoOp + CoOp	63.86	23.69	56.45	57.7	35.5	47.44	43.34
TPT (ours)	60.74	26.67	54.7	59.11	35.09	47.26	43.89
TPT + CoOp	64.73	30.32	57.83	58.99	35.86	49.55	45.75
TPT + CoCoOp	62.93	27.4	56.6	59.88	35.43	48.45	44.83

Table 6.8 Out-of-distribution evaluation of data augmentation baselines. We include two additional baselines that are based on data augmentations without any optimization (and without TPT). The limited performance of these baselines suggests that it is nontrivial to design an algorithm for using augmented images

Method	ImageNet Top1 acc. ↑	ImageNet-A Top1 acc. ↑	ImageNet-V2. Top1 acc. ↑	ImageNet-R. Top1 acc. ↑	ImageNet-sketch Top1 acc. ↑	Average	OOD average
CLIP-RN50	58.16	21.83	51.41	56.15	33.37	44.18	40.69
averaged prediction	57.94	22.71	52.02	51.25	29.51	42.69	38.87
majority vote	59.53	25.67	53.56	54.77	32.27	45.16	41.57
TPT (Ours)	60.74	26.67	54.70	59.11	35.09	47.26	43.89

Table 6.9 Apply confidence selection to other test-time methods. All methods are based on a standard ResNet-50. ρ denotes applying confidence selection to a method with a cutoff percentile ρ

Method	ImageNet Top1 acc. $\uparrow$	ImageNet-A Top1 acc. $\uparrow$	ImageNet-V2. Top1 acc. $\uparrow$	ImageNet-R. Top1 acc. $\uparrow$	ImageNet-sketch Top1 acc. $\uparrow$	Average	OOD average
ResNet-50	76.13	0.00	63.20	36.17	24.09	39.92	30.87
MEMO	77.23	0.75	65.03	41.34	27.72	42.41	33.71
MEMO ($\rho = 0.7$)	77.56	0.92	65.51	41.93	28.20	42.82	34.14
MEMO ($\rho = 0.5$)	77.72	1.15	65.77	42.29	28.55	43.10	34.44
MEMO ($\rho = 0.3$)	77.57	1.43	65.85	42.64	28.33	43.16	34.56
MEMO ($\rho = 0.1$)	77.38	2.59	65.37	42.90	28.04	43.26	34.72

Table 6.10 An overview of different prompting strategies for CLIP. “learnable” means the prompt is optimized based on certain objective functions. “no training data” means that no additional data are needed for tuning the prompt. “input-dependent” means the prompt is adaptive to each input instance

Prompting method	Learnable	No training data	Input-adaptive
Hand-crafted [1]	–	✓	–
CoOp [31]	✓	–	–
CoCoOp [14]	✓	–	✓
TPT (ours)	✓	✓	✓

Table 6.11 Comparison of training and testing budgets of prompting strategies for CLIP

	Hand-crafted [1]	CoOp [31]	CoCoOp [14]	TPT (Ours)
Inference speed (seconds/iter)	0.10	0.10	0.11	0.25
Number of training samples	0	16K	16K	0
Number of training iterations	0	12.5K	800K	0
Number of additional parameters	0	0	65.8K	0

An Overview of Different Prompting Strategies for CLIP

In addition to our discussion on related work in Sect. 6.2, we summarize the differences between existing prompting strategies for CLIP in Table 6.10. We focus on three preferred properties of a prompting strategy and use them to categorize the methods.

Comparison of Training and Inference Budgets We include a comparison of training and inference budgets for prompting strategies of CLIP in Table 6.11.

The major computation overhead of TPT comes from the 1-step optimization, which involves back propagations through the text encoder of CLIP. Another overhead comes from the data augmentation, which can be parallelized with little memory increase, as the CLIP’s image branch does not require backpropagation. However, note that TPT works solely at test time, and therefore, it does not have any training budget. In addition, as shown by our empirical results, prompt tuning without training data can generalize better to unseen distributions.

References

1. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021)
2. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)

3. Wang, J., Chen, D., Wu, Z., Luo, C., Zhou, L., Zhao, Y., Xie, Y., Liu, C., Jiang, Y.-G., Yuan, L.: Omnivl: One foundation model for image-language and video-language tasks. arXiv preprint arXiv:2209.07526 (2022)
4. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: MDETR - modulated detection for end-to-end multi-modal understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
5. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., Chang, K.-W., Gao, J.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
6. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
7. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with CLIP latents. CoRR (2022)
8. Cai, Z., Kwon, G., Ravichandran, A., Bas, E., Tu, Z., Bhotika, R., Soatto, S.: X-DETR: A versatile architecture for instance-wise vision-language tasks. CoRR abs/2204.05626 (2022)
9. Liu, T., Wu, Z., Xiong, W., Chen, J., Jiang, Y.-G.: Unified multimodal pre-training and prompt-based tuning for vision-language understanding and generation. CoRR abs/2112.05587 (2021)
10. Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., Bai, X.: A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. CoRR abs/2112.14757 (2021)
11. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (2021)
12. Wortsman, M., Ilharco, G., Li, M., Kim, J.W., Hajishirzi, H., Farhadi, A., Namkoong, H., Schmidt, L.: Robust fine-tuning of zero-shot models. CoRR abs/2109.01903 (2021)
13. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
14. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
15. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15262–15271 (2021)
16. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: Chaudhuri, K., Salakhutdinov, R. (eds.) International Conference on Machine Learning (2019)
17. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. In: Advances in Neural Information Processing Systems (2019)
18. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition CVPR09 (2009)
19. Jiang, H., Ma, X., Nie, W., Yu, Z., Zhu, Y., Anandkumar, A.: Bongard-hoi: Benchmarking few-shot visual reasoning for human-object interactions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
20. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) . Association for Computational Linguistics (2019)
21. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E.,

- Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: *Advances in Neural Information Processing Systems* (2020)
22. Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y.-H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: *International Conference on Machine Learning* (2021)
 23. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: *International Conference on Machine Learning* (2020)
 24. Zaken, E.B., Ravfogel, S., Goldberg, Y.: Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In: *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022* (2022)
 25. Zhang, R., Fang, R., Zhang, W., Gao, P., Li, K., Dai, J., Qiao, Y., Li, H.: Tip-adapter: Training-free clip-adapter for better vision-language modeling. *CoRR abs/2111.03930* (2021)
 26. Zhang, R., Qiu, L., Zhang, W., Zeng, Z.: VT-CLIP: Enhancing vision-language models with visual-guided texts. *CoRR abs/2112.02399* (2021)
 27. Wang, Z., Codella, N., Chen, Y.-C., Zhou, L., Yang, J., Dai, X., Xiao, B., You, H., Chang, S.-F., Yuan, L.: CLIP-TD: CLIP targeted distillation for vision-language tasks. *CoRR abs/2201.05729* (2022)
 28. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: *ACL/IJCNLP. Association for Computational Linguistics* (2021)
 29. Wu, Z., Wang, S., Gu, J., Hou, R., Dong, Y., Vinod Vydiswaran, V.G., Ma, H.: IDPG: An instance-dependent prompt generation method. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2022)
 30. Yang, Z., Liu, Y.: On robust prefix-tuning for text classification. In: *ICLR* (2022)
 31. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *Int. J. Comput. Vis.* **130**(9), 2337–2348 (2022)
 32. Du, Y., Wei, F., Zhang, Z., Shi, M., Gao, Y., Li, G.: Learning to prompt for open-vocabulary object detection with vision-language model. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022).
 33. Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J.G., Pfister, T.: Learning to prompt for continual learning. *CoRR* (2021)
 34. Huang, T., Chu, J., Wei, F.: Unsupervised prompt learning for vision-language models. *CoRR abs/2204.03649* (2022)
 35. Zhou, C., He, J., Ma, X., Berg-Kirkpatrick, T., Neubig, G.: Prompt consistency for zero-shot task generalization. *CoRR abs/2205.00049* (2022)
 36. Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S.M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., Liang, P.: WILDS: A benchmark of in-the-wild distribution shifts. In: *International Conference on Machine Learning* (2021)
 37. Dhamija, A.R., Günther, M., Boulton, T.E.: Reducing network agnostophobia. In: *Advances in Neural Information Processing Systems* (2018)
 38. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Precup, D., Teh, Y.W. (eds.) *International Conference on Machine Learning* (2017)
 39. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: *ICLR* (2019)
 40. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: *International Conference on Machine Learning* (2019)
 41. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: *International Workshop on Artificial Intelligence and Statistics* (2005)

42. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: *Advances in Neural Information Processing Systems* (2016)
43. Shocher, A., Cohen, N., Irani, M.: “zero-shot” super-resolution using deep internal learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018)
44. Bau, D., Strobel, H., Peebles, W.S., Wulff, J., Zhou, B., Zhu, J.-Y., Torralba, A.: Semantic photo manipulation with a generative image prior. *ACM Trans. Graph.* **38**, 1–11 (2019)
45. Kundu, J.N., Venkat, N., Rahul, M.V., Babu, R.V.: Universal source-free domain adaptation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020)
46. Huang, Y., Gornet, J., Dai, S., Yu, Z., Nguyen, T.M., Tsao, D.Y., Anandkumar, A.: Neural networks with recurrent generative feedback. In: *Advances in Neural Information Processing Systems* (2020)
47. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A.A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: *International Conference on Machine Learning* (2020)
48. Liu, Y., Kothari, P., van Delft, B., Bellot-Gurlet, B., Mordan, T., Alahi, A.: Ttt++: When does self-supervised test-time training fail or thrive? In: *Advances in Neural Information Processing Systems* (2021)
49. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: *Advances in Neural Information Processing Systems* (2004)
50. Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K.: Semi-supervised domain adaptation via minimax entropy. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE (2019)
51. Wang, D., Shelhamer, E., Liu, S., Olshausen, B.A., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: *ICLR* (2021)
52. Zhang, M., Levine, S., Finn, C.: MEMO: Test time robustness via adaptation and augmentation. *CoRR* (2021)
53. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. In: *ICLR Workshop* (2017)
54. Shu, M., Wu, Z., Goldblum, M., Goldstein, T.: Encoding robustness to image style via adversarial feature perturbations. In: *Advances in Neural Information Processing Systems* (2021)
55. Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., Bethge, M.: Improving robustness against common corruptions by covariate shift adaptation. In: *Advances in Neural Information Processing Systems* (2020)
56. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: *International Conference on Machine Learning* (2020)
57. Nie, W., Yu, Z., Mao, L., Patel, A.B., Zhu, Y., Anandkumar, A.: Bongard-logo: A new benchmark for human-level concept learning and reasoning. In: *Advances in Neural Information Processing Systems* (2020)
58. Chen, Y., Wang, X., Liu, Z., Xu, H., Darrell, T.: A new meta-baseline for few-shot learning. *CoRR abs/2003.04390* (2020)
59. Kumar, A., Raghunathan, A., Jones, R., Ma, T., Liang, P.: Fine-tuning can distort pretrained features and underperform out-of-distribution. *CoRR abs/2202.10054* (2022)
60. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
61. Nilsback, M.-E., Zisserman, A.: Automated flower classification over a large number of classes. In: *Indian Conference on Computer Vision, Graphics and Image Processing* (2008)
62. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012)

63. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010)
64. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
65. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, September 6–12, 2014, Proceedings, Part VI 13 (2014)
66. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2013)
67. Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Technical Report (2013)
68. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR abs/1212.0402 (2012)
69. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: a novel dataset and deep learning benchmark for land use and land cover classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. **12**, 2217–2226 (2019)
70. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: CVPR Workshops (2004)
71. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. In: International Conference on Learning Representations (2020)
72. Zou, C., Wang, B., Hu, Y., Liu, J., Wu, Q., Zhao, Y., Li, B., Zhang, C., Zhang, C., Wei, Y., Sun, J.: End-to-end human object interaction detection with HOI transformer. In: Computer Vision and Pattern Recognition (2021)
73. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., Beyer, L.: Lit: Zero-shot transfer with locked-image text tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
74. Li, J., Li, D., Xiong, C., Hoi, S.C.H.: BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. CoRR abs/2201.12086 (2022)
75. Feng, C.-M., Yu, K., Liu, Y., Khan, S.A., Zuo, W.: Diverse data augmentation with diffusions for effective test-time prompt tuning. In: IEEE International Conference on Computer Vision (2023)
76. Goyal, S., Sun, M., Raghunathan, A., Kolter, J.Z.: Test time adaptation via conjugate pseudo-labels. In: Advances in Neural Information Processing Systems (2022)
77. Snell, C., Lee, J., Xu, K., Kumar, A.: Scaling llm test-time compute optimally can be more effective than scaling model parameters. arXiv preprint arXiv: 2408.03314 (2024)
78. Akyürek, E., Damani, M., Qiu, L., Guo, H., Kim, Y., Andreas, J.: The surprising effectiveness of test-time training for abstract reasoning. arXiv preprint arXiv: 2411.07279 (2024)

Chapter 7

Learning Efficient Feature Adapters for Vision-Language Models



Renrui Zhang and Peng Gao

Abstract Contrastive Language-Image Pretraining (CLIP) has demonstrated superior generalization capabilities in image classification over general visual domains. However, for some specific downstream tasks, CLIP still suffers from unsatisfactory results due to semantic gaps. To this end, CoOp has been proposed to adopt learnable prompt tokens for few-shot learning. In this chapter, we showcase an alternative to utilize feature adapters to enhance the few-shot image classification performance of CLIP. We first introduce CLIP-Adapter, an adapter architecture equipped after CLIP’s vision encoder for injecting downstream knowledge. On top of that, we further present two upgraded variants, Tip-Adapter and Tip-Adapter-F, a training-free adaption method, and its fine-tuning variant. They incorporate the few-shot semantics with the pretrained CLIP with a constructed cache model, which further achieve both effectiveness and efficiency for leading image classification performance.

Keywords Vision-language model · Fine-tuning · Adapters · Training-free adaptation · Few-shot learning

7.1 Introduction

Driven by advances in large-scale computation and datasets, the “pretraining and fine-tuning” strategy has become a cornerstone in the vision domain. This approach involves training models on comprehensive datasets like ImageNet [30] before adapting them to downstream tasks. Recently, Contrastive Language-Image Pretraining (CLIP) [44] has been introduced, which leverages contrastive learning

R. Zhang (✉)
The Chinese University of Hong Kong, Hong Kong, China
e-mail: renruizhang@link.cuhk.edu.hk

P. Gao
Shanghai Artificial Intelligence Laboratory, Shanghai, China
e-mail: gaopeng@pjlab.org.cn

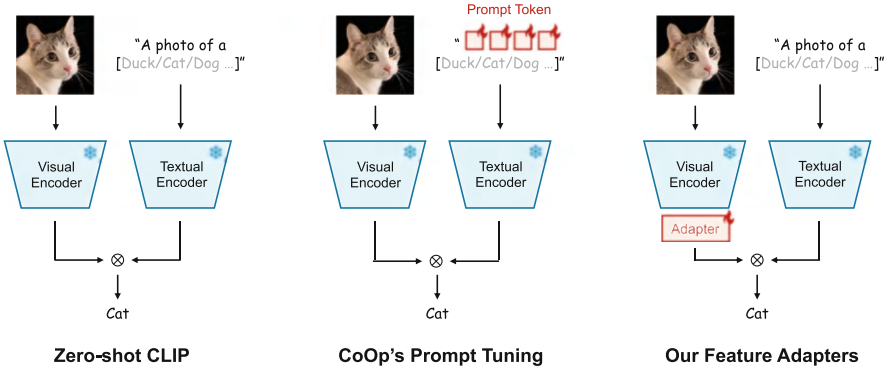


Fig. 7.1 Comparison of CLIP, CoOp [64], and our Methods with Feature Adapters. Different from CoOp’s prompt tuning, we discuss CLIP-Adapter [12] and Tip-Adapter(-F) [60] that append feature adapters on top of the frozen visual encoder of CLIP to enhance the downstream visual performance via few-shot learning

on vast collections of noisy image–text pairs. As a robust pretraining strategy, CLIP excels at zero-shot transfer by framing visual categories with carefully designed templates, allowing it to perform classification tasks without requiring labeled training data.

Although zero-shot learning using prompt engineering has shown encouraging results, crafting effective prompts for CLIP remains a complex and time-intensive task, often requiring domain-specific expertise. To address this, Context Optimization (CoOp) [64] introduced a method for learning continuous soft prompts from few-shot examples, replacing the reliance on manually designed templates. CoOp significantly enhances few-shot classification performance compared to zero-shot CLIP and linear probe CLIP, highlighting the potential of prompt-based tuning for vision-language models. In this chapter, we explore a different pathway for adapting vision-language models by focusing on feature adapters rather than prompt tuning, as compared in Fig. 7.1.

Motivated by parameter-efficient transfer learning techniques [23], we first discuss **CLIP-Adapter**, a lightweight and efficient approach that enhances CLIP’s few-shot performance by fine-tuning only a small set of additional parameters. Instead of modifying all CLIP parameters, CLIP-Adapter introduces a bottleneck architecture that adds two linear layers following the last layer of either the vision or language backbone. This design prevents overfitting in low-data scenarios while maintaining computational efficiency. Unlike the original adapter modules [23], which are embedded across all layers of a backbone, CLIP-Adapter focuses on the final layers, as CLIP’s pretrained encoder already provides strong representations that require only minimal residual adjustments. Additionally, CLIP-Adapter employs a residual-style blending mechanism, which mixes the original zero-shot visual or language embeddings with the fine-tuned features via a residual connection. This blending allows the model to effectively integrate the rich knowledge from

CLIP’s pretrained representations with new information from few-shot examples, achieving a balance between generalization and adaptation.

While CLIP-Adapter has demonstrated powerful few-shot performance, it still requires additional training and computational resources to optimize the newly introduced parameters. Such downstream training might limit the inherent generalization capability of the zero-shot CLIP. This raises a critical question: *Can we achieve the best of both worlds, leveraging CLIP’s training-free property for zero-shot classification while also attaining the strong performance of training-based few-shot methods?*

Given this, we further discuss **Tip-Adapter**, a novel training-free adaptation method for CLIP that appends the weight-frozen CLIP model with a non-parametric adapter. Unlike existing methods, Tip-Adapter eliminates the need for training by utilizing a cache model built from the few-shot dataset. Specifically, it extracts visual features using CLIP’s visual encoder, transforms corresponding labels into one-hot encodings, and constructs a cache model where these features and encodings act as paired keys and values. During inference, Tip-Adapter predicts outputs by computing the similarity between the test image’s features and the cached keys, aggregating the corresponding cached values for the final prediction. This prediction is combined with CLIP’s original output via a residual connection, effectively blending the knowledge from both pretrained CLIP and the few-shot dataset. This approach maintains the training-free efficiency of CLIP while achieving competitive performance, comparable to fine-tuned methods like CoOp and CLIP-Adapter.

To further enhance Tip-Adapter, we introduce **Tip-Adapter-F**, a fine-tuned version that unfreezes the cached keys as learnable parameters and optimizes them through a few training epochs. Remarkably, Tip-Adapter-F requires only 20 epochs to achieve state-of-the-art performance on ImageNet [7], significantly reducing the training overhead compared to CoOp and CLIP-Adapter, which require up to 200 epochs. As demonstrated in Table 7.1, our methods provide an excellent trade-off between accuracy, training efficiency, and inference speed, establishing Tip-Adapter and Tip-Adapter-F as practical solutions for efficient few-shot learning.

Through extensive experiments across various downstream tasks, we highlight the versatility and effectiveness of our discussed adapters, offering a novel pathway for enhancing CLIP’s performance while preserving its computational advantages.

7.2 Related Work

7.2.1 Vision-language Models

The integration of vision and language has emerged as a key area of research in artificial intelligence. Early advancements were led by attention-based models, such as bottom-up top-down attention [2], BAN [27], Intra-inter [14], and MCAN [57], which achieved notable success in multimodal tasks. The introduction

Table 7.1 Comparison of Performance and Efficiency. We report the classification accuracy (%) for different methods on 16-shot ImageNet [7]. All experiments are tested with batch size 32 on a single NVIDIA GeForce RTX 3090 GPU. The column in blue records the gain to Zero-shot CLIP

Models	Training	Epochs	Time	Accuracy	Gain	Infer. speed	GPU Mem.
Zero-shot CLIP [44]	Free	0	0	60.33	0	10.22 ms	2227 MiB
Linear-probe CLIP [44]	Required	–	13 min	56.13	–4.20	–	–
CoOp [64]	Required	200	14 h 40 min	62.95	+2.62	299.64 ms	7193 MiB
CLIP-adapter	Required	200	50 min	63.59	+3.26	10.59 ms	2227 MiB
Tip-adapter	Free	0	0	62.03	+1.70	10.42 ms	2227 MiB
Tip-adapter-F	Required	20	5 min	65.51	+5.18	10.53 ms	2227 MiB

of BERT [9] inspired the development of multimodal reasoning frameworks, including ViLBERT [39], LXMERT [49], UNITER [5], Oscar [32], ALBEF [33], and BEiT [51], which pushed the boundaries of cross-modal understanding. More recently, contrastive learning techniques like CLIP [44] and ALIGN [24] have showcased the power of aligning visual and textual representations using large-scale image–text datasets. These methods have set benchmarks in domains such as 3D reasoning [62, 65], video analysis [36], and depth understanding [59]. However, despite their successes, bridging the gap between CLIP’s zero-shot capabilities and the performance of fully supervised training remains an open challenge. CoOp addresses this issue by introducing continuous prompt optimization, improving vision-language models through refined textual prompts. While CoOp focuses on enhancing prompt designs, our methods, CLIP-Adapter and Tip-Adapter(-F), provide an alternative by employing lightweight feature adapters for efficient fine-tuning. This approach offers a straightforward yet effective solution for adapting vision-language models to downstream tasks.

7.2.2 *Prompt Engineering*

Prompt design [38] has gained significant attention, particularly following the success of the GPT series [4, 43]. GPT-3 demonstrated the ability of large-scale autoregressive models to perform diverse NLP tasks in a zero-shot or few-shot fashion, without requiring modifications. This paradigm [38] has inspired numerous prompt-based strategies. Initial methods emphasized discrete prompts, employing either handcrafted textual templates [1, 13, 50, 52, 55, 56] or automated prompt engineering techniques to mine or generate suitable templates [13, 26, 47]. In contrast, continuous prompts, which bypass the constraints of discrete language tokens, have been widely adopted in NLP [16, 31, 34, 37] and recently extended to vision tasks [25]. CLIP introduced a large-scale contrastive framework, demonstrating the viability of prompt-based zero-shot visual classification. Building on CLIP, CoOp [64] optimized continuous prompts, achieving significant improvements over manually designed discrete prompts in vision tasks. However, while prompt optimization has proven effective, it is not the only path for enhancing vision-language models. In this work, we explore fine-tuning small, lightweight feature adapters as an alternative. This method simplifies the adaptation process while achieving performance comparable to, or better than, prompt-based approaches.

7.2.3 *Transfer Learning*

Deep neural networks owe much of their success to the availability of large-scale, high-quality datasets [30]. However, creating such datasets is resource-intensive. Transfer learning has become a cornerstone of modern AI, enabling models to

leverage pretrained knowledge and adapt to various downstream tasks. Supervised pretraining on large datasets [7] has long been the standard, forming the basis for fine-tuning on applications such as object detection [46] and semantic segmentation [19]. Self-supervised approaches, including MoCo [20] and BYOL [15], mitigate the reliance on labeled data by learning features through contrastive pretext tasks. Recent advancements, such as CLIP [44], DeCLIP [35], and ALIGN [24], have further demonstrated the potential of contrastive learning on vision-language pairs, producing transferable features for zero-shot recognition across diverse datasets. Efforts to enhance CLIP’s few-shot performance include CoOp [64], which optimizes learnable prompts while freezing pretrained weights. Our discussed adapter-based methods follow a similar philosophy by preserving CLIP’s pretrained knowledge and introducing lightweight trainable modules. However, our approach emphasizes simplicity and efficiency, while achieving competitive performance.

7.3 Method

7.3.1 A Revisit of CLIP

CLIP is a vision-language model designed to align images with their corresponding textual descriptions. It employs two independent encoders: one for processing visual features and another for encoding textual inputs. During training, CLIP leverages a contrastive loss to learn a shared embedding space, where image and text representations are matched. The model is trained on a large-scale dataset of 400 million image-text pairs sourced from the internet, allowing it to associate images with a vast array of semantic concepts. This capability enables CLIP to perform zero-shot classification by aligning images with an open vocabulary of textual descriptions.

For a dataset with K “unseen” classes, CLIP constructs textual inputs by embedding the category names into a predefined prompt template. These prompts are processed through the textual encoder to create a zero-shot classifier, represented as $W_c \in \mathbb{R}^{K \times C}$, where each row encodes the pretrained weights of a category as a C -dimensional textual feature. Simultaneously, the visual encoder processes the features of a test image into $f_c \in \mathbb{R}^{1 \times C}$. The classification logits, which represent the similarity between the test image and the textual categories, are calculated as:

$$\text{logits} = f_c W_c^T. \quad (7.1)$$

This procedure bypasses the need for additional training on new datasets, leveraging CLIP’s pretrained encoders to deliver strong zero-shot classification performance.

7.3.2 CLIP-Adapter

Unlike prompt tuning in CoOp [64], which focuses on optimizing textual prompts, the adapter [23] is a lightweight neural module designed for parameter-efficient fine-tuning of pretrained models, such as BERT [8] in natural language processing. CLIP, on the other hand, learns visual representations through contrastive learning on large-scale image-text pairs, achieving remarkable zero-shot classification performance. Inspired by the adapter concept, we discuss CLIP-Adapter, which facilitates the transfer of CLIP’s knowledge to few-shot classification tasks. This is achieved by appending a lightweight two-layer Multi-Layer Perceptron (MLP) to the pretrained, frozen CLIP model, where the adapter predicts residual features for each input image.

The residual connection in CLIP-Adapter fuses the prior visual knowledge encoded by CLIP with the updated features derived from the few-shot training set, enabling effective adaptation with minimal additional parameters.

As illustrated in Fig. 7.2, for an input image, its L2-normalized feature f_c is first extracted using CLIP’s pretrained visual encoder. A two-layer MLP adapter, parameterized by W_1, b_1, W_2, b_2 , then refines this feature to produce the adapted feature f_a :

$$f_a = \varphi(f_c W_1^T + b_1) W_2^T + b_2, \quad (7.2)$$

where φ denotes the activation function in the MLP. The adapted feature f_a is then linearly combined with the pretrained feature f_c using a hyperparameter $\alpha \in [0, 1]$, resulting in the final classification logits:

$$\text{logits} = \alpha f_a W_c^T + f_c W_c^T. \quad (7.3)$$

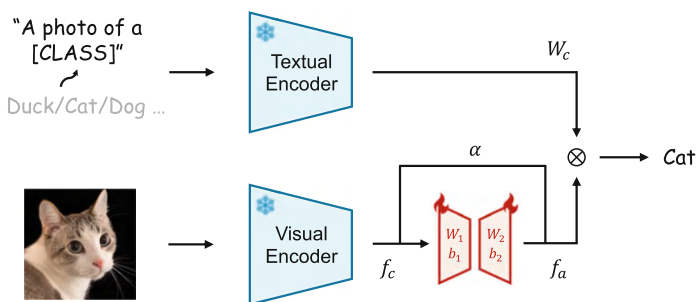


Fig. 7.2 Overview of CLIP-Adapter. We append a learnable MLP with a residual connection as the feature adapter after CLIP’s frozen visual encoder, enhancing few-shot image classification by injecting downstream knowledge

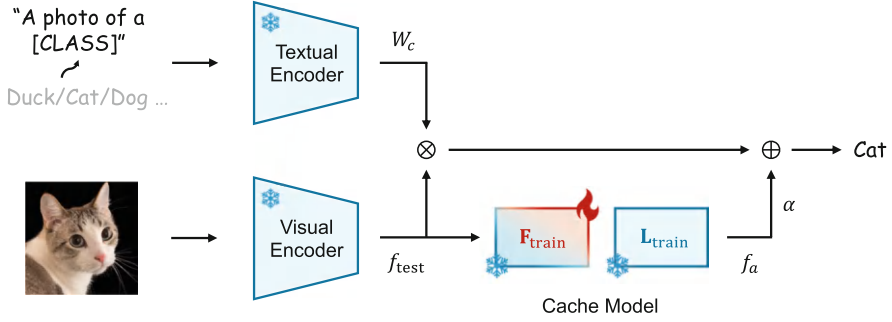


Fig. 7.3 Overview of Tip-Adapter (F). To further improve the efficiency and effectiveness of CLIP-Adapter, we construct a cache model to introduce a training-free method, Tip-Adapter (F_{train} frozen), and its fine-tuned variant, Tip-Adapter-F (F_{train} trainable)

Here, W_c represents the weights of the text classifier, which are constructed following the zero-shot CLIP method. Each category name is embedded into a predefined prompt template and encoded using CLIP’s pretrained textual encoder.

7.3.3 Tip-Adapter

While CLIP-Adapter offers significant improvements in few-shot classification, it requires additional training, which introduces computational costs and slower adaptation compared to the original zero-shot CLIP. To address this limitation, we discuss Tip-Adapter, a training-free adaptation method designed to enhance CLIP’s few-shot classification performance. As illustrated in Fig. 7.3, Tip-Adapter leverages a non-parametric key-value cache model constructed from the few-shot training data.

7.3.3.1 Cache Model Construction

Given a pretrained CLIP model and a new dataset with K -shot N -class training samples for few-shot classification, the dataset consists of K annotated images per class across N categories, denoted as I_K with their labels L_N . The goal is to construct a key-value cache model that acts as a feature adapter, encapsulating the few-shot knowledge within the N classes. For each training image, we use CLIP’s pretrained visual encoder to extract a C -dimensional L2-normalized feature and convert its ground-truth label into an N -dimensional one-hot vector. Collectively, the visual features and one-hot label vectors for all NK training samples are

represented as $\mathbf{F}_{\text{train}} \in \mathbb{R}^{NK \times C}$ and $\mathbf{L}_{\text{train}} \in \mathbb{R}^{NK \times N}$, where:

$$\mathbf{F}_{\text{train}} = \text{VisualEncoder}(I_K), \quad (7.4)$$

$$\mathbf{L}_{\text{train}} = \text{OneHot}(L_N). \quad (7.5)$$

In the key-value cache model, the CLIP-encoded representations $\mathbf{F}_{\text{train}}$ are treated as keys, while the one-hot label vectors $\mathbf{L}_{\text{train}}$ serve as their corresponding values. This cache model effectively stores all the new knowledge extracted from the few-shot training set, which can then be used to update the prior knowledge in CLIP.

7.3.3.2 Training-Free Adapter

Once the cache model is constructed, adapting CLIP for few-shot classification can be achieved through efficient matrix-vector multiplications. During inference, the L2-normalized feature of a test image, $f_{\text{test}} \in \mathbb{R}^{1 \times C}$, is extracted using CLIP's visual encoder and serves as a query for the key-value cache. The affinities between the query and the keys in the cache are computed as:

$$A = \exp(-\beta(1 - f_{\text{test}}\mathbf{F}_{\text{train}}^T)), \quad (7.6)$$

where $A \in \mathbb{R}^{1 \times NK}$ represents the affinity matrix, and β is a hyperparameter that modulates the sharpness of the affinities. Since both the query and key features are L2-normalized, $f_{\text{test}}\mathbf{F}_{\text{train}}^T$ directly corresponds to the cosine similarities between the test feature f_{test} and all few-shot training features $\mathbf{F}_{\text{train}}^T$. The exponential function ensures the similarities are transformed into non-negative values, with β controlling the sensitivity to differences in similarity. The prediction from the cache model is then obtained by weighting the cached values $\mathbf{L}_{\text{train}}$ using the affinities A , resulting in $A\mathbf{L}_{\text{train}} \in \mathbb{R}^{1 \times N}$.

In addition to the few-shot knowledge retrieved from the cache, Tip-Adapter incorporates the prior knowledge of the pretrained CLIP model. This is computed as $f_{\text{test}}W_c^T \in \mathbb{R}^{1 \times N}$, where W_c represents the classifier weights generated by CLIP's pretrained textual encoder. The final logits for the test image are obtained by combining these two components through a residual connection:

$$\begin{aligned} \text{logits} &= \alpha A\mathbf{L}_{\text{train}} + f_{\text{test}}W_c^T \\ &= \alpha\varphi(f_{\text{test}}\mathbf{F}_{\text{train}}^T)\mathbf{L}_{\text{train}} + f_{\text{test}}W_c^T, \end{aligned} \quad (7.7)$$

where α denotes the residual ratio, and we define $\varphi(x) = \exp(-\beta(1 - x))$. The first term adaptively summarizes the information from the few-shot training data, while the second term retains the prior knowledge encoded in CLIP's classifier W_c^T . The parameter α balances these two terms. Empirically, α should be larger when the domain gap between the pretrained and downstream few-shot tasks is significant,

as the model requires more knowledge from the few-shot dataset. Conversely, α is smaller when the domain gap is minimal.

7.3.4 *Tip-Adapter-F*

While Tip-Adapter significantly enhances CLIP’s performance by leveraging few-shot training data, its training-free nature can cause it to lag behind training-required methods such as CoOp and CLIP-Adapter as the number of shots increases. To bridge this gap while maintaining efficiency, we discuss Tip-Adapter-F, which treats the keys in the cache model as an effective initialization for learnable parameters and fine-tunes them using stochastic gradient descent (SGD).

Leveraging the strong starting point provided by the cache model, Tip-Adapter-F achieves state-of-the-art performance with only 20 epochs of fine-tuning on ImageNet [7], significantly reducing the training effort compared to the 200 epochs required by CoOp and CLIP-Adapter.

Specifically, Tip-Adapter-F unfreezes the cached keys $\mathbf{F}_{\text{train}}$ while keeping the values $\mathbf{L}_{\text{train}}$ and the pretrained CLIP encoders frozen. This approach focuses on improving the quality of the affinity estimation by updating the cached keys, thereby enabling more accurate similarity calculations between test and training features. In contrast, the cached values, represented as one-hot encodings of the ground-truth annotations, are left unchanged to ensure the category information remains intact.

7.3.5 *Relations Between CLIP-Adapter and Tip-Adapter(-F)*

A comparison of Eqs. (7.3) and (7.7) reveals the different mechanisms through which Tip-Adapter and CLIP-Adapter incorporate few-shot knowledge. Tip-Adapter retrieves knowledge directly from the cache model, while CLIP-Adapter utilizes a learnable adapter to predict the adapted features, which are then combined with CLIP’s W_c^T to produce the final output.

With further analysis, CLIP-Adapter can be regarded as a specific case of Tip-Adapter under the following conditions:

$$W_1 = \mathbf{F}_{\text{train}}, \quad W_2 = \mathbf{L}_{\text{train}}^T W_c^{-1}, \quad b_1 = 0, \quad b_2 = 0, \quad (7.8)$$

$$\varphi(x) = \exp(-\beta(1 - x)), \quad \text{where } x \in [0, 1]. \quad (7.9)$$

Despite this similarity, three key distinctions exist between the two methods:

1. **Initialization of Keys and Values:** CLIP-Adapter randomly initializes the keys (W_1) and values (W_2) and trains them using SGD. In contrast, Tip-Adapter directly constructs its cache model from the few-shot training features ($\mathbf{F}_{\text{train}}$) and one-hot label encodings ($\mathbf{L}_{\text{train}}$), making it non-parametric and training-free.

2. **Bottleneck Dimensions:** Tip-Adapter’s bottleneck dimension corresponds to the number of training samples (NK), allowing it to fully capture the information from the few-shot dataset. CLIP-Adapter, however, selects a lower bottleneck dimension to mitigate overfitting during training. This design enables Tip-Adapter to better leverage the capacity of large-scale pretrained models without overfitting.
3. **Activation Function:** Tip-Adapter introduces the activation function denoted in Eq. (7.8), which is naturally bounded between 0 and 1, as its inputs are cosine similarities in the normalized feature space. Conversely, CLIP-Adapter uses ReLU, which accommodates unbounded inputs.

In summary, compared to CLIP-Adapter, Tip-Adapter delivers an efficient, well-performing adapter without requiring training, making it particularly effective for few-shot classification tasks. On top of this, Tip-Adapter-F extends this advantage with minimal fine-tuning to achieve state-of-the-art performance.

7.4 Experiments

7.4.1 Settings

We evaluate the performance of CLIP-Adapter, Tip-Adapter, and Tip-Adapter-F on 11 widely-used image classification datasets: ImageNet [7], StanfordCars [29], UCF101 [48], Caltech101 [11], Flowers102 [41], SUN397 [54], DTD [6], EuroSAT [21], FGVCAircraft [40], OxfordPets [42], and Food101 [3]. For few-shot learning experiments, we evaluate models on 1, 2, 4, 8, and 16-shot training sets, testing against the full test sets of each dataset.

For the CLIP backbone, we employ ResNet-50 [18] as the visual encoder and a Transformer [10] as the textual encoder, both initialized with pretrained weights from [44]. These encoders remain frozen during all experiments. Data preprocessing follows the CLIP protocol [44], including random cropping, resizing, and random horizontal flipping. For prompt ensembling, we adopt the CLIP strategy, especially on ImageNet, while using single handcrafted prompts for the remaining 10 datasets. Tip-Adapter-F is fine-tuned using a batch size of 256, a learning rate of 0.001, and the AdamW optimizer [28] with a cosine scheduler. For most datasets, Tip-Adapter-F is trained for only 20 epochs, except for EuroSAT, where we allow 100 epochs of fine-tuning due to its unique characteristics.

We compare our models with three baseline methods:

1. **Zero-shot CLIP [44]**, which performs classification purely based on pretrained knowledge without any additional training.
2. **Linear-probe CLIP [44]**, which adds a linear classifier trained on the few-shot datasets while keeping CLIP weights frozen.

3. **CoOp** [64], which optimizes learnable textual prompts. For fair comparison, we use CoOp’s best-performing configuration: placing the class token at the end of a 16-token prompt without class-specific contexts.

To ensure fairness, we report official results from prior works for these baseline methods. This comprehensive evaluation highlights the efficiency and effectiveness of our discussed methods across diverse datasets and few-shot settings.

7.4.2 Comparison on ImageNet

7.4.2.1 Performance Analysis

As illustrated in Fig. 7.4 and Table 7.2, our adapter-based approaches consistently outperform other methods in few-shot image classification on ImageNet. Compared to the prompt-based CoOp, CLIP-Adapter demonstrates superior accuracy across all shot settings, establishing itself as a robust alternative for fine-tuning. Additionally, Tip-Adapter achieves consistent performance gains over Zero-shot CLIP without requiring any training. In low-data scenarios, Tip-Adapter significantly outperforms Linear-probe CLIP. Tip-Adapter-F, which fine-tunes the keys in the cache model, further enhances performance and achieves state-of-the-art results across all few-

Fig. 7.4 Comparison of few-shot classification accuracy on ImageNet [7] (The figure is sourced from [60])

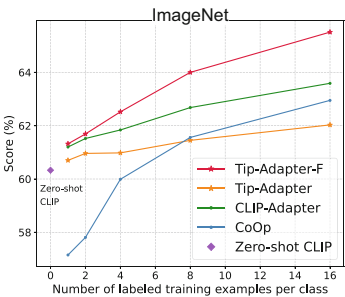


Table 7.2 Classification accuracy (%) on ImageNet [7] of different models with quantitative values. The last row in blue records the performance gain of Tip-Adapter-F brought by fine-tuning over Tip-Adapter

Few-shot setup	1	2	4	8	16
Zero-shot CLIP [44]: 60.33					
Linear-probe CLIP [44]	22.17	31.90	41.20	49.52	56.13
CoOp [64]	57.15	57.81	59.99	61.56	62.95
CLIP-adapter	61.20	61.52	61.84	62.68	63.59
Tip-adapter	60.70	60.96	60.98	61.45	62.03
Tip-adapter-F	61.32	61.69	62.52	64.00	65.51
	+0.62	+0.73	+1.54	+2.55	+3.48

shot configurations. The gap between Tip-Adapter-F and Tip-Adapter increases as the number of training samples grows, from +0.62% in the 1-shot setting to +3.44% in the 16-shot setting. This trend suggests that fine-tuning with additional training samples allows the network to construct a more robust and effective cache model.

7.4.2.2 Efficiency Comparison

Table 7.1 highlights the comparison of training time and inference speed across different models. CLIP-Adapter, Tip-Adapter, and Tip-Adapter-F benefit from caching textual features from CLIP, which can be precomputed and reused during training or inference. In contrast, CoOp employs learnable prompts that require repeated computation through the entire textual encoder during every training iteration. Linear-probe CLIP relies on logistic regression [53], which prevents direct measurement of training time by epochs or inference speed on GPUs. The results show that CoOp incurs the highest training cost while offering a modest +2.26% performance improvement over Zero-shot CLIP. CLIP-Adapter reduces training time compared to CoOp while achieving a higher accuracy gain of +3.26%. However, it still requires extensive training, with 200 epochs. Tip-Adapter, leveraging its training-free cache model, achieves a performance improvement of +1.70% over Zero-shot CLIP without any additional training, offering an excellent trade-off between efficiency and performance. Tip-Adapter-F further combines efficiency with high accuracy, achieving state-of-the-art results with only 1/10th of the training epochs required by CLIP-Adapter and CoOp. This approach effectively balances accuracy and efficiency.

7.4.3 Performance on Other Datasets

Figure 7.5 presents the performance comparison across the 10 additional datasets introduced in Sect. 7.4.1. Both CLIP-Adapter and the training-free Tip-Adapter achieve significant improvements in classification accuracy compared to Zero-shot CLIP. Moreover, they outperform CoOp in 1-shot and 2-shot settings on most datasets, showcasing their effectiveness in low-data scenarios. While Tip-Adapter underperforms CoOp and CLIP-Adapter when trained with more shots, Tip-Adapter-F, with its efficient few-epoch fine-tuning, eliminates this performance gap and surpasses all other methods, achieving the best results across all datasets. The consistent superiority of Tip-Adapter-F across these 10 datasets highlights its versatility.

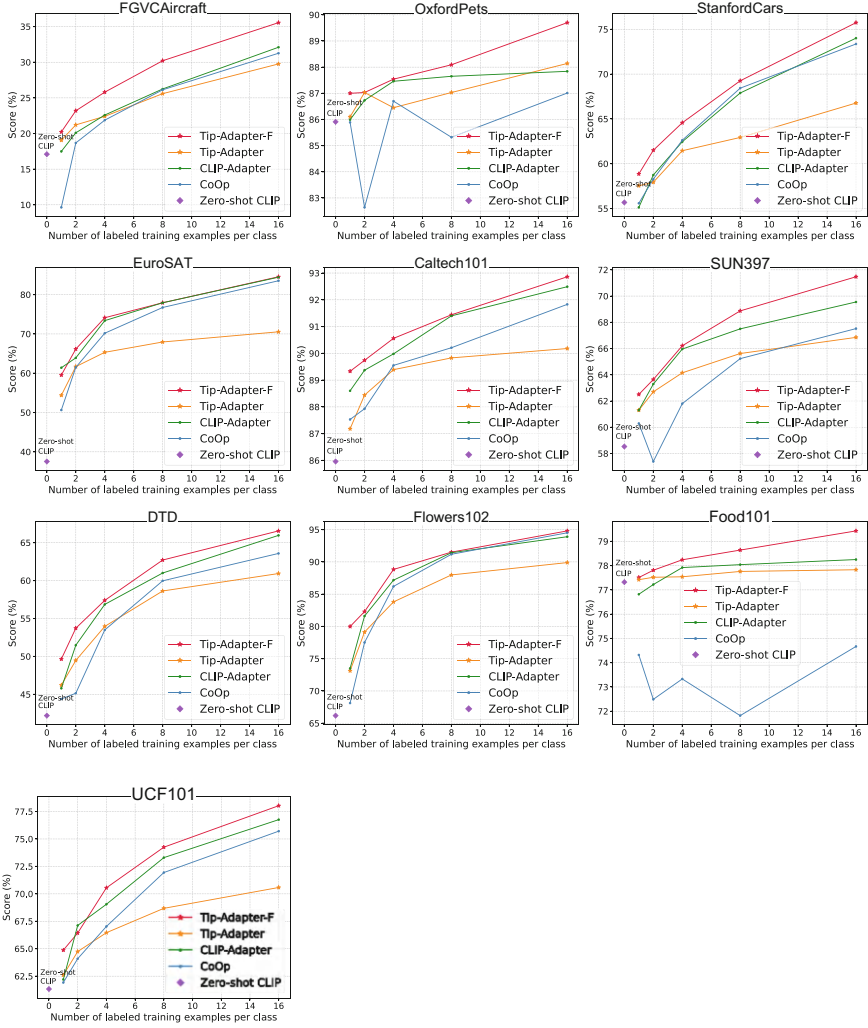


Fig. 7.5 Few-shot classification accuracy of different models on 10 datasets. CLIP-Adapter outperforms CoOp in most few-shot settings. Tip-Adapter largely improves Zero-shot CLIP without any training. Tip-Adapter-F consistently surpasses all compared methods by efficiently fine-tuning the cache model (The figures are sourced from [60])

7.4.4 Ablation Study

7.4.4.1 Vision Backbone

To evaluate the versatility of our methods across different architectures, we implemented models using a range of visual encoders based on ResNet [18] and

Table 7.3 Classification accuracy (%) of different visual encoders on 16-shot ImageNet [7]. ViT-B/32 and ViT-B/16 denote ViT-Base [10] with the patch size 32×32 and 16×16 , respectively, and RN50 $\times$ 16 denotes ResNet-50 [18] with 16 times more computation [44]

Models	ResNet-50	ResNet-101	ViT-B/32	ViT-B/16	RN50 $\times$ 16
Zero-shot CLIP [44]	60.33	62.53	63.80	68.73	70.94
CoOp [64]	62.95	66.60	66.85	71.92	–
CLIP-adapter	63.59	65.39	66.19	71.13	–
Tip-adapter	62.03	64.78	65.61	70.75	72.95
Tip-adapter-F	65.51	68.56	68.65	73.69	75.81

Table 7.4 Performance comparison of adapters inserted into different layers of CLIP-Adapter. We adopt ViT-B/16 as the image backbone, and report the 16-shot ImageNet performance

Insert Layers	All	12	10	8	6	4	2	0
Accuracy (%)	67.67	70.88	71.85	70.55	70.03	70.14	69.43	69.05
GPU memory (GiB)	5.98	2.22	2.65	2.89	3.23	4.46	4.88	5.14
Parameters (M)	5.20	0.52	0.78	0.78	0.78	0.78	0.78	0.78

ViT [10] backbones, as summarized in Table 7.3. Across all backbones and shot configurations, CLIP-Adapter consistently outperforms CoOp, demonstrating the strong generalization capability of our adapter-based approach. Furthermore, Tip-Adapter-F remains the best-performing method in every setting, underlining the effectiveness of fine-tuning the cache model. Additionally, as backbone sizes increase and transformer-based architectures are employed, the performance progressively improves, showcasing the scalability of our methods with more advanced models.

7.4.4.2 Insertion Position

By default, adapters are placed at the end of CLIP’s visual encoder. Experiments with CLIP-Adapter using the ViT-B/16 backbone (Table 7.4) show that inserting the adapter at the 12th layer (end) achieves the best performance with minimal computational cost. Inserting at earlier layers increases computational overhead and risks disrupting CLIP’s pretrained knowledge, while distributing adapters across all layers adds significant parameters (5.20M vs. 0.52M at the 12th layer), leading to overfitting on few-shot data. Placing the adapter at the final layer balances efficiency and performance, preserving pretrained knowledge while enhancing few-shot classification.

7.4.4.3 Prompt Design

We adopt the default prompt ensemble of seven templates from [44] for Zero-shot CLIP, CLIP-Adapter, and Tip-Adapter. To assess the impact of prompt variations,

Fig. 7.6 Classification performance with different prompt designs: single prompt (Cyan), prompt ensembling (Orange) and learnable prompt (Purple)

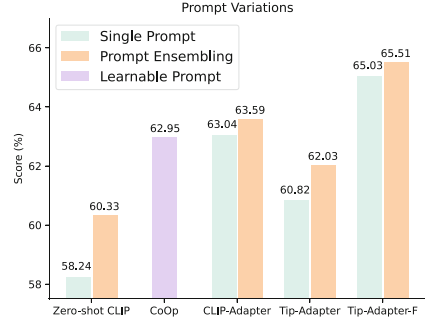


Table 7.5 The robustness (%) to distribution shift of different methods. The last row in blue records the performance gain of Tip-Adapter-F brought by further fine-tuning over Tip-Adapter

Datasets	Source	Target	
	ImageNet	-V2	-Sketch
	[7]	[45]	[22]
Zero-Shot CLIP [44]	60.33	53.27	35.44
Linear Probe CLIP [44]	56.13	45.61	19.13
CoOp [64]	62.95	54.58	31.04
CLIP-Adapter	63.59	55.69	35.68
Tip-Adapter	62.03	54.60	35.90
Tip-Adapter-F	65.51	57.11	36.00
	+3.48	+2.51	+0.10

we also evaluate these methods using a single prompt, “a photo of a [CLASS].”, as shown in Fig. 7.6. Results indicate a slight performance drop when using a single prompt. The decrease is more pronounced for Tip-Adapter and Zero-shot CLIP, while Tip-Adapter-F and CLIP-Adapter show greater resilience, suggesting that better-performing models are less sensitive to prompt variations.

7.4.5 Distribution Shift

We examine the out-of-distribution robustness of our adapter-based methods by training on one dataset and testing on another. Using ImageNet [7] with a 16-shot training set as the source dataset, we evaluate on two target datasets: ImageNetV2 [45] and ImageNet-Sketch [22], which share compatible categories with ImageNet but differ semantically. Table 7.5 shows that Tip-Adapter, even without training, demonstrates superior robustness to distribution shifts, outperforming CoOp [64] on ImageNet-V2 and CLIP-Adapter on ImageNet-Sketch. This highlights the advantage of the cache model, whose training-free construction mitigates overfitting risks associated with the source dataset. Tip-Adapter-F further combines the strengths of both worlds, achieving strong out-of-distribution performance from the cache model while maintaining leading in-distribution accuracy through fine-tuning.

7.4.6 Conclusion

In this chapter, we introduced CLIP-Adapter and its non-parametric extension, Tip-Adapter, as efficient and effective solutions for enhancing CLIP’s few-shot image classification capabilities. CLIP-Adapter replaces CoOp’s prompt tuning by lightweight bottleneck layers, incorporating a residual connection to dynamically blend zero-shot knowledge with adapted features. Building on CLIP-Adapter, Tip-Adapter eliminates the need for SGD-based training by leveraging a cache model constructed from few-shot datasets, enabling it to function as a training-free adapter. Remarkably, Tip-Adapter achieves comparable or even superior results to its parametric counterpart while maintaining high computational efficiency. Additionally, Tip-Adapter can be fine-tuned for just a few epochs to achieve further performance gains, termed Tip-Adapter-F, striking an excellent balance between efficiency and accuracy. We hope our investigation may inspire future work to further explore feature adapters for the downstream adaption of CLIP.

Further Reading

Interested readers may refer to the following material:

- Zhang et al. [61] for improved version of Tip-Adapter by integrating different vision-language foundation models.
- Zhu et al. [66] for more lightweight version of Tip-Adapter by reducing the visual redundancy of CLIP features.
- Guo et al. [17] for zero-shot enhancement of CLIP by modulating the cross-modal attention distributions.
- Zhang et al. [63] for learning efficient feature adapters into Large Language Models (LLMs) for visual understanding.
- Zhang et al. [58], Zhu et al. [65] for learning efficient feature adapters to solve 3D point cloud tasks.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., Simonyan, K.: Flamingo: A visual language model for few-shot learning. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) *Advances in Neural Information Processing Systems* (2022)
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: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018)

3. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101—mining discriminative components with random forests. In: *European Conference on Computer Vision*, pp. 446–461. Springer (2014)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020)
5. Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Learning universal image-text representations. In: *ECCV* (2020)
6. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613 (2014)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE (2009)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (2019)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations* (2021)
11. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pp. 178–178. IEEE (2004)
12. Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., Qiao, Y.: Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544* (2021)
13. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (2021)
14. Gao, P., Jiang, Z., You, H., Lu, P., Hoi, S.C., Wang, X., Li, H.: Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6639–6648 (2019)
15. Grill, J.B., Strub, F., Althé, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733* (2020)
16. Gu, Y., Han, X., Liu, Z., Huang, M.: Ppt: Pre-trained prompt tuning for few-shot learning. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8410–8423 (2022)
17. Guo, Z., Zhang, R., Qiu, L., Ma, X., Miao, X., He, X., Cui, B.: Calip: Zero-shot enhancement of clip with parameter-free attention. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 2023 Oral (2022)
18. 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)
19. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969 (2017)
20. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738 (2020)

21. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **12**(7), 2217–2226 (2019)
22. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271 (2021)
23. Hounsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for NLP. In: *International Conference on Machine Learning* (2019)
24. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: *International Conference on Machine Learning* (2021)
25. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: *ECCV*, pp. 709–727 (2022)
26. Jiang, Z., Xu, F.F., Araki, J., Neubig, G.: How can we know what language models know? *Trans. Assoc. Comput. Linguist.* **8**, 423–438 (2020)
27. Kim, J.H., Jun, J., Zhang, B.T.: Bilinear attention networks. *arXiv preprint arXiv:1805.07932* (2018)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
29. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561 (2013)
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems* (2012)
31. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (2021)
32. Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, August 23–28, 2020, Proceedings, Part XXX 16* (2020)
33. Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., Hoi, S.C.H.: Align before fuse: Vision and language representation learning with momentum distillation. *Adv. Neural Inf. Process. Syst.* **34**, 9694–9705 (2021)
34. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (2021)
35. Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., Yan, J.: Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208* (2021)
36. Lin, Z., Geng, S., Zhang, R., Gao, P., de Melo, G., Wang, X., Dai, J., Qiao, Y., Li, H.: Frozen clip models are efficient video learners. *ECCV 2022* (2022)
37. Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., Tang, J.: Gpt understands, too. *arXiv preprint arXiv:2103.10385* (2021)
38. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **55**(9), 1–35 (2023)
39. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: *Advances in Neural Information Processing Systems* (2019)
40. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151* (2013)

41. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, pp. 722–729. IEEE (2008)
42. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3498–3505. IEEE (2012)
43. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**, 9 (2019)
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
45. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: International Conference on Machine Learning, pp. 5389–5400. PMLR (2019)
46. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. **28**, 91–99 (2015)
47. Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In: Empirical Methods in Natural Language Processing (2020)
48. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
49. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490 (2019)
50. Tsimpoukelli, M., Menick, J.L., Cabi, S., Eslami, S., Vinyals, O., Hill, F.: Multimodal few-shot learning with frozen language models. Adv. Neural Inf. Process. Syst. **34**, 200–212 (2021)
51. Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O.K., Singhal, S., Som, S., et al.: Image as a foreign language: Beit pretraining for all vision and vision-language tasks. arXiv preprint arXiv:2208.10442 (2022)
52. Wang, Z., Yu, J., Yu, A.W., Dai, Z., Tsvetkov, Y., Cao, Y.: SimVLM: Simple visual language model pretraining with weak supervision. In: International Conference on Learning Representations (2022)
53. Wright, R.E.: Logistic regression. In: Grimm, L.G., Yarnold, P.R. (eds.) Reading and Understanding Multivariate Statistics, pp. 217–244. American Psychological Association, Washington, DC (1995)
54. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3485–3492. IEEE (2010)
55. Yao, Y., Zhang, A., Zhang, Z., Liu, Z., Chua, T.S., Sun, M.: Cpt: Colorful prompt tuning for pre-trained vision-language models. arXiv preprint arXiv:2109.11797 (2021)
56. Yao, Y., Chen, Q., Zhang, A., Ji, W., Liu, Z., Chua, T.S., Sun, M.: PEVL: Position-enhanced pre-training and prompt tuning for vision-language models. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 11104–11117 (2022)
57. Yu, Z., Yu, J., Cui, Y., Tao, D., Tian, Q.: Deep modular co-attention networks for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
58. Zhang, R., Guo, Z., Gao, P., Fang, R., Zhao, B., Wang, D., Qiao, Y., Li, H.: Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. arXiv preprint arXiv:2205.14401 (2022)
59. Zhang, R., Zeng, Z., Guo, Z.: Can language understand depth? In: Proceedings of the 30th ACM International Conference on Multimedia 2022 (2022)
60. Zhang, R., Zhang, W., Fang, R., Gao, P., Li, K., Dai, J., Qiao, Y., Li, H.: Tip-adapter: Training-free adaption of clip for few-shot classification. In: ECCV 2022. Springer Nature Switzerland (2022)
61. Zhang, R., Hu, X., Li, B., Huang, S., Deng, H., Li, H., Qiao, Y., Gao, P.: Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2023 (2023)

62. Zhang, R., Wang, L., Qiao, Y., Gao, P., Li, H.: Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2023* (2023)
63. Zhang, R., Han, J., Liu, C., Zhou, A., Lu, P., Qiao, Y., Li, H., Gao, P.: Llama-adapter: Efficient fine-tuning of large language models with zero-initialized attention. In: *International Conference on Learning Representations 2024* (2024)
64. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *Int. J. Comput. Vision* **130**, 1–12 (2022)
65. Zhu, X., Zhang, R., He, B., Zeng, Z., Zhang, S., Gao, P.: Pointclip v2: Adapting clip for powerful 3d open-world learning. In: *ICCV 2023* (2022)
66. Zhu, X., Zhang, R., He, B., Zhou, A., Wang, D., Zhao, B., Gao, P.: Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In: *ICCV 2023* (2023)

Chapter 8

Efficient Tuning of Vision Foundation Models with Neural Prompt Search



Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu

Abstract The size of vision models has grown exponentially in recent years, particularly with the rise of Vision Transformers. This rapid growth has driven the development of parameter-efficient tuning methods, such as learning adapter layers or low-rank adaptation layers, which enable fine-tuning of a small subset of model parameters while keeping the vast majority of pretrained parameters frozen. However, designing an effective tuning method is not straightforward: it often involves exploring numerous design choices, and each downstream dataset may require custom-tailored solutions. In this chapter, we introduce **Neural prOmppt seArcH (NOAH)**, a novel approach that leverages a neural architecture search algorithm to automatically learn the optimal design of prompt modules for large vision models, tailored specifically for each downstream dataset.

Keywords Prompt learning · Adapters · Low-rank adaptation · Neural architecture search · Foundation model · Fine-tuning · Image classification · Transformers

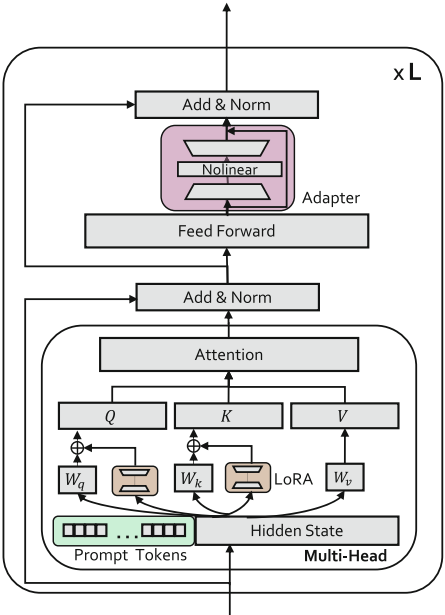
8.1 Introduction

In recent years, the size of vision models has grown dramatically, from tens of millions of parameters in earlier architectures (e.g., ResNet [12]) to hundreds of millions or even billions in today's Transformer-based models [2, 8, 10, 34, 35]. While this growth has led to significant advancements, it has also introduced challenges, particularly in transfer learning. Larger models are prone to overfitting

Y. Zhang · Z. Liu (✉)
S-Lab, Nanyang Technological University, Singapore, Singapore
e-mail: yuanhan002@e.ntu.edu.sg; ziwei.liu@ntu.edu.sg

K. Zhou
CS Department, Hong Kong Baptist University, Hong Kong, China
e-mail: kyzhou@hkbu.edu.hk

Fig. 8.1 Neural Prompt Search (NOAH) subsumes three representative parameter-efficient tuning methods—Adapter, LoRA, and VPT—and leverages neural architecture search to learn the optimal design from data. This approach is motivated by the observation that none of the three individual methods consistently outperforms the others across the VTAB-1k benchmark



when fine-tuned on typical datasets and demand significantly more computational and storage resources.

To address these challenges, the computer vision community has turned to parameter-efficient tuning methods [16–18]. These techniques aim to adapt pre-trained models by modifying only a small subset of their parameters, typically through the addition of compact, trainable components optimized for specific tasks. Prominent examples include Adapter [16], Low-Rank Adaptation (LoRA)[17], and Visual Prompt Tuning (VPT) [18]. As illustrated in Fig. 8.1, these methods differ in implementation: Adapter adds a small bottleneck-shaped network to a block’s output; LoRA uses rank-decomposed residual layers; and VPT introduces learnable “tokens” to the input of Transformer blocks, akin to adding artificial “pixels.”

However, our evaluation of these methods on the VTAB-1k benchmark [42], which encompasses 19 diverse vision datasets, reveals several limitations. *First*, no single method consistently performs well across all datasets. For instance, while VPT excels at scene structure understanding tasks like SmallNORB/azimuth [21], it struggles on others such as SmallNORB/elevation [21] and Clevr/count [19], where Adapter and LoRA perform better. This suggests the need for extensive evaluations to identify the best method for each dataset. *Second*, performance is highly sensitive to parameter choices, such as the feature dimensions in Adapter or token lengths in VPT, echoing observations from prior work [18].

This chapter discusses a unified approach termed Neural prOmpt seArCh (NOAH). NOAH treats existing parameter-efficient tuning methods as *prompt modules* and automates the search for optimal designs using a neural architecture

search (NAS) algorithm. Specifically, we incorporate Adapter, LoRA, and VPT into the search space of each Transformer block (see Fig. 8.1). A one-shot NAS algorithm determines the optimal configurations, such as Adapter dimensions or VPT token lengths. Unlike traditional NAS approaches, NOAH performs this search with a pretrained backbone frozen, focusing exclusively on parameter-efficient scenarios.

Our experiments on VTAB-1k demonstrate that NOAH significantly outperforms individual methods on 10 out of 19 datasets, with competitive performance on the remainder (see Fig. 8.1). Beyond standard image classification, we tested NOAH in few-shot learning and domain generalization, where it again outperformed handcrafted prompt modules. Further experiments on dense prediction and natural language tasks highlight NOAH’s versatility, showing that it can be easily extended to tasks requiring distinct modalities or supervision signals.

8.2 Related Work

8.2.1 *Parameter-Efficient Tuning*

The rapid increase in model sizes has sparked significant interest in developing parameter-efficient tuning methods [13, 16–18, 43, 46]. These methods aim to adapt large pretrained models for downstream tasks by modifying only a small subset of parameters or adding lightweight trainable modules. Broadly, these methods can be categorized into two groups: (1) Fine-tuning a small subset of parameters, such as biases [40]. (2) Adding small trainable modules, such as Adapter [16] or LoRA [17], which are most relevant to this work.

Adapter and LoRA share similar bottleneck-like architectures but differ in placement: Adapter modules are typically appended to the output of a network block, while LoRA modules act as residuals to the projection matrices in Transformer [35] blocks. These methods were initially introduced in natural language processing (NLP) [2, 8] to address the challenges of fine-tuning massive language models. More recently, their concepts have been adapted for vision-language models [5, 11, 38, 39].

Prompt learning [22, 23, 43] is another approach that turns textual prompts into learnable vectors and has inspired visual counterparts such as Visual Prompt Tuning (VPT) [18], which introduces learnable “pixels” as input to vision models like ViT [10]. However, as noted in recent studies [4, 9] and confirmed in our experiments, the effectiveness of these tuning methods varies significantly depending on the model architecture and dataset.

Recent works [7, 28, 47] have proposed enhancements to foundational methods like Adapter, LoRA, and VPT. While we focus on these foundational techniques in our experiments, the adaptability of our approach, NOAH, makes it promising for future integration with these advancements.

Efforts to unify parameter-efficient tuning methods have also emerged. For instance, [13] connects Adapter and prompt learning by framing them as the learning of modification vectors, providing a unified perspective and stronger baselines. UNIPELT [26] combines multiple prompt modules in a block and uses gating functions to activate them selectively. Our work differs in two critical aspects (1) Focus on computer vision: Previous studies [13, 26] primarily target NLP applications. (2) Fine-grained NAS-based unification: We unify prompt modules via neural architecture search (NAS), enabling precise control over hyper-parameters like token length and embedding dimensions. This approach is particularly suited for resource-constrained scenarios.

Recently, AutoPEFT [47] introduced NAS for selecting optimal prompt modules for NLP. While AutoPEFT and NOAH share the goal of optimizing prompt modules, they differ in methodologies and components. NOAH focuses on VPT, Adapter, and LoRA, using AutoFormer for NAS, whereas AutoPEFT incorporates Serial Adapter, Parallel Adapter, and prefix-tuning with Bayesian optimization.

8.2.2 Neural Architecture Search

NAS explores optimal network designs by combining two components: a search space and a search algorithm. Search spaces can encompass neuron connections [50], combinations of hyper-parameters [49], or arrangements of specific modules like normalization layers [45]. Search algorithms have evolved from computationally expensive approaches such as reinforcement learning [49] and evolutionary search [32] to efficient strategies like weight-sharing [30] and differentiable optimization [24].

Our work is most closely aligned with AutoFormer [3], a NAS framework for Transformers [35]. AutoFormer employs a weight-sharing strategy that allows subnets sampled from a larger supernet to share weights. We leverage AutoFormer to optimize parameter-efficient tuning methods, demonstrating its potential for enhancing efficient transfer learning. By integrating NAS into the tuning process, we hope to inspire future work to further refine and expand upon this paradigm.

8.3 Neural Prompt Search

8.3.1 Background

8.3.1.1 Vision Transformer (ViT)

We begin by briefly reviewing Vision Transformer (ViT) [10], the foundation of our approach. ViT consists of alternating blocks of multi-head self-attention (MSA) and multi-layer perceptron (MLP) modules. Given an input sequence $x \in \mathbb{R}^{N \times D}$,

where N denotes the token length and D is the embedding dimension, MSA maps x to queries $Q \in \mathbb{R}^{N \times d}$, keys $K \in \mathbb{R}^{N \times d}$, and values $V \in \mathbb{R}^{N \times d}$ using projection matrices $W_q \in \mathbb{R}^{D \times d}$, $W_k \in \mathbb{R}^{D \times d}$, and $W_v \in \mathbb{R}^{D \times d}$, respectively, where d is the hidden dimension. MSA then computes the weighted sum over the values using the self-attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \right) V, \quad (8.1)$$

where $\frac{1}{\sqrt{d}}$ is a scaling factor.

Below, we review three representative parameter-efficient tuning methods: Adapter [16], LoRA [17], and Visual Prompt Tuning (VPT) [18], all of which are incorporated into our search space. An illustration of these methods is provided in Fig. 8.1.

8.3.1.2 Adapter

The Adapter method introduces a bottleneck-like neural network with a down-sampling layer $W_{\text{down}} \in \mathbb{R}^{d \times r}$ and an up-sampling layer $W_{\text{up}} \in \mathbb{R}^{r \times d}$, where r is the reduced dimension. A non-linear activation function $\phi(\cdot)$, such as ReLU, is applied between these layers. The computation is defined as:

$$h' = \phi(h W_{\text{down}}) W_{\text{up}}, \quad (8.2)$$

where $h \in \mathbb{R}^{N \times d}$ is a normalized output of the MLP in a Transformer block.

8.3.1.3 LoRA

LoRA modifies the query and key projection layers, W_q and W_k , by introducing low-rank decomposition. Specifically, updates are represented as $\Delta W_q = W_q^{\text{down}} W_q^{\text{up}}$ and $\Delta W_k = W_k^{\text{down}} W_k^{\text{up}}$, where $W_q^{\text{down}}, W_k^{\text{down}} \in \mathbb{R}^{D \times r}$ and $W_q^{\text{up}}, W_k^{\text{up}} \in \mathbb{R}^{r \times d}$. For an input x , the updated queries and keys are computed as:

$$Q = x W_q + s \cdot x W_q^{\text{down}} W_q^{\text{up}}, \quad K = x W_k + s \cdot x W_k^{\text{down}} W_k^{\text{up}}, \quad (8.3)$$

where s is a fixed scaling parameter for modulating the updates.

8.3.1.4 Visual Prompt Tuning (VPT)

VPT introduces learnable tokens as additional input to a Transformer block, effectively appending virtual “pixels” in the input space. We focus on the VPT-

Deep variant [18], which applies prompt tuning across multiple layers. Specifically, for an input $x \in \mathbb{R}^{N \times D}$, comprising a class token [CLS] and image patch embeddings $E = e_i \mid e_i \in \mathbb{R}^D, i = 1, \dots, N - 1$, VPT prepends m learnable tokens $P = p_k \mid p_k \in \mathbb{R}^D, k = 1, \dots, m$, yielding:

$$x = [\text{CLS}, P, E]. \quad (8.4)$$

8.3.1.5 AutoFormer

AutoFormer [3] is an efficient neural architecture search (NAS) framework designed for Vision Transformers (ViTs). The methodology involves three key steps: defining the search space, training a supernet, and searching for the optimal subnet given predefined constraints. For a comprehensive overview, we refer readers to [3].

Search Space The AutoFormer search space includes architectural factors such as the number of layers (e.g., 14, 16, 18), the number of attention heads (e.g., 8, 10, 12), the embedding dimensions (e.g., 528, 624, 768), the MLP reduction ratios (e.g., 3, 4, 5), and the dimensions of query-key-value embeddings (e.g., 512, 640, 768).

Supernet Training The supernet is initialized with the largest configuration in the search space and trained by randomly sampling subnets during each step. Subnet weights are shared via a weight-entanglement strategy, enabling efficient training.

Subnet Search Once the supernet is trained, AutoFormer employs an evolutionary search algorithm to identify optimal subnets. Candidates are evaluated based on classification accuracy and model size, ensuring resource-efficient architectures.

8.3.2 Prompt Search Algorithm

As discussed, none of the individual parameter-efficient tuning methods, or *prompt modules* called in this chapter, shows dominance in the transfer learning benchmark. Our approach, neural prompt search (NOAH), incorporates Adapter [16], LoRA [17] and VPT [18] into each Transformer block and learns the design that best suits a dataset through neural architecture search (NAS). Specifically, we adopt AutoFormer [3], for prompt module search. NOAH adopts AutoFormer for the parameter-efficient tuning (PET) task by designing the specific search space for this task (please refer to Section 8.2.2 for details). Notable, other NAS approaches are also available and can be a future work to extend NOAH, but designing a specific search space is novel for PET tasks.

8.3.2.1 Search Space

As illustrated in Fig. 8.1, we integrate three prompt modules—Adapter, LoRA, and VPT—into each Transformer block based on their original designs [16–18]. Specifically, VPT is applied at the input position, LoRA is added alongside the projection matrices as residuals, and Adapter is inserted after the normalized output of the MLP.

The search space primarily focuses on the parameters of these prompt modules, with two key tunable factors for each module: (i) the embedding dimension $\in \{1, 5, 10, 50, 100\}$; (ii) the depth $\in \{3, 6, 9, 12\}$. The depth determines the layers to which the module is applied. For example, a depth of 3 for VPT means the first three layers (indexed 0, 1, 2) have VPT installed, while the remaining layers do not.¹ For VPT, the embedding dimension refers to the token length, whereas for Adapter and LoRA, it represents the down-sampled dimension, denoted as r .

8.3.2.2 Supernet Training

The supernet, derived from ViT-B/16, includes 12 Transformer layers, each augmented with three prompt modules initialized with their largest embedding dimension of 100. During training, five subnets are sampled per forward pass using an attentive sampling strategy [37], which focuses on optimizing subnets constrained by a predefined model size (e.g., 0.64M parameters). This sampling is complemented by the inclusion of the largest and smallest subnets. By targeting specific model sizes instead of training for all possible configurations, this strategy reduces computational overhead and mitigates convergence issues caused by uniform sampling.

Sampling a subnet involves two steps for each prompt module: (1) **Depth Selection:** A depth is randomly sampled from 3, 6, 9, 12, determining the layers containing the module. (2) **Embedding Dimension Selection:** For the chosen depth, an embedding dimension is sampled uniformly from 1, 5, 10, 50, 100.

NOAH employs a module-wise sampling approach, distinguishing it from AutoFormer’s block-wise sampling [3]. Unlike AutoFormer, which retains all internal modules (e.g., attention and MLP) within a block, NOAH allows selective retention of individual modules across blocks. For example, one module might be retained in all layers, while another is limited to shallow layers. This design leverages findings from VPT [18], which demonstrate that focusing VPT on shallow layers achieves better parameter efficiency. Furthermore, NOAH trains only the prompt module weights within the supernet, contrasting AutoFormer’s approach of training all parameters.

¹ We use zero-based indexing for clarity.

During training, NOAH uses weight entanglement [3], where weights are shared across prompt modules of different configurations. For instance, if 100 tokens are trained for a VPT module, previously trained weights for 50 tokens are reused and updated alongside the remaining tokens. This strategy accelerates convergence and reduces memory usage.

The formal representation of supernet training is as follows: For each training iteration, a subnet M with weights W is sampled, where:

$$W = W^0, W^i, \dots, W^l, M = M^0, M^i, \dots, M^l, \quad (8.5)$$

where i denotes the $i + 1$ Transformer block, and $l + 1$ is the total number of blocks. Adapter, LoRA, and VPT weights within each block are defined as:

$$W^i = W^i(V), W^i(A), W^i(L), M^i = M^i(V), M^i(A), M^i(L), \quad (8.6)$$

where $W^i(\cdot)$ and $M^i(\cdot)$ are sampled from respective candidate sets:

$$W^i(\cdot) \in W_0^i(\cdot), W_j^i(\cdot), \dots, W_s^i(\cdot), M^i(\cdot) \in M_0^i(\cdot), M_j^i(\cdot), \dots, M_s^i(\cdot), \quad (8.7)$$

where $W_j^i(\cdot)$ and $M_j^i(\cdot)$ represent candidate weights and modules, respectively. Weights are entangled as follows:

$$W_j^i(\cdot) \subseteq W_k^i(\cdot), ; \text{ for } j \leq k. \quad (8.8)$$

8.3.2.3 Evolutionary Search

Following supernet training, an evolutionary search identifies the optimal subnet architecture under a parameter size constraint [3]. The search begins with K randomly generated architectures, and the top- k performers are selected as parents for the next generation through crossover and mutation:

- **Crossover:** Two candidate architectures are combined to produce a child architecture by exchanging prompt module designs at identical layers.
- **Mutation:** Each module in a candidate undergoes a random modification (e.g., depth or embedding dimension) with a probability P_m .

After each epoch, the subnet pool $\mathcal{P}$ is updated by retaining only the top- k architectures based on validation performance. The process concludes after a predefined number of epochs.

8.4 Experiments

In this section, we address the following key questions: (i) Is NOAH better than the individual prompt modules? (ii) Can NOAH work in a few-shot setting? (iii) Are models learned by NOAH robust to domain shift? (iv) Can NOAH be extended to tasks beyond image classification? The answers to these questions are discussed in Sects. 8.4.2, 8.4.3, 8.4.4, and Sect. 8.4.5, respectively. In Sect. 8.4.5, we also delve deeper into NOAH, examining aspects such as the structure of the selected subnet and its transferability beyond the dataset used to discover its architecture (Fig. 8.2).

8.4.1 Baselines

The primary competitors are the three representative prompt modules incorporated into NOAH: Adapter [16], LoRA [17], and VPT [18]. Among these, only VPT was specifically designed for vision models, while the other two originated in language model research. We also compare NOAH with two commonly used fine-tuning methods on the VTAB-1k benchmark: **(1) Full Tuning (Full)**: Fine-tunes all parameters of the model. **(2) Linear Probing (Linear)**: Freezes the pretrained model and adjusts only the newly added linear classification layer.² It is noteworthy that Full tuning has been considered a strong baseline in prior studies [16, 18].

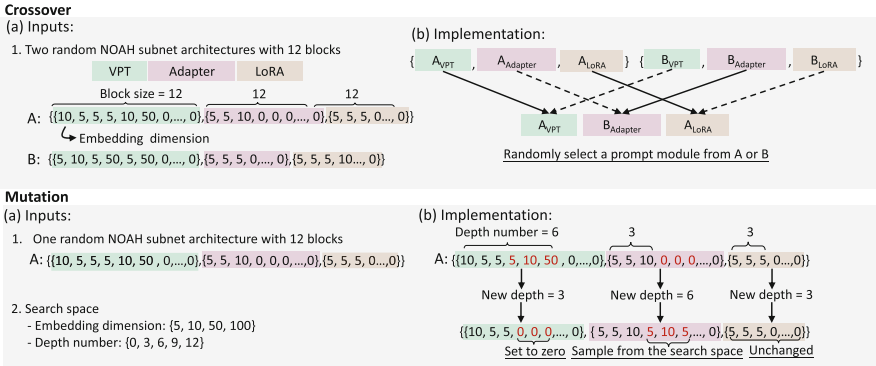


Fig. 8.2 Illustration of the three stages in NOAH. For each dataset, NOAH finds the optimal designs of the prompt modules through supernet training and evolutionary search. NOAH retrains the subnet, which performs the best on the validation set, to achieve better performance if needed

² Note that all methods include a newly added linear classification layer for learning.

8.4.2 Experiments on Image Classification

8.4.2.1 Datasets

To evaluate the transfer learning performance of our approach on image classification, we utilize the VTAB-1k benchmark [42]. VTAB-1k comprises 19 vision datasets categorized into three groups:

- **Natural:** Includes images captured by standard cameras, encompassing a broad range of concepts such as generic, fine-grained, and abstract objects.
- **Specialized:** Features images obtained from specialized equipment, such as aerial imagery for remote sensing and medical imaging.
- **Structured:** Focuses on scene structure understanding tasks, including object counting, depth prediction, and orientation prediction.

Each dataset in VTAB-1k contains 1,000 labeled examples, which are divided into an 80% train set and a 20% val set. The val set is used for hyperparameter tuning, while the original test set from each dataset serves as the evaluation set. For the final evaluation, models are trained on the full set of 1,000 examples for each dataset. Performance is measured using Top-1 classification accuracy.

8.4.2.2 Results

Figure 8.3 presents the complete results on the VTAB-1k benchmark. The average performance for each dataset group is also summarized in Fig. 8.3. Below are our key observations: *Observation 1: Overall, NOAH is the best parameter-efficient tuning method.* Our results demonstrate that searching for the optimal combination of prompt modules outperforms individual modules. NOAH achieves a 1% average improvement over the strongest individual module, LoRA. While a 1% gain may seem modest, it is significant given the diversity of the VTAB-1k benchmark. Notably, while Adapter has been shown to perform best in NLP tasks [26], LoRA outperforms Adapter in the computer vision domain, further confirming that search-based approaches are superior to manual engineering in identifying optimal configurations. *Observation 2: NOAH slightly underperforms in*

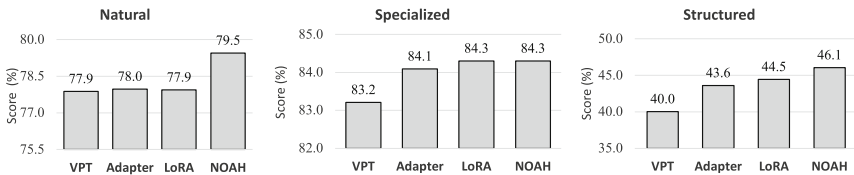


Fig. 8.3 Group-wise average results on VTAB-1k. NOAH performs the best in the Natural and Structured groups while its performance in the Specialized group is similar to that of LoRA—but NOAH does not require a manual search over the architecture and hyper-parameters

the Specialized group. While NOAH shows strong performance across most tasks, it slightly underperforms in the Specialized group. For instance, NOAH achieves an average accuracy of 84.9%, compared to LoRA’s 84.6%. On some datasets, such as EuroSAT, NOAH lags behind its competitors despite outperforming them on others, such as Retinopathy. This suggests that NOAH faces challenges in identifying the optimal configuration for Specialized tasks. The reduced stability during supernet training, especially for tasks less aligned with the pretraining domain (e.g., ImageNet-21K), likely contributes to this. In contrast, the Natural group benefits from more stable optimization due to closer alignment with the pretraining data. Despite this, NOAH’s automated approach still offers significant advantages over manual tuning, as evidenced by its competitive results.

8.4.3 Experiments on Few-Shot Learning

8.4.3.1 Datasets

We evaluate NOAH on five fine-grained visual recognition datasets: Food101 [1], OxfordFlowers102 [27], StanfordCars [20], OxfordPets [29], and FGVCAircraft [25]. These datasets cover diverse categories, including food, plants, vehicles, and animals, all closely related to everyday concepts. Following prior studies [31, 46], we evaluate models with 1, 2, 4, 8, and 16 shots to observe performance trends under varying amounts of training data.

8.4.3.2 Results

Figure 8.4 summarizes the results. Our key findings include: (1) In low-data regimes (e.g., 1 or 2 shots), NOAH, LoRA, and Adapter perform similarly, while VPT significantly underperforms. (2) As more shots become available (e.g., 16 shots), NOAH shows clear dominance, with a performance gap of approximately 2% over the runner-up. Analysis of individual datasets reveals that no single prompt module consistently excels, further supporting the argument that search-based approaches like NOAH are more effective than manually engineered solutions.

8.4.4 Experiments on Domain Generalization

8.4.4.1 Datasets

Domain shift is a common challenge in real-world applications [44]. To evaluate domain generalization, we consider two scenarios:

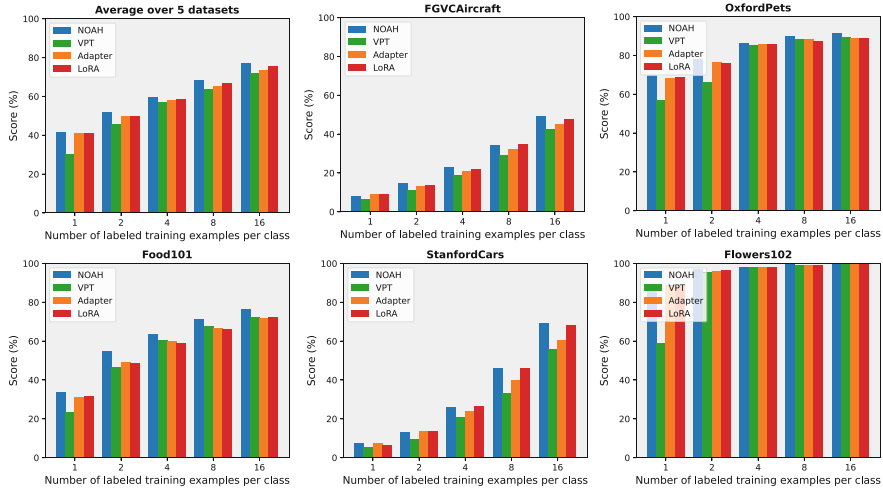


Fig. 8.4 Few-shot learning results on five fine-grained visual recognition datasets. NOAH outperforms the individual modules on average

Table 8.1 Results on image-style generalization.

NOAH demonstrates significant improvements over individual prompt modules across all domain-shifted datasets

	Source	Target			
	ImageNet	-V2	-Sketch	-A	-R
Adapter [16]	70.5	59.1	16.4	5.5	22.1
VPT [18]	70.5	58.0	18.3	4.6	23.2
LoRA [17]	70.8	59.3	20.0	6.9	23.3
NOAH	71.5	66.1	24.8	11.9	28.5

- **Image-style generalization:** Models are trained on ImageNet [6] (16 shots per class) and tested on four domain-shifted datasets: ImageNet-V2 [33], ImageNet-Sketch [36], ImageNet-A [15], and ImageNet-R [14].
- **Image-context generalization:** We use the DOSCO-2k benchmark [48], which includes seven datasets where training and testing data share the same labels but differ in contextual information.

Results are averaged over three random seeds for robustness.

8.4.4.2 Results

Table 8.1 shows that NOAH achieves comparable results to individual modules on the source dataset (ImageNet), with a gap of approximately 1%. However, on the four domain-shifted datasets, NOAH consistently outperforms the baselines, with improvements of 6.8%, 4.8%, 5.0%, and 5.2% on ImageNet-V2, Sketch, A, and R, respectively. These results highlight NOAH’s robustness in handling domain shifts.

As shown in Table 8.2, NOAH also outperforms individual modules in the image-context generalization setting, achieving a 1.6-point improvement over the

Table 8.2 Results on image-context generalization. NOAH surpasses the best individual module by 1.6 points on average

	P-Air	P-Cars	P-Ctech	P-Ins	P-Mam	P-Pets	P-UCF	Average
VPT [18]	24.0	27.1	92.2	63.9	66.1	86.5	68.3	61.2
Adapter [16]	25.6	29.4	93.0	63.2	66.1	86.3	68.5	61.7
LoRA [17]	28.3	34.9	92.7	62.1	63.6	85.4	68.1	62.2
NOAH	28.3	33.6	93.5	64.2	68.7	88.3	69.8	63.8

- Neural Prompt Search (NOAH)

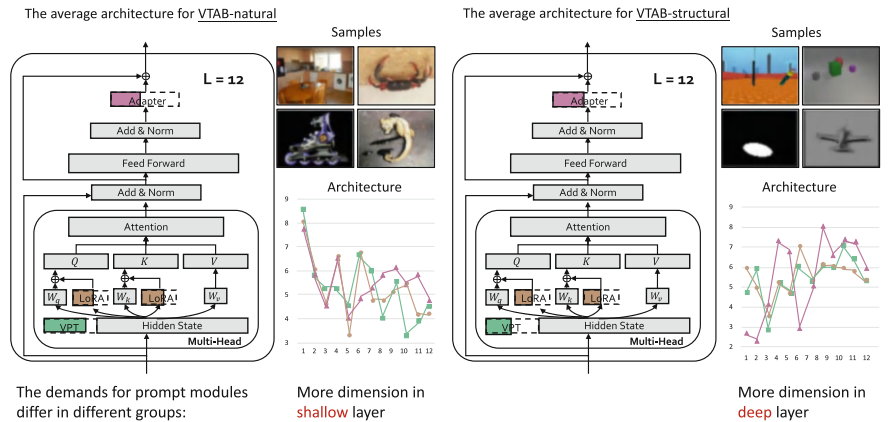


Fig. 8.5 Average subnets (architectures) for the three groups in VTAB-1k. Adapter and LoRA tend to live in deep layers while VPT is found nearly in all depths. The demands for VPT (indicated by the embedding dimension) differ in different groups. The co-existence of the three modules, especially in deep layers, serves as strong evidence of their complementarity, and such a synergy is difficult to obtain by hand-engineering

best-performing baseline. Although LoRA achieves strong results on specific datasets (e.g., P-Air and P-Cars), it performs poorly on others, underscoring the inconsistent performance of individual modules.

8.4.5 Further Analysis

8.4.5.1 Architecture of Subnet

A key question to address is: what does NOAH’s subnet—i.e., the ultimate architecture—look like? To provide a comprehensive answer, we visualize the average architecture—rather than individual ones—discovered for each group in VTAB-1k, as well as the global average across all datasets, in Fig. 8.5. The x-axis represents the network depth, while the y-axis indicates the embedding dimension.

Our analysis reveals several intriguing patterns: (1) **Structured Group Assigns Parameters to Deeper Layers:** For the Structured group, most parameters are allocated to prompt modules in the deeper layers. This phenomenon likely stems from the relationship between the pretrained dataset (ImageNet [10]) and the transferred datasets. Structured datasets, such as SmallNORB/elevation [21] and Clevr/count [19], have distinct label spaces compared to ImageNet. As a result, deeper layers require more tuned parameters to learn task-specific semantic features. (2) **Complementary Roles of Adapter, LoRA, and VPT:** Across all groups (Fig. 8.5a), Adapter and LoRA predominantly appear in the deeper layers, with embedding dimensions larger than four, while their presence diminishes in the shallow layers. On the other hand, VPT is distributed across nearly all layers, with its embedding dimension varying significantly across groups. For example: In the Natural group (Fig. 8.5b), VPT modules are concentrated in the shallow layers. In the Structured group (Fig. 8.5d), VPT modules are more prominent in the deeper layers. (3) **Co-Existence and Synergy:** The co-existence of Adapter, LoRA, and VPT—particularly in the deeper layers—highlights their complementary nature. This synergy underscores the difficulty of achieving an optimal combination through manual design.

In summary, the high variability in module design across groups underscores the efficiency and effectiveness of search-based approaches like NOAH compared to manual engineering for parameter-efficient tuning methods. This adaptability allows NOAH to tailor the architecture to the unique requirements of each dataset group, demonstrating the strength of automated search in addressing diverse tasks.

8.4.5.2 Transferability of Subnet

As discussed previously, the subnet (i.e., architecture) found for different datasets differs dramatically. Here we study whether, or in what circumstances, the subnet found from one dataset can be transferred to another. To this end, we train NOAH on ImageNet and apply the ultimate subnet to the VTAB-1k benchmark where the model is retrained and evaluated. To measure transferability, we compare the ImageNet subnet with the dataset-specific subnets on VTAB-1k. Figure 8.6 shows the comparisons. Overall, the gap between the ImageNet subnet and the 19 dataset-specific subnets on VTAB-1k is below 3%, meaning that NOAH has fair transferability. By digging deeper into the results, we find that the transfer gap is smaller when the source (i.e., ImageNet) and target datasets are closer, and vice versa. For instance, the gaps in the Natural group are less than 1%, which makes sense because the ImageNet images and those from the Natural group share similar visual concepts, such as generic objects, flowers and animals.

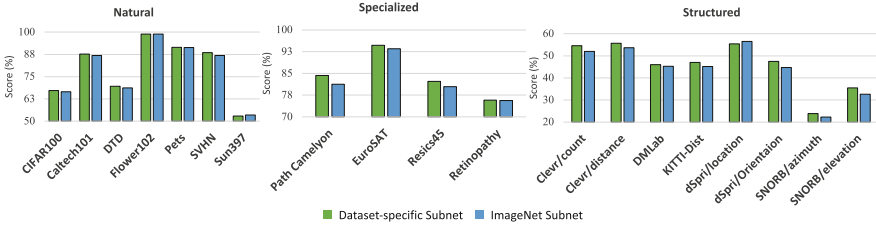


Fig. 8.6 Evaluation on the transferability of subnets. Dataset-specific subnet means the architecture is found from the target dataset. ImageNet subnet means the architecture is found from ImageNet and transferred to the target dataset. All target datasets come from VTAB-1k. In general, better transferability is achieved when the source and target datasets are closer, and vice versa

8.4.5.3 Computational Cost Comparison

In Table 8.3, we provide the GPU hour/dataset to compare the computation cost of different methods and their variants. To gain more insights, we show their performance on three datasets selected from VTAB-Natural, VTAB-Specified, and VTAB-Structural. We observe that (i) compared to the VPT, Adapter, and LoRA, NOAH achieves at least 1.5% average accuracy gain on these three datasets at manageable computational overhead (1.2 vs. 6.3). We would like to emphasize how to accelerate NOAH further is worthwhile future work. (ii) NOAH achieves the best performance (89.9) with the lowest storage cost (0.5), compared to individual prompt modules with similar computational costs to NOAH. Specifically, we study how individual prompts perform under a similar computational cost to NOAH. We conduct a grid search on the Adapter, LoRA, following the grid search strategy in VPT [18], obtaining Adapter* and LoRA*. Under a similar computational budget (7.2 vs. 6.3), the performance of Adapter* and LoRA* still lags behind NOAH by 0.5% with 3 and 6 times more parameters than NOAH. It demonstrates the superiority of the neural prompt search over manual search.

8.4.5.4 Random Search vs. Evolutionary Search

Following Table 8.3, we show the comparison of two searching strategy performances on three datasets selected from VTAB-Natural/Specified/Structural. Table 8.4 shows that evolutionary search, as employed by NOAH, outperforms random search by 1.2% points, demonstrating the effectiveness of our approach.

8.4.5.5 With vs. Without Retraining

Thanks to the weight entanglement strategy in AutoFormer [3] and the attentive sampling method [37], the subnet extracted from the supernet can be directly

Table 8.3 Computational Cost. We illustrate the computational cost of NOAH and other individual prompt modules. VPT, Adapter*, and LoRA* on each dataset are the best ones over grid search of search space. The computational cost for them is, therefore $1.2 * 6 = 7.2$. Param. of VPT, Adapter*, and LoRA* are the average architecture parameters of their best architecture on each dataset

	Time (GPU hour/dataset)				Param. (M)	Top-1 accuracy			
	Supernet	Search	Retrain	Total		Caltech101	EuroSAT	KITTI	Average
VPT [18]	–	–	–	7.2	0.8	90.8	96.1	72.8	86.6
Adapter [16]	–	–	–	1.2	0.2	90.1	94.9	78.3	87.7
Adapter* [16]	–	–	–	7.2	1.5	91.9	95.7	80.6	89.4
LoRA [17]	–	–	–	1.2	0.3	91.4	95.3	78.5	88.4
LoRA* [17]	–	–	–	7.2	3.0	92.0	95.8	80.5	89.4
NOAH	3.1	2.0	1.2	6.3	0.5	92.7	95.4	81.7	89.9

Table 8.4 Random search vs. NOAH’s evolutionary search. The results show that evolutionary search is an optimal subnet searching strategy

	Caltech101	EuroSAT	KITTI	Average
Random	91.8	95.4	79.0	88.7
Evolution	92.7	95.4	81.7	89.9

Table 8.5 With vs. without retraining NOAH’s subnets. The results show no significant difference between the subnets with and without retraining, indicating that retraining can be safely omitted in practice when computing resources are limited

	Nat.	Spe.	Str.	Average
VPT [18]	78.5	82.4	55.0	72.0
Adapter [16]	79.0	84.1	58.5	73.9
LoRA [17]	79.5	84.6	60.5	74.5
NOAH (<i>inherited</i>)	80.0	84.7	60.3	75.1
NOAH (<i>retrained</i>)	80.2	84.9	61.3	75.5

deployed without retraining. To assess whether this applies to NOAH as well, we compare the subnets with and without retraining on the VTAB-1k benchmark. The results, averaged over each group, are presented in Table 8.5. We observe that NOAH without retraining (denoted as *inherited*) remains competitive, still outperforming VPT and Adapter. These findings suggest that the retraining step can be omitted without a significant performance loss.

8.5 Conclusion

This chapter provides insights into the performance of recently proposed parameter-efficient tuning methods, or prompt modules, in the context of computer vision tasks. The study uncovers a critical challenge, i.e., hand-designing the optimal prompt module for specific downstream datasets is highly complex. The problem is solved from a Neural Architecture Search (NAS) perspective and the method excels in few-shot learning regimes and demonstrates robustness to domain shifts—an issue pervasive in real-world scenerios [44].

This chapter also reveals intriguing insights. Notably, three prompt modules have its distinct architectural design in the subnet, depending on the characteristics of the dataset. Since neural networks often progress from pixel-wise features to high-level semantic feature in deeper layers [41], the results suggest that different prompt modules are best suited for extracting features at varying levels of abstraction. We believe these insights could inspire the development of more advanced prompt modules in the future.

Despite these advantages, the method does have some limitations. Training the supernet introduces additional computational overhead, which increases the overall development cost. Additionally, as indicated by the few-shot learning results, the method requires a sufficient amount of labeled data to fully realize its potential.

References

1. Bossard, L., Guillaumin, M., Gool, L.V.: Food-101—mining discriminative components with random forests. In: European conference on computer vision (ECCV), pp. 446–461. Springer (2014)
2. 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)
3. Chen, M., Peng, H., Fu, J., Ling, H.: Autoformer: Searching transformers for visual recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 12270–12280 (2021)
4. Chen, G., Liu, F., Meng, Z., Liang, S.: Revisiting parameter-efficient tuning: Are we really there yet? *arXiv preprint arXiv:2202.07962* (2022)
5. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. *arXiv preprint arXiv:2205.13535* (2022)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255. Ieee (2009)
7. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314* (2023)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018)
9. Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.M., Chen, W., et al.: Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904* (2022)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020)
11. Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., Qiao, Y.: Clip-adapter: Better vision-language models with feature adapters. *Int. J. Comput. Vision* **132**(2), 581–595 (2024)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
13. He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G.: Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366* (2021)
14. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8340–8349 (2021)
15. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15262–15271 (2021)
16. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for NLP. In: International Conference on Machine Learning (ICML), pp. 2790–2799. PMLR (2019)
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. *arXiv preprint arXiv:2106.09685* (2021)
18. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. *arXiv preprint arXiv:2203.12119* (2022)
19. Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2901–2910 (2017)

20. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561 (2013)
21. LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. II–104. IEEE (2004)
22. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021)
23. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190* (2021)
24. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* (2018)
25. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151* (2013)
26. Mao, Y., Mathias, L., Hou, R., Almahairi, A., Ma, H., Han, J., Yih, W.t., Khabsa, M.: Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv preprint arXiv:2110.07577* (2021)
27. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 1447–1454. IEEE (2006)
28. Pan, J., Lin, Z., Zhu, X., Shao, J., Li, H.: St-adapter: Parameter-efficient image-to-video transfer learning. *Adv. Neural Inf. Process. Syst.* **35**, 26462–26477 (2022)
29. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3498–3505. IEEE (2012)
30. Pham, H., Guan, M., Zoph, B., Le, Q., Dean, J.: Efficient neural architecture search via parameters sharing. In: *International Conference on Machine Learning (ICML)*, pp. 4095–4104. PMLR (2018)
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. In: *International Conference on Machine Learning (ICML)*, pp. 8748–8763. PMLR (2021)
32. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A.: Large-scale evolution of image classifiers. In: *International Conference on Machine Learning (ICML)*, pp. 2902–2911. PMLR (2017)
33. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: *International Conference on Machine Learning (ICML)*, pp. 5389–5400. PMLR (2019)
34. Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S.G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J.T., et al.: A generalist agent. *arXiv preprint arXiv:2205.06175* (2022)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: *Advances in neural information processing systems (NeurIPS)*, 30 (2017)
36. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. In: *Advances in Neural Information Processing Systems (NeurIPS)*, 32 (2019)
37. Wang, D., Li, M., Gong, C., Chandra, V.: Attentivenas: Improving neural architecture search via attentive sampling. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6418–6427 (2021)
38. Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., et al.: Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175* (2023)

39. Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al.: mplug-owl: Modularization empowers large language models with multimodality. arXiv preprint arXiv:2304.14178 (2023)
40. Zaken, E.B., Ravfogel, S., Goldberg, Y.: Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv preprint arXiv:2106.10199 (2021)
41. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision (ECCV), pp. 818–833. Springer (2014)
42. Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., et al.: A large-scale study of representation learning with the visual task adaptation benchmark. arXiv preprint arXiv:1910.04867 (2019)
43. Zhong, Z., Friedman, D., Chen, D.: Factual probing is [mask]: Learning vs. learning to recall. arXiv preprint arXiv:2104.05240 (2021)
44. Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C.: Domain generalization: A survey. arXiv preprint arXiv:2103.02503 (2021)
45. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Learning generalisable omni-scale representations for person re-identification. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 5056–5069 (2021)
46. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *Int. J. Comput. Vision* **130**, 2337–2348 (2022)
47. Zhou, H., Wan, X., Vulić, I., Korhonen, A.: Autopeft: Automatic configuration search for parameter-efficient fine-tuning. arXiv preprint arXiv:2301.12132 (2023)
48. Zhou, K., Zhang, Y., Zang, Y., Yang, J., Loy, C.C., Liu, Z.: On-device domain generalization. In: <https://arxiv.org/abs/2209.07521> (2022)
49. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)
50. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8697–8710 (2018)

Chapter 9

Confidence Calibration in Contrastive Vision-Language Models



Shuoyuan Wang, Kaiyang Zhou, and Hongxin Wei

Abstract Contrastive Vision-Language Models (VLMs) have emerged as powerful tools, excelling in various open-vocabulary tasks such as image recognition, retrieval-augmented task adaptation, and visual chatbots. To better adapt to downstream tasks, various parameter-efficient fine-tuning approaches have been developed by the community, e.g., prompt learning. However, an important issue has received little attention: the confidence calibration problem in zero-shot or fine-tuned VLMs, which can significantly undermine the reliability of these models in downstream applications. This chapter addresses this issue by systematically studying the confidence calibration problem in the context of prompt learning for CLIP. The analysis reveals that existing calibration techniques are inadequate, particularly in open-vocabulary scenarios. This chapter then discusses a simple yet effective approach called Distance-Aware Calibration (DAC), which automatically adjusts the temperature scaling parameter based on the distance between predicted text labels and base classes. The effectiveness of the approach is validated on 7 prompt learning methods across 11 downstream tasks.

Keywords Vision-language model · Confidence calibration · Prompt learning

S. Wang · H. Wei (✉)

Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen, China

e-mail: wangsy2024@mail.sustech.edu.cn; weihx@sustech.edu.cn

K. Zhou

Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, Hong Kong, China

e-mail: kychou@hkbu.edu.hk

9.1 Introduction

Vision-language models (VLMs), such as CLIP [31], have demonstrated remarkable zero-shot performance in open-vocabulary image recognition by utilizing natural language supervision [31]. To enhance VLM performance in downstream tasks, various fine-tuning approaches have been explored, with prompt learning [56] standing out for its parameter efficiency and robustness. In open-vocabulary scenarios, it is essential for VLMs to provide accurate and reliable predictions across both seen (base) and unseen (new) classes. Although prompt learning methods significantly boost accuracy, the issue of reliability remains underexplored, posing challenges to deploying these models in critical applications such as medical diagnosis and autonomous driving.

Previous studies have shown that the pretrained CLIP model is well-calibrated during zero-shot inference [24]. However, fine-tuning for downstream tasks introduces miscalibration, meaning the predicted class probabilities no longer reflect the true likelihood of correctness [28]. Notably, existing research has focused solely on the calibration of fine-tuned CLIP models for base classes, which are optimized during fine-tuning. In this chapter, we extend the discussion to confidence calibration of fine-tuned VLMs in an open-vocabulary setting. Our experiments reveal a distinct calibration issue: fine-tuned VLMs tend to be overconfident on new classes while underconfident on base classes. Furthermore, we observe that state-of-the-art post-hoc calibration methods effectively address miscalibration on base classes but fail to generalize to new classes. This limitation motivates the development of post-hoc calibration approaches tailored for open-vocabulary scenarios.

The fundamental idea behind DAC is to modify the temperature value according to the distance between text embeddings of new classes and those of base classes. This involves computing a textual deviation score for each new class, which quantifies the degree of deviation in the normalized text features. By employing class-wise scaling factors, DAC allocates higher temperature values to new classes that are more distant from the base classes, thereby improving calibration performance in open-vocabulary settings.

We evaluate DAC by integrating it into seven distinct prompt learning methods across 11 diverse downstream datasets. The experimental results consistently show that DAC enhances open-vocabulary calibration for all tested methods. For example, DAC achieves an average Expected Calibration Error (ECE) reduction of 6.84% for CoOp [56] across the 11 datasets, with reductions reaching as high as 16% in certain cases. Similar improvements are also observed for other prompt learning approaches, such as MaPLe [16] and PromptSRC [17]. Additionally, DAC demonstrates its versatility by improving the calibration performance of existing post-hoc methods, including Density-Ratio Calibration [46].

In short, we mainly discuss the following key points in this chapter. We first check the confidence calibration in several contrastive fine-tuned Vision-Language Models (VLMs) and highlight that they frequently exhibit miscalibration. Specifically, we

reveal that VLMs become overconfident in predictions for classes that are distant from base classes after prompt tuning. Moreover, existing post-hoc calibration methods often fail to address this issue in open-vocabulary scenarios. Based on the observation of textual divergence, we utilize Distance-Aware Calibration (DAC) to fix the miscalibration in the base-to-new evaluation. DAC is a straightforward and effective post-hoc calibration technique that adjusts predicted confidence without compromising classification accuracy. Comprehensive experiments across 7 VLM tuning methods and 11 datasets validate the efficacy of DAC.¹

9.2 Background

CLIP is a contrastive vision-language model that assesses the alignment between images and text, enabling zero-shot inference for open-vocabulary classes [31]. Let ϕ and θ denote CLIP’s image and text encoders, respectively. Given an image instance $\mathbf{x} \in \mathbb{R}^d$ and a text label c , the CLIP logit function is formulated as:

$$L_c^{clip}(\mathbf{x}_i) = \tau \cdot \text{sim}(\phi(\mathbf{x}), \psi(\mathbf{t}_c)), \quad (9.1)$$

where ϕ and ψ are the image and text encoders in CLIP. The text embedding $\mathbf{t}_c$ is derived from a handcrafted prompt like “a photo of a class,” where “class” is replaced with the text label c . The scalar τ is generally set to a constant value of 100.

For multi-class classification, the prediction involves selecting the label with the highest probability from the candidate set $C = c_{i=1}^K$, as shown below:

$$c^* = \arg \max_{c \in C} p(c|\mathbf{x}) = \arg \max_{c \in C} \frac{e^{L_c^{clip}(\mathbf{x})}}{\sum_{i=1}^K e^{L_i^{clip}(\mathbf{x})}} \quad (9.2)$$

where $p(c|\mathbf{x})$ represents the predicted probability of class c for the instance $\mathbf{x}$.

Prompt Tuning To strengthen the CLIP model in downstream applications, prompt tuning algorithms are normally designed to optimize the context prompt [55, 56] for high efficiency. In particular, CoOp [56] replace the hand-crafted textual tokens with a set of learnable textual token $\mathcal{T} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_M\}$, where M is the length of tokens. Then, the logit function is:

$$L_c^{\text{tuned}}(\mathbf{x}) = \tau \cdot \text{sim}(\phi(\mathbf{x}), \psi(\mathbf{t}'_c)), \quad (9.3)$$

where $\mathbf{t}'_c = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \dots, \mathbf{v}_M, \mathbf{c}]$ and $\mathbf{c}$ denotes the textual embedding of class c . During tuning, the learnable textual tokens $\mathcal{T}$ are optimized to minimize the cross-

¹ The codebase has been released at https://github.com/ml-stat-Sustech/CLIP_Calibration.

entropy loss ℓ_{ce} with labeled few-shot samples $\mathcal{D} = \{(\mathbf{x}_i, c_i)\}_{i=1}^N$. Formally, the optimization of CoOp is:

$$\mathcal{T}^* = \arg \min_{\mathcal{T}} \frac{1}{N} \sum_{i=1}^N \ell_{ce}(p(c_i | \mathbf{x}_i))$$

To preserve generalization on unseen classes within the same task—namely, open-vocabulary classes—CoCoOp [55] utilizes instance-conditional prompts given by $\mathbf{t}_m(\mathbf{x}) = \mathbf{t}_m + h_{\theta}(\mathbf{x})$, where $h_{\theta}(\cdot)$ is a neural network parameterized by θ and $m \in \{1, 2, \dots, M\}$.

While these tuning methods demonstrate promising performance, the reliability of the predicted probability $p(c|\mathbf{x})$ remains uncertain for tuned VLMs. Consequently, we proceed to introduce calibration in the prompt tuning context.

Confidence Calibration In multi-class classification tasks, vision-language models are designed to produce class probabilities $p(c|\mathbf{x})$, as defined in Eq. 9.2. Beyond achieving high accuracy, these models are expected to be well-calibrated, meaning that the predicted probabilities should reliably reflect the true likelihood of correctness. Formally, a *perfectly calibrated* model satisfies the condition $Pr(c^* = c \mid p(c|\mathbf{x}) = q) = q, \forall q \in [0, 1]$.

To measure the extent of miscalibration, the *Expected Calibration Error* (ECE) is commonly used. ECE quantifies the average discrepancy between accuracy and confidence. Given N samples divided into K bins $\{b_1, b_2, \dots, b_K\}$, ECE is computed as:

$$\text{ECE} = \sum_{k=1}^K \frac{|b_k|}{N} |\text{acc}(b_k) - \text{conf}(b_k)|, \quad (9.4)$$

where $\text{acc}(\cdot)$ and $\text{conf}(\cdot)$ represent the average accuracy and confidence within bin b_k , respectively.

9.3 An Empirical Study

In the literature, it has been shown that pretrained CLIP is well-calibrated in zero-shot inference [24]. In this section, we empirically investigate the calibration performance of tuned vision-language models (VLMs). Specifically, we compare the zero-shot CLIP [31] with a variety of popular prompt tuning algorithms, including: CoOp [56], CoCoOp [55], ProDA [22], KgCoOp [47], MaPLe [16], ProGrad [57] and PromptSRC [17].

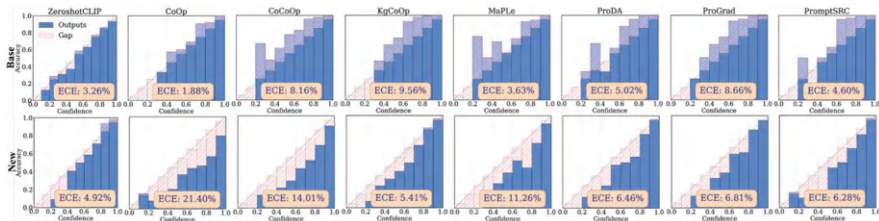


Fig. 9.1 Reliability analysis of fine-tuned CLIP (ViT-B/16) on the Flower102 dataset, where Expected Calibration Error (ECE) is used as the metric (lower values indicate better calibration). Miscalibration is visualized using pink to represent overconfidence and grey to represent underconfidence. The results indicate that CLIP exhibits underconfidence on base classes while displaying a tendency toward overconfidence on new classes

9.3.1 Are Fine-Tuned VLMs Well-Calibrated?

To illustrate the miscalibration issue in fine-tuned VLMs, we fine-tuned the pretrained CLIP model using seven tuning methods across 11 downstream datasets and evaluated calibration performance using Expected Calibration Error (ECE) as the metric. Details of the methods and datasets are provided in Sect. 9.5.1. Figure 9.1 visualizes the miscalibration of fine-tuned CLIP on the Flower102 dataset using reliability diagrams. Intuitively, one might expect that fine-tuned deep models would exhibit higher confidence on learned base classes and adopt more conservative predictions for new classes in open-vocabulary settings, as these are not explicitly learned during fine-tuning. However, contrary to this expectation, we observe a surprising phenomenon: *fine-tuned VLMs tend to be underconfident on base classes and overconfident on new classes*.

9.3.2 Can Fine-Tuned VLMs Be Calibrated?

To address the miscalibration observed in fine-tuned CLIP, we employ commonly-used post-hoc confidence calibration methods with the goal of improving calibration performance on both base and new classes. Post-hoc calibration methods are broadly categorized into two types: *scaling-based methods* and *binning-based methods*. For scaling-based methods, we utilize Temperature Scaling (TS) [9] and Density-Ratio Calibration (DEN) [46]. For binning-based methods, we evaluate Histogram Binning [50], Isotonic Regression (IR) [51], and Multi-Isotonic Regression (MIR) [53]. The calibration results on the ImageNet-1k dataset are summarized in Table 9.1. Based on these results, we can make the following key observations:

Post-hoc Calibration Effectively Mitigates Base Class Miscalibration In line with former studies [28, 37], we find that both scaling-based techniques (e.g., TS, DEN) and binning-based strategies (e.g., MIR) substantially enhance calibration on base classes, effectively resolving the miscalibration issue in this context.

Table 9.1 The ECE (%) of fine-tuned CLIP with various calibration methods is presented. Using ProDA, we fine-tune CLIP-ViT-B/16 on ImageNet-1k. “ZS” refers to zero-shot CLIP, and “Conf” represents the confidence score without calibration post-tuning. “–” indicates that results are not applicable. “Conf” demonstrates underconfidence for base classes, while “TS” and “DEN” display overconfidence for new classes

	ZS	Conf	TS	DEN	HB	IR	MIR
Base	3.58	4.82	1.94	0.73	4.23	2.09	0.82
New	2.09	1.59	3.90	3.86	–	–	–

Post-hoc Calibration Does Not Generalize to New Classes For example, the calibration performance of methods like “TS” and “DEN” is worse than the uncalibrated model (“Conf”) or even the zero-shot model (“ZS”) when applied to new classes. This outcome indicates that simply applying a calibrator trained on base classes is inadequate for handling broader unseen distributions [48]. Additionally, binning-based methods require probability inputs from base classes, making them incompatible with zero-shot predictions. Such a limitation raises an important question: *How can we effectively calibrate fine-tuned VLMs for open-vocabulary classes?*

9.4 Open-Vocabulary Calibration

In this section, we first explore open-vocabulary calibration by visualizing the joint vision-language embedding space, highlighting the textual gap between base and new classes as a crucial factor influencing calibration. Next, we quantify this gap using a distance-based metric to evaluate the extent of deviation in new classes. Finally, we introduce the method and underscore its advantages in enhancing open-vocabulary calibration.

9.4.1 Feature Space Analysis

In the prompt tuning of CLIP, the primary effects of tuning are predominantly observed in the textual features. This insight motivates us to examine how modifications in text features contribute to the miscalibration observed in new classes. To better understand open-vocabulary miscalibration in vision-language models (VLMs) after prompt tuning, we first analyze it within the joint vision-language representation space. Specifically, inspired by the concept of *modality gap* [20], paired image-text samples from downstream datasets are processed using both zero-shot and tuned CLIP. Their embeddings are then projected into a 2D space using singular value decomposition (SVD) for visualization.

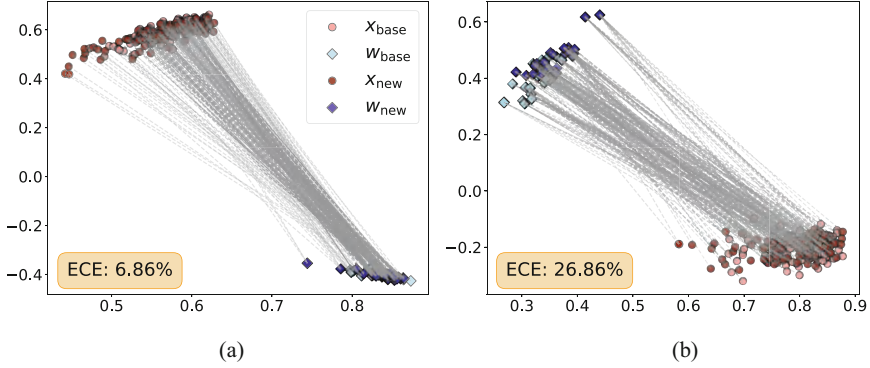


Fig. 9.2 Paired inputs from image (x) and text (w) are sampled from the DTD dataset and processed through zero-shot and tuned CLIP models. The resulting representations are visualized using singular value decomposition (SVD). Compared to zero-shot CLIP, CoOp shows a larger textual distribution gap between base and new classes. (a) Base. (b) New

Consistent with the findings in [20], we observe that images and text are embedded at a moderate distance from each other in the shared representation space. In zero-shot CLIP, embeddings within the same modality are relatively well-clustered. However, as shown in Fig. 9.2b, prompt tuning introduces a pronounced textual distribution gap between base and new classes. Furthermore, the Expected Calibration Error (ECE) increases significantly, highlighting the model’s miscalibration. Based on these observations, we informally hypothesize that *the extent of deviation in the textual gap is a critical factor influencing open-vocabulary calibration in CLIP*.

9.4.2 Motivation

To verify the assumption, we first introduce a distance-based metric, referred to as *proximity*, to quantify the degree of deviation in features.

Definition 9.1 (Feature Proximity [42]) Consider a feature $\mathbf{z} \in \mathbb{R}^d$ representing the embedding of a test sample and a set of held-out feature embeddings $\mathcal{E} \in \mathbb{R}^{n \times d}$. Proximity is defined as a function that inversely correlates with the mean distance between the test sample and its K nearest neighbors in the held-out set:

$$P(\mathbf{z}, \mathcal{E}) = \sigma \left(\frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{N}_K(\mathbf{z}, \mathcal{E})} \text{dist}(\mathbf{z}, \mathbf{x}_i) \right), \quad (9.5)$$

where $\mathcal{N}_K(\mathbf{z}, \mathcal{E})$ denotes the set of K nearest neighbors of $\mathbf{z}$, and $\text{dist}(\cdot, \cdot)$ is a predefined metric for distance calculation. The function σ is a non-negative,

monotonically decreasing function. This definition implies that as the proximity of a test sample increases, the likelihood that the sample originates from the same distribution as the held-out set also increases. In this context, proximity can be employed to approximate the textual distribution gap in VLMs.

In VLMs, we have a set of ℓ_2 -normalized [31] textual embeddings for seen classes, denoted as $\mathcal{W} = \{\mathbf{w}_j\}_{j=1}^S$. Given an arbitrary class $c_i \in \mathcal{U}$ at test time, with its corresponding class token c_i , we first compute its normalized textual embedding $\mathbf{w}_i = \theta(\mathbf{t}_i)$, where $\mathbf{t}_i = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \dots, \mathbf{v}_M, c_i\}$ represents the textual token sequence. The textual distribution gap between this class c_i and the seen classes can then be defined as follows:

$$P(\mathbf{w}_i, \mathcal{W}) = \exp \left(-\frac{1}{K} \sum_{\mathbf{w}_j \in \mathcal{N}_K(\mathbf{w}_i, \mathcal{W})} \|\mathbf{w}_i - \mathbf{w}_j\|_2 \right). \quad (9.6)$$

Here we use e^{-x} as $\sigma(\cdot)$ and ℓ_2 -distance for $\text{dist}(\cdot, \cdot)$.

We illustrate the correlation between the deviation degree of textual features and calibration in Fig. 9.3. For each textual embedding of the new classes, its proximity to the held-out set of base textual features is calculated. The observations are as follows:

Lower Proximity is Associated with Higher Confidence and ECE Specifically, when a textual feature is distant from the base textual distribution, the tuned VLM tends to exhibit higher confidence, resulting in poorer calibration performance. Similar to the pretraining stage [20], the contrastive loss used during vision-

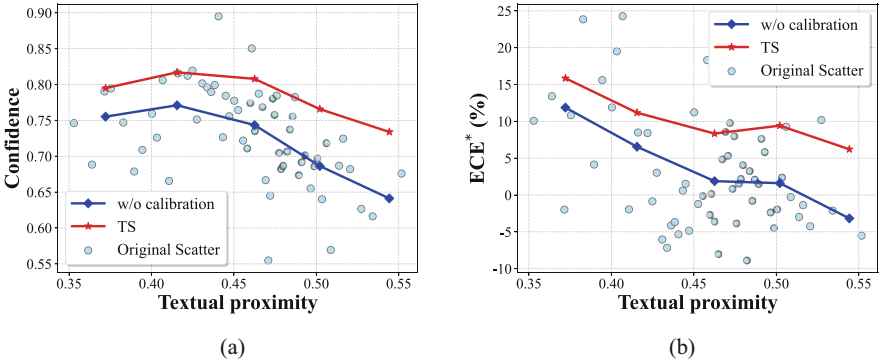


Fig. 9.3 Class-wise performance on the new classes of the StanfordCars dataset after tuning is illustrated. ECE* represents ECE (bin=1), where a positive value indicates overconfidence and a negative value indicates underconfidence. The scatter points represent the original results, while the broken line corresponds to the bin-based results. Both confidence and ECE* increase as proximity decreases, highlighting the correlation between lower proximity and miscalibration. Notably, Temperature Scaling (TS) fails to mitigate the overconfidence issue. (a) Confidence. (b) ECE*

language tuning establishes a connection between visual and textual modalities, embedding each modality into a relatively compact distribution. However, during inference, if a class deviates significantly from the textual distribution learned during tuning, the tuned CLIP model may fail to align well across modalities, leading to improper confidence predictions. Moreover, as shown in Fig. 9.3b, widely-used calibration techniques, such as Temperature Scaling (TS), not only fail to address this issue but may, in some cases, worsen it. These observations motivate us to adjust the confidence level when predictions exhibit lower textual proximity.

9.4.3 Distance-Aware Calibration

Recall the analysis above, predictions for classes with lower textual proximity are associated with higher confidence and poorer calibration performance. To address this issue, we introduce a post-hoc method to adjust the predicted confidence levels. Specifically, we introduce a textual deviation-based score to scale the temperature value, thereby updating the confidence levels to better align with the true correctness likelihood.

Textual Deviation Estimation Ideally, the model should preserve higher uncertain predictions for examples from new classes, which are expected to exhibit relatively low accuracy. Guided by this principle, we define the textual deviation score by calculating the deviation distances of new classes from base classes using fine-tuned CLIP models and comparing these to the corresponding distances in the pretrained CLIP model.

Let $\mathbf{w}_i$ and $\mathbf{w}'_i$ represent the normalized text features of class c_i from the pretrained and tuned VLMs, respectively. The **Textual Deviation (TD)** score for class c_i can be formulated as:

$$\gamma(c_i) = \frac{P(\mathbf{w}'_i, \mathcal{W}')}{P(\mathbf{w}_i, \mathcal{W})}, \quad (9.7)$$

where $\mathcal{W}$ and $\mathcal{W}'$ represent the sets of normalized textual features for base classes in the original and fine-tuned VLMs, respectively. For example, a low TD score implies that the associated class is more likely to diverge from the textual distribution of base classes. As shown in Sect. 9.4.2, fine-tuned VLMs often exhibit overconfidence in their predictions for such classes. Hence, we suggest adjusting the confidence level based on the TD score for each class.

Calibrated Inference Before evaluation, we first collect the TD scores $\Gamma = \{\gamma_i\}_{i=1}^K$ for each class. Given a test instance $\mathbf{x}$, we first calculate its predicted class $\hat{c} = \arg \max_c p(c | \mathbf{x})$ using the tuned CLIP model. Finally, the rectified logit is calculated as :

$$L_c^{dac}(\mathbf{x}) = \gamma(\hat{c}) \cdot \tau \cdot \text{sim}(\phi(\mathbf{x}), \psi(\mathbf{t}'_c)), \quad (9.8)$$

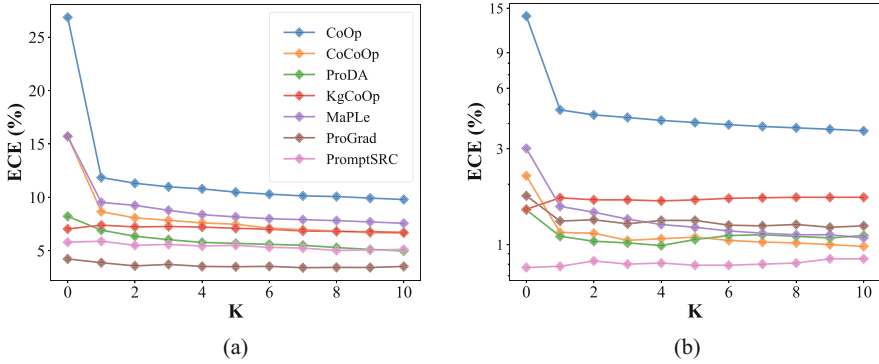


Fig. 9.4 Hyperparameter sensitivity analysis for the number of neighbors (K) used in computing textual proximity. The results indicate that miscalibration is significantly mitigated when $K > 1$. (a) DTD. (b) SUN397

where $\mathbf{t}'_c$ represents the textual token of class c , as defined in Eq. 9.3. Here, we adjust the confidence level of the prediction by modulating the temperature value τ using the TD score $\gamma(\hat{c})$. For base classes, we assign $\gamma_{\hat{c}} = 1$, reducing our DAC to standard temperature scaling. Consequently, our method does not alter the calibration performance for base classes. Notably, DAC offers several benefits:

- **Realistic:** The calculation of DAC relies exclusively on information from text labels. As a result, our method does not require access to visual data from open-vocabulary concepts, which is often challenging to obtain in real-world applications.
- **Algorithm-agnostic:** DAC can be effortlessly integrated with existing prompt tuning algorithms, consistently improving calibration performance on new classes (see Figs. 9.2 and 9.3).
- **Easy to use:** Our method eliminates the need for hyperparameter tuning (see Fig. 9.4) and introduces no additional computational overhead (see Table 9.5).

9.5 Experiments

9.5.1 Experimental Setup

Benchmark Setting We assess the confidence calibration performance of widely used prompt tuning approaches. According to the few-shot evaluation protocol [55], datasets are partitioned into base and new classes. The model is trained solely on the base classes under a few-shot setup, and calibration performance is typically measured on the new classes.

Datasets In this section, we adopt the standard base-to-new setting [55, 56] and utilize 11 diverse image recognition datasets. These include general object datasets such as ImageNet [6] and Caltech101 [7], as well as fine-grained classification datasets like OxfordPets [30], StanfordCars [18], Flowers102 [26], Food101 [2], and FGVCAircraft [23]. For specialized categories, the datasets include SUN397 [45] for scene recognition, UCF101 [34] for action recognition, DTD [5] for texture classification, and EuroSAT [10] for satellite imagery classification.

Compared Methods We compare and integrate our method with existing VLM tuning algorithms, which can be categorized into two groups: *prompt tuning* and *multimodal tuning*. For prompt tuning, we evaluate Context Optimization (CoOp) [56], Conditional Context Optimization (CoCoOp) [55], Prompt Distribution Learning (ProDA) [22], Knowledge-guided Context Optimization (KgCoOp) [47], and Prompt-aligned Gradient (ProGrad) [57]. For multimodal tuning, we include Multimodal Prompt Learning (MaPLE) [16] and Prompting with Self-regulating Constraints (PromptSRC) [17].

Evaluation Metrics We use four standard metrics to evaluate open-vocabulary confidence calibration: Expected Calibration Error (ECE) [9], Maximum Calibration Error (MCE) [9], Adaptive Calibration Error (ACE) [27], and Proximity-Informed Expected Calibration Error [46].

9.5.2 Results

DAC Improves Open-Vocabulary Calibration in Prompt Tuning Table 9.2 presents the detailed calibration error on the new classes across all datasets. For open-vocabulary calibration performance, DAC effectively mitigates miscalibration across various tuning methods and all metrics, as shown in Table 9.3. For instance, DAC significantly reduces the ECE from 6.29 to 4.82%, thereby enhancing the reliability of CoCoOp in terms of predicted confidence. Moreover, DAC is also applicable to *multimodal tuning* methods, such as MaPLE and PromptSRC, which adjust both the vision and language branches. DAC consistently improves the open-vocabulary calibration performance of these methods. These results underscore that the textual distribution gap is a widespread issue across various tuning approaches. DAC effectively addresses this challenge, consistently boosting open-vocabulary calibration performance when applied to existing tuning techniques.

DAC Significantly Reduces Calibration Error, Especially for High-Confidence Predictions To identify where DAC provides the most improvement over baseline methods, we analyzed calibration performance across various confidence levels. Specifically, we measure the gap between confidence and accuracy for each bin on the DTD dataset, as shown in Table 9.4. A negative gap indicates underconfident predictions, while a positive gap indicates overconfident predictions. For ease of comparison, the difference between the baseline and our method is also presented

Table 9.2 Calibration performance on new classes of 11 datasets. Calibration error is given by $\times 10^{-2}$. **Bold** numbers are significantly better results

Methods	Caltech101	Pets	Cars	Flowers	Food101	FGVC	SUN397	DTD	EuroSAT	UCF101	ImageNet	AVG
ZeroshotCLIP	1.60	3.42	3.31	4.91	1.83	6.55	3.48	6.86	9.12	5.52	2.09	4.43
CoOp	4.08	1.83	12.50	18.34	3.83	28.44	13.70	26.86	12.71	19.24	10.69	13.84
CoOp+DAC	3.17	1.82	5.16	10.19	1.78	17.38	4.05	10.48	8.62	8.67	5.67	7.00
CoCoOp	3.89	2.19	2.31	11.49	1.70	11.26	2.20	15.71	12.32	4.48	1.65	6.29
CoCoOp+DAC	3.65	2.43	2.21	7.74	1.64	9.03	1.09	7.47	13.49	2.70	1.61	4.82
ProDA	4.61	6.60	3.02	6.67	1.51	5.36	1.49	8.22	4.84	3.11	1.59	4.27
ProDA+DAC	4.87	4.72	3.28	6.32	0.70	7.40	1.06	5.68	3.33	4.14	2.42	3.99
KgCoOp	1.96	3.16	2.81	5.99	1.98	12.53	1.28	7.03	6.82	2.64	1.80	4.36
KgCoOp+DAC	1.84	3.11	3.12	5.90	1.94	11.78	1.67	7.09	6.59	2.69	1.83	4.32
MaPLe	1.33	2.51	2.02	13.43	1.26	10.38	3.01	15.71	9.06	2.38	2.39	5.77
MaPLe+DAC	1.26	2.51	2.75	11.28	1.50	9.06	1.22	8.16	8.55	2.30	2.11	4.61
ProGrad	1.91	2.98	2.88	5.45	2.04	13.18	1.75	4.22	7.40	2.85	1.81	4.22
ProGrad+DAC	1.97	3.31	2.29	5.04	1.85	10.46	1.32	3.49	6.90	2.42	2.11	3.74
PromptSRC	1.62	3.08	2.16	5.77	1.60	9.79	0.77	5.79	7.35	2.64	1.71	3.84
PromptSRC+DAC	1.58	2.98	2.39	5.03	1.55	8.55	0.79	5.50	7.24	2.46	1.84	3.63

Table 9.3 Average ECE (%) across 11 datasets. “Vanilla” refers to the original performance on open-vocabulary classes using existing tuning methods, while “DAC” represents our method applied to these tuning methods. A downward arrow ($\downarrow$) indicates that smaller values are better. Calibration error is reported in units of $\times 10^{-2}$. **Bold** values indicate significantly superior results

Metric	Method	CoOp	CoCoOp	ProDA	KgCoOp	MaPLe	ProGrad	PromptSRC
ACE ($\downarrow$)	Vanilla	13.76	6.21	4.35	4.43	5.71	4.27	3.92
	DAC	6.91	4.77	4.08	4.38	4.64	3.74	3.69
MCE ($\downarrow$)	Vanilla	3.80	1.79	1.27	1.18	1.82	1.22	1.09
	DAC	1.71	1.40	1.32	1.13	1.42	1.09	1.08
PIECE ($\downarrow$)	Vanilla	14.71	8.07	6.57	6.67	7.59	6.75	6.26
	DAC	9.02	7.15	6.35	6.63	6.98	6.55	6.17

Table 9.4 Calibration results of ECE (%) across different confidence levels on new classes. Δ shows the improvement achieved by DAC. **Bold** numbers denote the top-3 most significant improvements

Method	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
CoOp	0.00	19.24	18.86	15.87	20.42	33.28	30.30	37.84	41.60	18.57
+DAC	0.00	4.95	8.17	11.33	6.51	11.42	24.12	17.41	11.37	−0.94
Δ	0.00	−14.29	−10.69	−4.54	−13.91	−21.86	−6.18	−20.43	−30.23	−19.51
MaPLe	0.00	0.00	−12.80	3.90	16.73	10.50	38.07	23.93	19.11	12.13
+DAC	0.00	−3.62	−15.32	5.72	6.74	3.12	15.45	6.16	9.29	6.55
Δ	0.00	−3.62	−2.52	1.82	−10.99	−7.38	−22.62	−17.77	−9.82	−5.58
ProGrad	0.00	−3.82	0.14	−0.10	4.29	6.31	3.48	8.11	1.23	4.86
+DAC	0.00	−0.71	0.03	1.30	−1.32	0.40	−0.65	−0.04	0.44	−0.34
Δ	0.00	3.11	−0.11	1.40	−5.61	−5.91	−4.13	−8.15	−0.79	−5.20

for each bin. The results demonstrate that DAC primarily enhances calibration performance in high-confidence predictions (confidence bins 0.6–0.9). In particular, DAC significantly mitigates the overconfidence issue. For instance, in the case of CoOp, DAC reduces the gap from 41.60 to 11.37 in the confidence bin 0.9, achieving a direct improvement of 30.23.

DAC Is Robust to the Number of Nearest Neighbors We evaluate the impact of K (the number of nearest neighbors) on the open-vocabulary calibration performance of DAC. Specifically, we conduct this ablation study using ECE on the DTD and SUN397 datasets, varying the number of textual neighbors $K \in \{1, 2, 3, \dots, 10\}$. As shown in Fig. 9.4, increasing the number of neighbors beyond 0 results in a noticeable reduction in ECE, with performance gradually saturating as more neighbors are added. Notably, DAC demonstrates robustness to the choice of K , with even $K = 1$ delivering significant calibration improvements. This observation suggests that adjusting the confidence level based on the textual distribution gap between base and new classes is highly beneficial. For simplicity, we use a moderate value of $K = 5$ throughout our experiments.

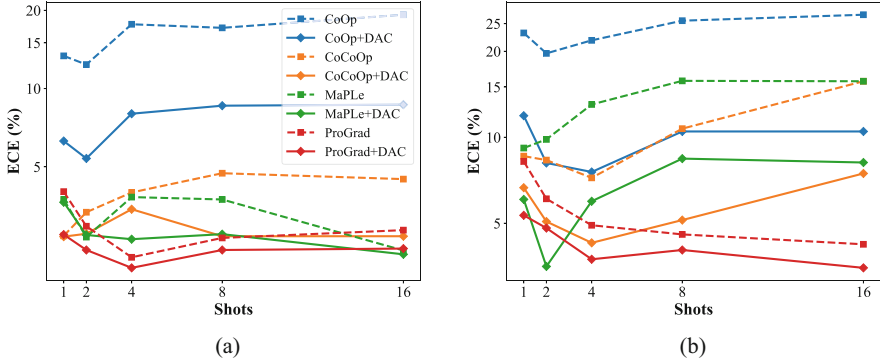


Fig. 9.5 Comparison of ECE (%) results using different shots. Miscalibration remains a common issue, but DAC effectively reduces it across varying numbers of shots. The Y-axis is presented in an exponential scale. (a) UCF101. (b) DTD

DAC Is Effective Across Different Training Instances We analyze the performance of DAC in few-shot learning scenarios, where a “shot” refers to the number of labeled samples per class used during tuning. Unlike accuracy, which generally improves as the number of shots increases [56], calibration errors persist across a wide range of few-shot settings. Figure 9.5 visualizes the ECE performance across different numbers of shots, demonstrating that DAC effectively addresses calibration issues, regardless of the number of labeled samples per class.

9.6 Discussion

Visual Proximity vs. Textual Proximity Proximity is a crucial factor in reliable machine learning [13, 21, 46, 49]. Former research has predominantly emphasized proximity estimation of model inputs, which has substantially mitigated miscalibration in supervised learning. Nonetheless, we show that textual proximity holds a more significant influence than visual proximity in the context of prompt tuning. To validate this, we compare the calibration performance and inference efficiency of DAC against Density-Ratio Calibration [46] on the ImageNet-1k dataset. The findings are summarized in Table 9.5. Regarding calibration accuracy, DAC outperforms DEN in calibrating new classes, with even better results achieved when DAC is combined with DEN. Furthermore, we assess inference efficiency. In terms of computational cost, DAC retains high efficiency with minimal extra computational overhead compared to DEN and the baseline (Conf). In summary, DAC demonstrates exceptional calibration accuracy and inference efficiency for new classes, surpassing methods dependent on image proximity.

Textual Feature Normalization Is Critical In DAC, we standardize the features generated by the textual encoder when computing TD scores. However, the neces-

Table 9.5 Calibration performance and inference time on ImageNet-1k. CoOp is employed for prompt tuning. “base” refers to the calibration performance on base classes. DAC demonstrates both superior calibration performance and inference efficiency for new classes

	Conf	DAC	DEN	DAC+DEN
ECE_{base}	1.65	1.65	1.19	1.19
ECE_{new}	10.69	5.67	8.08	3.29
time (s)	0.12	0.13	15.88	15.91

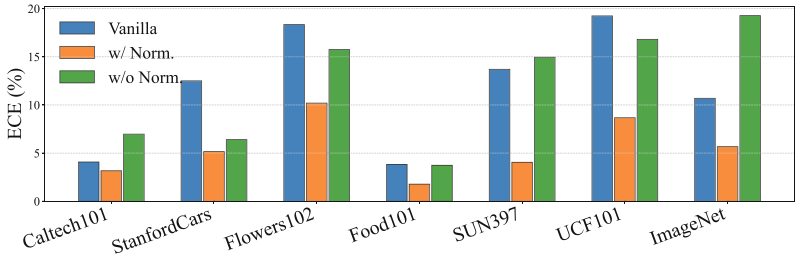


Fig. 9.6 ECE (%) of textual feature normalization with CoOp. We compare the effect of using normalization in the textual feature vs. without normalization

sity of this standardization is not immediately apparent. To investigate its influence, we perform an ablation study comparing the effectiveness of textual proximity-based calibration with and without feature normalization. As an example, we present the calibration results of tuned CLIP with CoOp in this study. Our findings, illustrated in Fig. 9.6, reveal that normalized features enable a more accurate estimation of class-wise proximity, leading to improved calibration performance compared to raw features. Previous studies [14, 35, 46] have highlighted that feature norms can introduce misestimation in K-nearest-neighbor-based scores when Euclidean distance is used. Empirically, we confirm that normalization is a critical factor in the effectiveness of our distance-based calibration method.

9.7 Related Work

Prompt Tuning in Contrastive Vision-Language Models Leveraging rich natural language supervision, pretrained large vision-language models (VLMs) [12, 31] have demonstrated strong capabilities in understanding visual concepts and applying them to downstream tasks such as image classification [22, 25, 31, 56], image captioning [11, 40, 41], and visual question answering [29], often in a zero-shot setting. While VLMs excel in generalizing to novel visual concepts, their zero-shot performance often falls short when compared to fine-tuned models on specific downstream tasks [54]. To address this gap, parameter-efficient tuning methods, such as prompt tuning [47, 52, 55–57], have been proposed to enhance downstream adaptation while maintaining efficiency. Moreover, multimodal tuning

approaches [16, 17, 33] have been introduced, focusing on jointly adapting the vision and language branches of models like CLIP to improve modality alignment. Despite the significant progress in vision-language tuning, its safety-related evaluations, such as calibration performance, remain underexplored. In this chapter, we address this gap by analyzing the calibration performance of fine-tuned VLMs. Our findings indicate that fine-tuned CLIP models frequently suffer from miscalibration. We discuss the effectiveness of Distance-Aware Calibration (DAC), which mitigates this issue by adjusting the logit scale based on textual information.

Confidence Calibration Confidence calibration has been extensively explored to ensure that the confidence scores produced by models accurately represent their true reliability. A prevalent strategy involves applying post-hoc calibration techniques after model training. These methods are generally divided into two categories: *scaling-based* methods [9, 15, 36, 46, 48, 53] and *bin-based* methods [32, 50, 51, 53]. For scaling-based calibration, a well-known approach is temperature scaling [9], which learns a single scalar to adjust the scale of the logits. Mix-n-Match [53] integrates ensemble and compositional techniques to improve data efficiency while preserving accurate confidence estimates. ATS [15] refines predicted confidence by introducing a per-sample adaptive temperature. For bin-based calibration, samples are grouped into bins based on their confidence values, with each bin calibrated individually. Moreover, research on calibration has been extended to vision-language models (VLMs) [19, 28]. This chapter centers on open-vocabulary calibration for fine-tuned VLMs. We first illustrate that existing post-hoc calibration techniques for base classes do not generalize well to new classes. To overcome this shortcoming, DAC modifies the logit scale of predictions using a textual deviation-based score specifically tailored for new classes.

9.8 Conclusion

In this chapter, we discuss the challenge of confidence calibration in contrastive vision-language models. we highlight the role of textual features in calibration and the corresponding method named Distance-Aware Calibration (DAC), a straightforward yet effective enhancement to temperature scaling for fine-tuned CLIP, aimed at improving open-vocabulary calibration in combination with existing prompt tuning techniques. DAC modifies the temperature value by leveraging the distance between the predicted text label and the base classes, allowing fine-tuned CLIP to generate more reliable confidence estimates for new classes. Through extensive experiments, we show that DAC not only improves calibration performance on new classes following prompt tuning but also maintains the original performance on base classes. Additionally, DAC integrates easily with existing scaling-based calibration approaches without introducing any additional inference latency.

Further Reading

To get a better knowledge, interested readers may refer to the following material:

- Wang [39] for a comprehensive survey on calibration in deep learning.
- Xiong et al. [46], Yuksekgonul et al. [49] for distance-based methods in confidence calibration.
- Zhang et al. [54] for a comprehensive survey of vision-language models on vision tasks.
- Oh et al. [28] for calibration in CLIP under various covariate shifts.
- Wang et al. [43] for mitigating miscalibration during the fine-tuning phase of CLIP.

Acknowledgments This study was funded by the Shenzhen Fundamental Research Program (Grant No. JCYJ20230807091809020).

References

1. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
2. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101—mining discriminative components with random forests. In: European Conference on Computer Vision, pp. 446–461 (2014)
3. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1721–1730 (2015)
4. Cho, E., Kim, J., Kim, H.J.: Distribution-aware prompt tuning for vision-language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 22004–22013 (2023)
5. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3606–3613 (2014)
6. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
7. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: 2004 Conference on Computer Vision and Pattern Recognition workshop, pp. 178–178 (2004)
8. Goyal, S., Kumar, A., Garg, S., Kolter, Z., Raghunathan, A.: Finetune like you pretrain: Improved finetuning of zero-shot vision models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19338–19347 (2023)
9. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning, pp. 1321–1330, PMLR (2017)
10. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **12**(7), 2217–2226 (2019)

11. Hu, X., Gan, Z., Wang, J., Yang, Z., Liu, Z., Lu, Y., Wang, L.: Scaling up vision-language pre-training for image captioning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17980–17989 (2022)
12. Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: *International Conference on Machine Learning*, pp. 4904–4916, PMLR (2021)
13. Jiang, H., Kim, B., Guan, M., Gupta, M.: To trust or not to trust a classifier. In: *Advances in Neural Information Processing Systems*, 31 (2018)
14. Jiang, W., Cheng, H., Chen, M., Wang, C., Wei, H.: DOS: Diverse outlier sampling for out-of-distribution detection. In: *The Twelfth International Conference on Learning Representations* (2024)
15. Joy, T., Pinto, F., Lim, S.-N., Torr, P.H.S., Dokania, P.K.: Sample-dependent adaptive temperature scaling for improved calibration. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12), pp. 14919–14926 (2023)
16. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: MaPLe: Multi-modal prompt learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19113–19122 (2023)
17. Khattak, M.U., Wasim, S.T., Naseer, M., Khan, S., Yang, M.-H., Khan, F.S.: Self-regulating prompts: Foundational model adaptation without forgetting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15190–15200 (2023)
18. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561 (2013)
19. LeVine, W., Pikus, B., Raj, P., Gil, F.A.: Enabling calibration in the zero-shot inference of large vision-language models. In: *ICLR workshop on Pitfalls of limited data and computation for Trustworthy ML* (2023)
20. 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. *Adv. Neural Inf. Process. Syst.* **35**, 17612–17625 (2022)
21. Liu, J.Z., Padhy, S., Ren, J., Lin, Z., Wen, Y., Jerfel, G., Nado, Z., Snoek, J., Tran, D., Lakshminarayanan, B.: A simple approach to improve single-model deep uncertainty via distance-awareness. *J. Mach. Learn. Res.* **24**(42), 1–63 (2023)
22. Lu, Y., Liu, J., Zhang, Y., Liu, Y., Tian, X.: Prompt distribution learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5206–5215 (2022)
23. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151* (2013)
24. Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., Lucic, M.: Revisiting the calibration of modern neural networks. *Adv. Neural Inf. Process. Syst.* **34**, 15682–15694 (2021)
25. Naeem, M.F., Khan, M.G.Z.A., Xian, Y., Afzal, M.Z., Stricker, D., Van Gool, L., Tombari, F.: I2MVFormer: Large language model generated multi-view document supervision for zero-shot image classification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15169–15179 (2023)
26. Nilsback, M.-E., Zisserman, A.: Automated flower classification over a large number of classes. In: *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729 (2008)
27. Nixon, J., Dusenberry, M.W., Zhang, L., Jerfel, G., Tran, D.: Measuring Calibration in Deep Learning. In: *Computer Vision and Pattern Recognition workshops*, 2 (2019)
28. Oh, C., Lim, H., Kim, M., Han, D., Yun, S., Choo, J., Hauptmann, A. G., Cheng, Z.-Q., Song, K.: Towards calibrated robust fine-tuning of vision-language models. In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems* (2024)
29. Parelli, M., Delitzas, A., Hars, N., Vlassis, G., Anagnostidis, S., Bachmann, G., Hofmann, T.: CLIP-guided vision-language pre-training for question answering in 3D scenes. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.

- 5606–5611 (2023)
30. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3498–3505 (2012)
 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. In: International Conference on Machine Learning, pp. 8748–8763, PMLR (2021)
 32. Roelofs, R., Cain, N., Shlens, J., Mozer, M.C.: Mitigating bias in calibration error estimation. In: International Conference on Artificial Intelligence and Statistics, pp. 4036–4054, PMLR (2022)
 33. Shen, S., Yang, S., Zhang, T., Zhai, B., Gonzalez, J.E., Keutzer, K., Darrell, T.: Multitask vision-language prompt tuning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 5656–5667 (2024)
 34. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
 35. Sun, Y., Ming, Y., Zhu, X., Li, Y.: Out-of-distribution detection with deep nearest neighbors. In: International Conference on Machine Learning, pp. 20827–20840, PMLR
 36. Tomani, C., Cremers, D., Buettner, F.: Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In: European Conference on Computer Vision, pp. 555–569 (2022)
 37. Tu, W., Deng, W., Gedeon, T.: A closer look at the robustness of contrastive language-image pre-training (CLIP). In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
 38. Udandaraao, V., Gupta, A., Albanie, S.: Sus-x: Training-free name-only transfer of vision-language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2725–2736 (2023)
 39. Wang, C.: Calibration in deep learning: A survey of the state-of-the-art. arXiv preprint arXiv:2308.01222 (2023)
 40. Wang, N., Xie, J., Wu, J., Jia, M., Li, L.: Controllable image captioning via prompting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 2, pp. 2617–2625 (2023)
 41. Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O.K., Singhal, S., Som, S., et al.: Image as a foreign language: BEiT pretraining for vision and vision-language tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19175–19186 (2023)
 42. Wang, S., Wang, J., Wang, G., Zhang, B., Zhou, K., Wei, H.: Open-vocabulary calibration for fine-tuned CLIP. In: Forty-first International Conference on Machine Learning (2024)
 43. Wang, S., Li, Y., Wei, H.: Understanding and Mitigating Miscalibration in Prompt Tuning for Vision-Language Models. arXiv preprint arXiv:2410.02681 (2024)
 44. Wei, H., Xie, R., Cheng, H., Feng, L., An, B., Li, Y.: Mitigating neural network overconfidence with logit normalization. In: International Conference on Machine Learning, pp. 23631–23644, PMLR (2022)
 45. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3485–3492 (2010)
 46. Xiong, M., Deng, A., Koh, P.W., Wu, J., Li, S., Xu, J., Hooi, B.: Proximity-informed calibration for deep neural networks. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
 47. Yao, H., Zhang, R., Xu, C.: Visual-language prompt tuning with knowledge-guided context optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6757–6767 (2023)
 48. Yu, Y., Bates, S., Ma, Y., Jordan, M.: Robust calibration with multi-domain temperature scaling. Adv. Neural Inf. Process. Syst. **35**, 27510–27523 (2022)

49. Yuksekgonul, M., Zhang, L., Zou, J., Guestrin, C.: Beyond confidence: Reliable models should also consider atypicality. In: *Thirty-seventh Conference on Neural Information Processing Systems* (2023)
50. Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: *International Conference on Machine Learning*, vol. 1, pp. 609–616 (2001)
51. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 694–699 (2002)
52. Zang, Y., Li, W., Zhou, K., Huang, C., Loy, C.C.: Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225* (2022)
53. Zhang, J., Kailkhura, B., Han, T.Y.-J.: Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In: *International Conference on Machine Learning*, pp. 11117–11128, PMLR (2020)
54. Zhang, J., Huang, J., Jin, S., Lu, S.: Vision-language models for vision tasks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(8), 5625–5644 (2024)
55. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825 (2022)
56. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *Int. J. Comput. Vision* **130**(9), 2337–2348 (2022)
57. Zhu, B., Niu, Y., Han, Y., Wu, Y., Zhang, H.: Prompt-aligned gradient for prompt tuning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15659–15669 (2023)

Part III
Applying Intelligence: Real-World
Applications of Vision-Language Models

Chapter 10

Open-Vocabulary Object Detection Based on Detection Transformers



Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy

Abstract Open-vocabulary object detection involves detecting and recognizing objects from an unbounded set of categories. Ideally, an open-vocabulary detector should be extendable to produce bounding boxes based on user input, whether it is a natural language instruction or an exemplar image. This enhances flexibility and improves the user experience in human-computer interaction. In this chapter, we discuss *OV-DETR*, an open-vocabulary detector built upon the Detection Transformer (DETR) architecture. Once trained, OV-DETR can detect any object provided its class name or an exemplar image. A primary challenge in adapting DETR for open-vocabulary detection lies in the inability to compute the classification cost matrix for novel classes without access to their labeled images. We overcome this challenge by reformulating the learning objective as a binary matching task between input queries and their corresponding objects. This strategy enables the model to learn robust correspondences that generalize effectively to unseen queries at test time. During training, we condition the Transformer decoder on input embeddings derived from a pretrained vision-language model such as CLIP. This allows for matching both text and image queries. Through extensive experiments on LVIS and COCO datasets, we demonstrate that OV-DETR, the first end-to-end Transformer-based open-vocabulary detector, significantly outperforms existing baselines.

Y. Zang (✉)
Shanghai AI Laboratory, Shanghai, China
e-mail: zangyuhang@pjlab.org.cn

W. Li · C. C. Loy
S-Lab, Nanyang Technological University, Singapore, Singapore
e-mail: wei.l@ntu.edu.sg; ccloy@ntu.edu.sg

K. Zhou
CS Department, Hong Kong Baptist University, Hong Kong, China
e-mail: kzyzhou@hkbu.edu.hk

C. Huang
Apple Inc., Cupertino, CA, USA
e-mail: chen-huang@apple.com

Keywords Open-vocabulary detection · Detection · Transformer · Image-text alignment · Conditional matching

10.1 Introduction

Object detection is a computer vision technique that aims to identify and locate objects within an image [7], video [16], or point cloud [11]. Traditional object detection algorithms typically exhibit limited scalability with respect to the number of object categories. They are often constrained to a predefined set of classes, as defined by popular detection datasets such as Pascal VOC [7] and COCO [21]. For instance, a detector trained on COCO can only reliably identify the 80 object classes included in the dataset (e.g., ‘person’, ‘cat’, ‘dog’), rendering it incapable of detecting objects from categories not encountered during training.

A common approach to detecting novel classes is to expand the training dataset with images of those classes and then retrain the detection model. However, this method is often impractical and inefficient due to the high costs associated with data collection and model retraining. In the detection literature, the challenge of generalizing from known (base) classes to unseen (novel) classes is often addressed as a zero-shot detection problem [1]. These approaches use zero-shot learning techniques, such as word embedding projections [8], to enable the model to recognize novel classes without requiring any labeled training examples.

Following the popularity of vision-language models, open-vocabulary detection has garnered significant attention within the research community [13, 41]. A core principle of existing open-vocabulary detection methods is to align the visual features extracted by the object detector with textual embeddings generated by models trained on massive image-text datasets, such as CLIP [27] (see Fig. 10.1a). This alignment enables the classifier to recognize novel object classes solely based on their textual descriptions, without the need for any corresponding training images.

A significant limitation of current open-vocabulary detectors [13, 41] is their reliance on region proposals, which often fail to accurately cover all instances of novel classes within an image due to the absence of training data for these classes (see Fig. 10.1a). This issue has also been highlighted by recent research [17], which suggests that the binary nature of the Region Proposal Network (RPN) can predispose the model to overfit to the seen classes, hindering its ability to generalize effectively to novel classes.

In this chapter, we discuss training an end-to-end open-vocabulary detector, dubbed OV-DETR, built upon the DETR framework [2]. Our primary objective is to enhance the model’s ability to generalize to novel object classes without relying on an intermediate RPN. OV-DETR is designed to detect any object given either its class name or an exemplar image. This unique capability offers greater flexibility compared to traditional open-vocabulary detection methods, which typically rely solely on natural language descriptions.

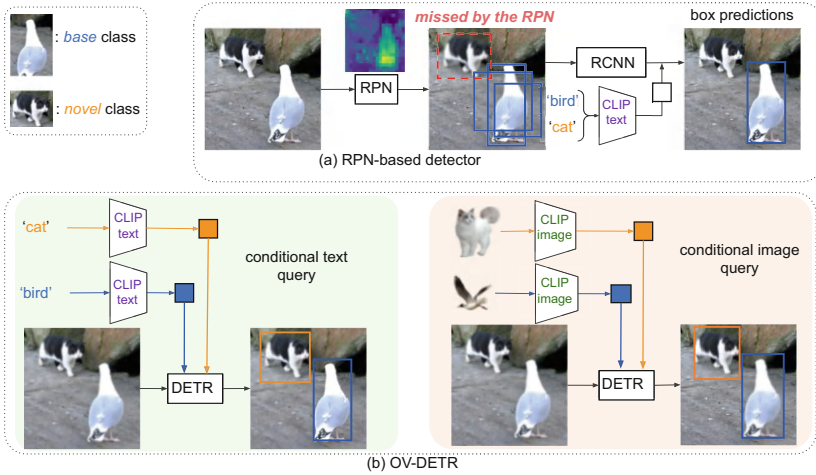


Fig. 10.1 Comparison between a RPN-based detector and our Open-Vocabulary Transformer-based detector (OV-DETR) using conditional queries. The RPN trained on closed-set object classes is easy to ignore novel classes (e.g., the “cat” region receives little response). Hence the cats in this example are largely missed with few to no proposals. By contrast, our OV-DETR is trained to perform matching between a conditional query and its corresponding box, which helps to learn correspondence that can generalize to queries from unseen classes. Note we can take input queries in the form of either text (class name) or exemplar images, which offers greater flexibility for open-vocabulary object detection (Image sourced from [39])

Despite the elegance of end-to-end DETR training, adapting it for open-vocabulary detection presents significant challenges. Primarily, computing classification costs for novel classes becomes intractable due to the absence of their training labels. To address this, we reformulate the learning objective as a binary matching problem between input queries (class names or exemplar images) and corresponding objects within the image. This matching loss, trained on diverse query-object pairs, fosters the learning of robust correspondences that generalize effectively to unseen queries during inference. To facilitate this, we extend the DETR decoder to incorporate conditional input queries. Specifically, we condition the decoder on query embeddings derived from a pretrained vision-language model like CLIP [27], enabling conditional matching for both text and image queries. Figure 10.1 illustrates this high-level concept, demonstrating superior performance in detecting novel classes compared to RPN-based closed-set detectors.

The main findings are derived from experiments on two challenging open-vocabulary object detection datasets, and show consistent performance improvements. Specifically, OV-DETR achieves a novel class mask mAP of 17.4 on the open-vocabulary LVIS dataset [13] and a novel class box mAP of 29.4 on the open-vocabulary COCO dataset [41], surpassing baseline methods by 1.3 and 1.8 mAP, respectively.

10.2 Related Work

10.2.1 Open-Vocabulary Object Detection

Open-vocabulary object detection focuses on detecting *novel* classes that were not encountered during the training phase, thus serving as a crucial benchmark for evaluating generalization capabilities. As shown in Fig. 10.2a, the dataset is divided into two distinct sets: *base* and *novel* classes. The detector model undergoes training on bounding box annotations of *base* classes, supplemented with extra image-level image-text pairs. Subsequently, the detector is evaluated on its ability to detect objects belonging to the *novel* classes.

Open-vocabulary object detectors commonly use the advancements in large pretrained vision-language models, such as those explored in [10, 13, 24, 41, 44], to effectively integrate open-vocabulary information within the object detection framework. OVR-CNN [41] initially pretrains a Faster R-CNN detector [29] on image-caption pairs using BERT [6]. Subsequently, the model undergoes fine-tuning on downstream detection datasets. ViLD [13] employs a distillation-based approach that aligns the image feature extractor of Mask R-CNN [15] with the image and text encoder of CLIP [27]. This alignment enables CLIP to synthesize classification weights for any novel class effectively. Detic [44] uses image-level labels from the ImageNet dataset [5] as weak supervision to train the open-vocabulary detector. OWL-ViT [24] directly trains a Vision Transformer (ViT)-based detector on a massive corpus of image-text pairs. In this chapter, we present OV-DETR that differs from previous works in that we train a Transformer-based detector end-to-end, with a conditional matching framework.

Zero-Shot object detection is a closely related task to Open-Vocabulary detection, both of which focus on the challenging problem of detecting novel object classes [1, 20, 28, 31, 45]. However, the zero-shot setting presents a more challenging and less practical scenario due to the stringent limitation on accessing any resources related to the unseen classes [41], as shown in Fig. 10.2b. A common strategy for zero-shot detection involves utilizing pretrained word embeddings, such as GloVe [26], as classifier weights [1]. Furthermore, research has demonstrated that leveraging external resources, such as textual descriptions, can significantly enhance

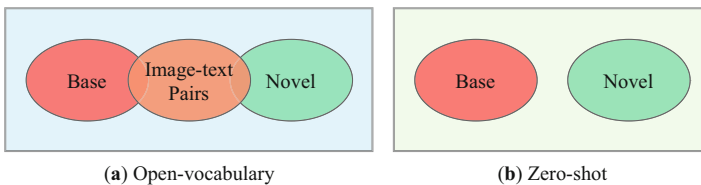


Fig. 10.2 The difference between (a) open-vocabulary object detection and (b) zero-shot object detection. Extra image-text pairs are allowed to be used in the open-vocabulary setting (Image sourced from [38])

the generalization capabilities of these classifier embeddings [20, 28]. Alternatively, Zhao et al. [31] proposed utilizing Generative Adversarial Networks (GANs) [12] to generate feature representations for novel classes. In contrast, Zhu et al. [45] explored a data augmentation strategy to synthesize representations of unseen classes.

10.2.2 *Detection Transformer*

For the transformer-based object detection models, the pioneering DETR approach [2] revolutionized transformer-based object detection by framing the detection task as a set-to-set matching problem, significantly simplifying the traditional detection pipeline. Subsequent research has focused on enhancing DETR’s performance and training efficiency. Deformable DETR [46] introduced a deformable attention module, which selectively samples sparse pixel locations for attention computation, effectively mitigating the slow convergence issue observed in the original DETR. Moreover, Deformable DETR incorporates a multi-scale feature extraction mechanism to further improve performance. SMCA [9] accelerates training convergence by employing a location-aware co-attention mechanism. Conditional DETR [23] also addresses slow convergence by introducing conditional spatial queries derived from reference points and decoder embeddings. DINO [42] uses denoising training strategies to significantly accelerate the training process. Beyond training efficiency, recent advancements have explored hybrid mechanisms [3, 25, 47] that integrate both IoU-based and Hungarian matching for improved label assignment and overall performance.

10.3 OV-DETR

OV-DETR aims to develop a concise and effective open-vocabulary object detector capable of detecting objects described by arbitrary textual inputs or exemplified by images. We use the success of DETR [2], which elegantly reformulates object detection as an end-to-end set matching problem for closed-set scenarios, thereby eliminating the need for manually designed components such as anchor generation and non-maximum suppression. This streamlined pipeline provides a compelling foundation for constructing our end-to-end open-vocabulary object detection system.

Adapting a standard DETR, designed for closed-set matching, to the open-vocabulary setting, where matching against unseen classes is required, presents a significant challenge. A straightforward approach involves training a class-agnostic module, such as in ViLD [13]), to handle all classes. However, this approach remains inadequate for detecting open-vocabulary classes with no associated labeled images. We introduce a novel perspective on the matching process within DETR to address

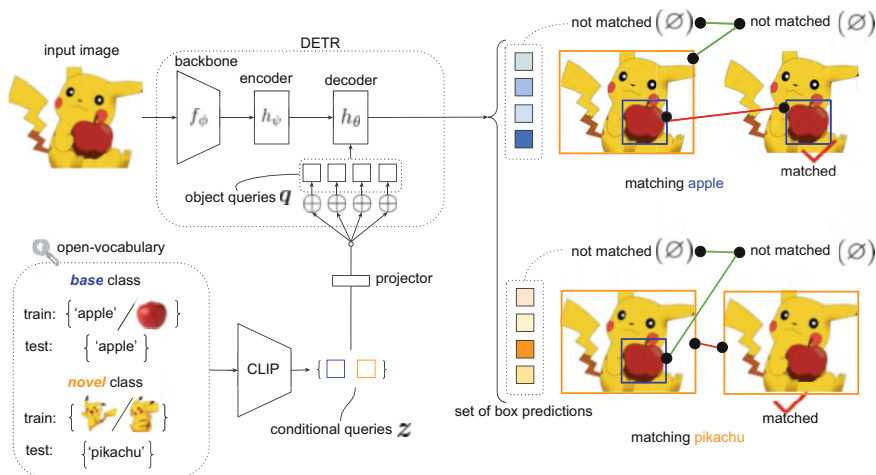


Fig. 10.3 Overview of OV-DETR. Unlike the standard DETR, our method does not separate ‘objects’ from ‘non-objects’ for a closed set of classes. Instead, OV-DETR performs open-vocabulary detection by measuring the matchability (‘matched’ vs. ‘not matched’) between some conditional inputs (text or exemplar image embeddings from CLIP) and detection results. We show such pipeline is flexible to detect open-vocabulary classes with arbitrary text or image inputs (Image sourced from [39])

this limitation. We reformulate the conventional fixed set-matching objective into a conditional binary matching framework. This reformulation enables matching between conditional inputs, such as textual descriptions or exemplar images, and the predicted object detection outputs Fig. 10.3 ([39]).

10.3.1 Revisiting Closed-Set Matching in DETR

Given an input image x , DETR predicts a set of N object predictions $\hat{y}$, where N is fixed and determined by the number of learnable object queries q that serve as learnable positional encodings. DETR operates in two main steps: (i) set prediction, and (ii) optimal bipartite matching.

Set Prediction For an input image x , global context representations c are obtained by first extracting features using a CNN backbone f_ϕ and then processing these features with a Transformer encoder h_ψ :

$$c = h_\psi(f_\phi(x)), \quad (10.1)$$

where the c corresponds to a sequence of feature embeddings of q . Given context features c and object queries q , the Transformer decoder h_θ generates the set

prediction $\hat{\mathbf{y}} = \{\hat{\mathbf{y}}_i\}_{i=1}^N$:

$$\hat{\mathbf{y}} = h_\theta(\mathbf{c}, \mathbf{q}), \quad (10.2)$$

where $\hat{\mathbf{y}}$ comprises both bounding box predictions $\hat{\mathbf{b}}$ and class predictions $\hat{\mathbf{p}}$ for a closed-set of training classes.

Optimal Bipartite Matching aims to find the optimal one-to-one correspondence between the set of N predictions $\hat{\mathbf{y}}$ and the set of ground truth objects $\mathbf{y} = \{\mathbf{y}_i\}_{i=1}^M$, which may also include “no object” ($\emptyset$). This involves finding a permutation of N elements $\sigma \in \mathfrak{S}_N$ predictions that minimizes the overall matching cost:

$$\hat{\sigma} = \arg \min_{\sigma \in \mathfrak{S}_N} \sum_i^N \mathcal{L}_{\text{cost}}(\mathbf{y}_i, \hat{\mathbf{y}}_{\sigma(i)}), \quad (10.3)$$

where $\mathcal{L}_{\text{cost}}(\mathbf{y}_i, \hat{\mathbf{y}}_{\sigma(i)})$ represents the pairwise *matching cost* between the ground truth object $\mathbf{y}_i$ and the prediction $\hat{\mathbf{y}}_{\sigma(i)}$ with index $\sigma(i)$. Note $\mathcal{L}_{\text{cost}}$ is comprised of the losses for both class prediction $\mathcal{L}_{\text{cls}}(\hat{\mathbf{p}}, \mathbf{p})$ and bounding box localization $\mathcal{L}_{\text{box}}(\hat{\mathbf{b}}, \mathbf{b})$. Bipartite matching establishes one-to-one correspondences, assigning each prediction $\hat{\mathbf{y}}_i$ to a single ground-truth annotation $\mathbf{y}_j$ or no object ($\emptyset$). The optimal assignment can be efficiently determined using the Hungarian algorithm [18].

Challenge As mentioned above, bipartite matching methods are not directly applicable to open-vocabulary scenarios. This limitation stems from the requirement of label information to compute matching costs, as evident in Eq. (10.3). Since novel classes lack training labels, the model’s predictions for object queries cannot generalize to them. Consequently, bipartite matching can only be applied to base classes with available training labels, as shown in Fig. 10.4a.

10.3.2 Conditional Matching for Open-Vocabulary Detection

To extend DETR to open-vocabulary detection, we enhance the Transformer decoder with conditional inputs and reformulate the learning objective as a binary matching problem.

Conditional Inputs Given an object detection dataset with standard annotations for all training (*base*) classes, we must convert these annotations into suitable conditional inputs to support our new training paradigm. Specifically, for each ground-truth annotation with bounding box $\mathbf{b}_i$ and class label name $\mathbf{y}_i^{\text{class}}$, we

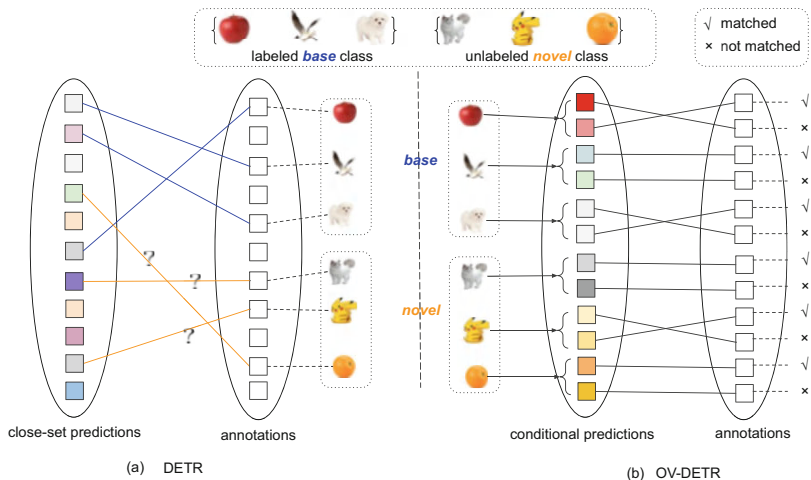


Fig. 10.4 Comparing the label assignment mechanisms of DETR and our OV-DETR. (a) For DETR, the classification cost can only be computed for labeled base classes. (b) On the contrary, our OV-DETR performs conditional matching for any object specified by a conditional input, which is *class-agnostic* and can generalize to novel classes (Image sourced from [39])

employ the CLIP model [27] to generate corresponding image and text embeddings: z_i^{image} and z_i^{text} :

$$\begin{aligned} z_i^{\text{image}} &= \text{CLIP}_{\text{image}}(\mathbf{x}, \mathbf{b}_i), \\ z_i^{\text{text}} &= \text{CLIP}_{\text{text}}(\mathbf{y}_i^{\text{class}}). \end{aligned} \quad (10.4)$$

Given the strong alignment of image and text embeddings in CLIP, we can utilize either as input queries to condition the DETR decoder. This allows us to train the model to effectively match objects based on these learned representations. After training, we can use arbitrary input queries to perform open-vocabulary detection. To ensure balanced training across both image and text query modalities, we randomly select z_i^{text} or z_i^{image} with probability ξ as the conditional input during training. Moreover, we follow previous works [13, 37] to generate additional object proposals for *novel* classes to enrich our training data. For novel-class proposals, we exclusively use image embeddings (z_i^{image}) as conditional inputs, as their class names are unavailable for generating text embeddings.

Conditional Matching Our core training objective is to optimize the matchability between conditional input embeddings and the corresponding detection results. To enable conditional matching, we first employ a fully-connected layer ($\mathbb{F}_{\text{proj}}$) to

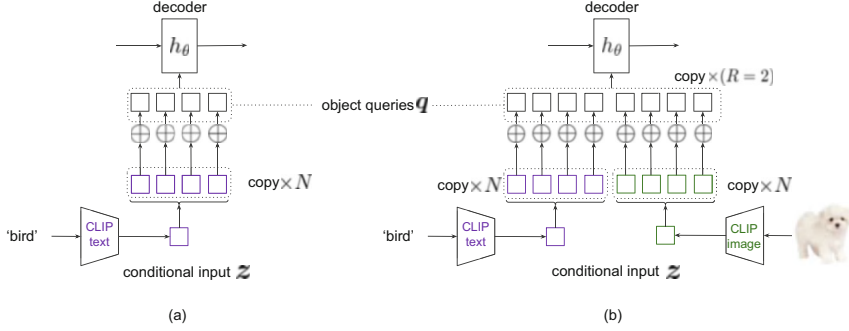


Fig. 10.5 DETR decoder with (a) single conditional input or (b) multiple conditional inputs in parallel (Image sourced from [39])

project the conditional input embeddings (z_i^{text} or z_i^{image}) into the same feature space as the object queries (q). Then the input to the DETR decoder q' is given by:

$$q' = q \oplus \mathbb{F}_{\text{proj}}(z_i^{\text{mod}}), \quad \text{mod} \in \{\text{text}, \text{image}\}, \quad (10.5)$$

where we incorporate the projected conditional input embeddings ($\mathbb{F}_{\text{proj}}(z_i^{\text{mod}})$) into the class-agnostic object queries (q) using a simple addition operation ($\oplus$) to obtain class-specific queries (q').

Constraining the addition of conditional input embeddings (z) to a single object query severely limits the model's ability to detect multiple instances of the same or different object classes within an image. This limitation arises because typical object detection datasets contain multiple instances of various object classes per image. To enrich the training signal for conditional matching, we replicate the object queries (q) R times and the conditional inputs (z_i^{text} or z_i^{image}) N times before applying the conditioning operation in Eq. (10.5). Consequently, we generate a total of $N \times R$ queries for matching during each forward pass, as shown in Fig. 10.5b. To maintain independence between different query copies during the final conditioning process, we incorporate an attention mask, similar to the approach employed in [4].

With conditioned query features q' , our binary matching loss for label assignment is defined as:

$$\mathcal{L}_{\text{cost}}(y, \hat{y}_\sigma) = \mathcal{L}_{\text{match}}(p, \hat{p}_\sigma) + \mathcal{L}_{\text{box}}(b, \hat{b}_\sigma), \quad (10.6)$$

where $\mathcal{L}_{\text{match}}(p, \hat{p}_\sigma)$ represents a novel matching loss that replaces the classification loss $\mathcal{L}_{\text{cls}}(p, \hat{p}_\sigma)$ in Eq. (10.3). In our implementation, p is a 2-dimensional probability vector representing the matchability (i.e., 'matched' vs. 'not matched'). The matching loss, $\mathcal{L}_{\text{match}}$, is implemented using a Binary Cross-Entropy (BCE) loss ($\mathcal{L}_{\text{BCE}}$) between the predicted probabilities ($\hat{p}_\sigma$) and the ground-truth probabilities (p). As shown in Fig. 10.4b, when the query is 'bird', the matching loss should

facilitate the identification of all bird instances within the image while correctly classifying instances of other classes as ‘not matched’.

10.3.3 Optimization

By optimizing Eq. (10.6), we determine the optimal label assignments (σ) for each object query. This process yields a set of detected objects, each characterized by predicted bounding box coordinates ($\hat{\mathbf{b}}$) and a 2-dimensional matching probability ($\hat{\mathbf{p}}$). These predictions are then used to compute the final training loss. To further enhance the model’s learning, we incorporate an embedding reconstruction head. This component learns to predict an embedding ($\mathbf{e}$) that can effectively reconstruct the corresponding conditional input embedding ($\mathbf{z}^{\text{text}}$ or $\mathbf{z}^{\text{image}}$):

$$\mathcal{L}_{\text{embed}}(\mathbf{e}, \mathbf{z}) = \left\| \mathbf{e} - \mathbf{z}^{\text{mod}} \right\|_1, \quad \text{mod} \in \{\text{text}, \text{image}\}. \quad (10.7)$$

The intuition behind embedding reconstruction is that it encourages the model to learn distinct representations for different concepts in feature space by reconstructing the various conditional input embeddings ($\mathbf{z}^{\text{text}}$ or $\mathbf{z}^{\text{image}}$). This, in turn, provides more discriminative guidance for the conditional training process.

The final training loss, $\mathcal{L}_{\text{loss}}$, is a combination of embedding reconstruction loss ($\mathcal{L}_{\text{embed}}$) with matching loss ($\mathcal{L}_{\text{match}}(\mathbf{p}, \hat{\mathbf{p}})$) and the bounding box loss ($\mathcal{L}_{\text{box}}(\mathbf{b}, \hat{\mathbf{b}})$):

$$\begin{aligned} \mathcal{L}_{\text{loss}}(\mathbf{y}, \hat{\mathbf{y}}) &= \mathcal{L}_{\text{match}}(\mathbf{p}, \hat{\mathbf{p}}) + \mathcal{L}_{\text{box}}(\mathbf{b}, \hat{\mathbf{b}}) + \mathcal{L}_{\text{embed}}(\mathbf{e}, \mathbf{z}) \\ &= \lambda_{\text{LBCE}} \mathcal{L}_{\text{BCE}} + \lambda_{\text{LL1}} \mathcal{L}_{\text{L1}} + \lambda_{\text{LGIOU}} \mathcal{L}_{\text{GIOU}} + \lambda_{\text{Lembed}} \mathcal{L}_{\text{embed}}, \end{aligned} \quad (10.8)$$

where $\mathcal{L}_{\text{box}}$ consists of the L1 loss and the generalized IoU (GIOU) [30] loss for box regression. The λ_{LBCE} , λ_{LL1} , λ_{LGIOU} and λ_{Lembed} are the weighting parameters.

10.3.4 Inference

During inference, for each image, we feed the text embeddings ($\mathbf{z}^{\text{text}}$) of all base and novel classes into the model. Subsequently, we merge the detection results by selecting the top- k predictions with the highest scores. Following the approach in [13], we set $k = 100$ for the COCO dataset and $k = 300$ for the LVIS dataset. To obtain the context representation ($\mathbf{c}$) in Eq. (10.1), we process the input image through the CNN backbone f_ϕ followed by the Transformer encoder h_ψ . Notably, $\mathbf{c}$ is computed only once and shared across all conditional inputs for computational efficiency. Subsequently, the conditioned object queries from different classes are processed in parallel by the Transformer decoder. To enhance the model’s capacity, we replicate the object queries R times, as illustrated in Fig. 10.5b.

10.4 Experiments

Datasets We evaluate OV-DETR on two standard open-vocabulary detection benchmarks derived from LVIS [14] and COCO [21] respectively. LVIS [14] contains 100K images with 1,203 classes. The classes are divided into three groups, namely frequent, common and rare, based on the number of training images. Following ViLD [13], we categorize these classes into frequent, common, and rare based on their image frequency. Subsequently, we designate the 337 rare classes as novel and train the model exclusively on the frequent and common classes. Following OVR-CNN [41], resulting in 48 base categories and 17 novel categories. Notably, we exclude 15 categories lacking a synset in the WordNet hierarchy. The training set encompasses the entire COCO dataset, with the restriction that only images containing at least one base class are utilized. We refer to these modified datasets as OV-LVIS and OV-COCO, respectively.

Evaluation Metrics For OV-LVIS, we evaluate performance using mask mAP, with separate scores reported for rare, common, and frequent classes, denoted as AP_r^m , AP_c^m and AP_f^m . The rare classes are considered *novel* classes, and their corresponding mAP is denoted as (AP_{novel}^m) . We also report the overall mask mAP AP^m across all the classes. For OV-COCO, we follow previous work that only reports the $AP50^b$ metric, which means the box mAP at IoU threshold 0.5.

Extension for Instance Segmentation For OV-LVIS, instance segmentation results are required for evaluation. While DETR [2] and its successors [23, 46] were initially designed for object detection, they can be readily extended to perform instance segmentation. Following the approach in DETR, we incorporate an external, class-agnostic segmentation head into our model. This head, based on the Fully Convolutional Network (FCN) architecture [22], takes as input the feature maps extracted from the Transformer decoder and produces segmentation masks.

Text Prompts Prompt tuning is crucial for adapting pretrained language models to downstream computer vision tasks [43]. We adopt the prompt construction strategy from ViLD [13]. For each class, we generate 63 different textual prompts (e.g., “there is a class name in the photo”) and feed them to CLIP’s text encoder. Subsequently, we average the resulting 85 text embeddings, using the concept of prompt ensembling from [27].

Implementation Details Our model is founded upon Deformable DETR [46]. Training was conducted on 8 Tesla V100 GPUs. We employ ResNet50-C4 as the default backbone. Following ViLD, we utilize the open-source CLIP model based on ViT-B/32 for extracting text and image embeddings. Most hyper-parameters are maintained consistent with prior works [13, 46]. For the loss function, the weighting parameters are set as follows: $\mathcal{L}_{BCE} = 3.0$, $\mathcal{L}_{L1} = 5.0$, $\mathcal{L}_{IoU} = 2.0$ and $\mathcal{L}_{embed} = 1.0$.

10.4.1 Results on Open-Vocabulary Benchmarks

Table 10.1 summarizes our results. We compare OV-DETR with the following open-vocabulary detection baselines: (1) OVR-CNN [41] (see Table 10.1 row 4). This method pretrains the detector’s projection layer on image-caption pairs using a contrastive loss and subsequently fine-tunes it for object detection. (2) Variants of ViLD [13] such as ViLD-text and ViLD-ensemble (see Table 10.1 rows 5–7). ViLD is the first study that uses CLIP embeddings [27] for open-vocabulary detection. Compared with ViLD-text, ViLD uses knowledge distillation from the CLIP visual encoder, improves AP_{novel} at the cost of hurting AP_{base} . ViLD-ensemble combines the two models and shows improvements on both metrics. However, this ensemble approach incurs additional computational overhead in terms of time and memory.

For completeness, we also list the results of some previous zero-shot methods such as SB [1], DELO [45] and PL [28] in Table 10.1 rows 1–3. On *OV-LVIS* benchmark, OV-DETR improves the ViLD by 4.1 on AP^m and 1.3 on AP^m_{novel} . Compared with ViLD, OV-DETR will not affect the performance of *base* classes when improving the *novel* classes. Even compared with the ensemble result of ViLD-ensemble, OV-DETR still boosts the performance by 1.5, 0.8, 1.0, and 2.2, respectively (%). Note that OV-DETR only uses a single model and does not use any ensemble-based technique. On *OV-COCO* benchmark, OV-DETR improves the baseline and outperforms OVR-CNN [41] by a large margin, notably, the 6.6 mAP improvements on *novel* classes. Compared with ViLD [13], OV-DETR still achieves 1.4 mAP gains on all the classes and 1.8 mAP gains on *novel* classes. In summary, OV-DETR demonstrates superior performance across various datasets when compared to other methods.

10.4.2 Generalization Ability of OV-DETR

We follow ViLD [13] to test the generalization ability of OV-DETR by training the model on LVIS [14] dataset and evaluated on other datasets (PASCAL VOC [7], COCO [21]). We keep the same implementation details with ViLD [13]. For transfer to other datasets, we simply switched the conditional text queries (z^{text}) to those corresponding to the new target datasets. As shown in Table 10.2, OV-DETR demonstrated superior transfer performance compared to ViLD. These experimental results underscore the strong transferability of our model, which is attributed to the effectiveness of our conditional-based training mechanism.

Table 10.1 Main results on OV-LVIS and OV-COCO. For OV-LVIS (w/ 886 base classes and 317 novel classes), we report mask mAP and a breakdown on novel (rare), common, and frequent classes. For OV-COCO (w/ 48 base classes and 17 novel classes), we report bounding box mAP at IoU threshold 0.5. †: zero-shot methods that do not use captions or image-text pairs. ‡: ensemble model

#	Method	OV-LVIS				OV-COCO			
		AP ^m	AP ^m _{novel}	AP ^m _c	AP ^m _f	AP50 ^b	AP50 ^b _{novel}	AP50 ^b _{base}	
1	SB [1]†	–	–	–	–	24.9	0.3	29.2	
2	DELO [45]†	–	–	–	–	13.0	3.1	13.8	
3	PL [28]†	–	–	–	–	27.9	4.1	35.9	
4	OVR-CNN [41]	–	–	–	–	46.0	22.8	39.9	
5	ViLD-text [13]	24.9	10.1	23.9	32.5	49.3	5.9	61.8	
6	ViLD [13]	22.5	16.1	20.0	28.3	51.3	27.6	59.5	
7	ViLD-ens. [13]‡	25.5	16.6	24.6	30.3	–	–	–	
8	OV-DETR	26.6	17.4	25.0	32.5	52.7	29.4	61.0	
	(ours vs. #6)	(+4.1)	(+1.3)	(+5.0)	(+4.2)	(+1.4)	(+1.8)	(+1.5)	

Table 10.2 Generalization to other datasets. We evaluate OV-DETR trained on LVIS when transferred to other datasets such as PASCAL VOC 2007 test set and COCO validation set by simply replacing the text embeddings. The experimental setting is the same as that of ViLD [13]. We observe that OV-DETR achieves better generalization performance than ViLD [13]

#	Method	Pascal VOC		COCO		
		AP ₅₀ ^b	AP ₇₅ ^b	AP ^b	AP ₅₀ ^b	AP ₇₅ ^b
1	ViLD-text [13]	40.5	31.6	28.8	43.4	31.4
2	ViLD [13]	72.2	56.7	36.6	55.6	39.8
3	OV-DETR	76.1	59.3	38.1	58.4	41.1
	(ours vs #2)	(+3.9)	(+2.6)	(+1.5)	(+2.8)	(+1.3)

10.4.3 Qualitative Results

Open-Vocabulary LVIS Figure 10.6 visualizes the detection and segmentation results of OV-DETR. The top, middle, and bottom rows showcase results based on conditional text queries, conditional image queries, and a combination of both, respectively. Overall, OV-DETR demonstrates the ability to accurately localize and precisely segment target objects from novel classes, despite the absence of any training annotations for these classes. Significantly, OV-DETR exhibits robustness even when presented with conditional image queries that differ substantially from the target images. For instance, in Fig. 10.6d and h, the conditional image queries (“crape” and “fork”) are visually distinct from the corresponding objects in the target images, yet OV-DETR successfully detects them.

Open-Vocabulary COCO We provide more qualitative results of OV-DETR on Open-Vocabulary COCO setting (Fig. 10.7). We visualize the detection results on novel classes, along with the activation maps of OV-DETR and the Region Proposal Network (RPN) employed by ViLD. This comparative analysis provides further validation of our motivation: OV-DETR exhibits significantly higher activation values on objects belonging to novel classes compared to RPN.

Web Images To further demonstrate generalization ability, Fig. 10.8 presents qualitative results on anime characters, a domain entirely unseen during training. Despite this lack of training data, OV-DETR effectively detects regions that correspond to the provided conditional image queries. This observation highlights the model’s capacity to generalize to novel domains and visual concepts.

Failure Cases Figure 10.9 illustrates several failure cases observed in OV-DETR. Challenges include:

- **Detection of small or occluded objects:** The model exhibits difficulty in accurately detecting objects that are small in size or significantly occluded.
- **Sensitivity to unrelated out-of-distribution text queries:** The performance can be adversely affected by unrelated or out-of-distribution text queries.

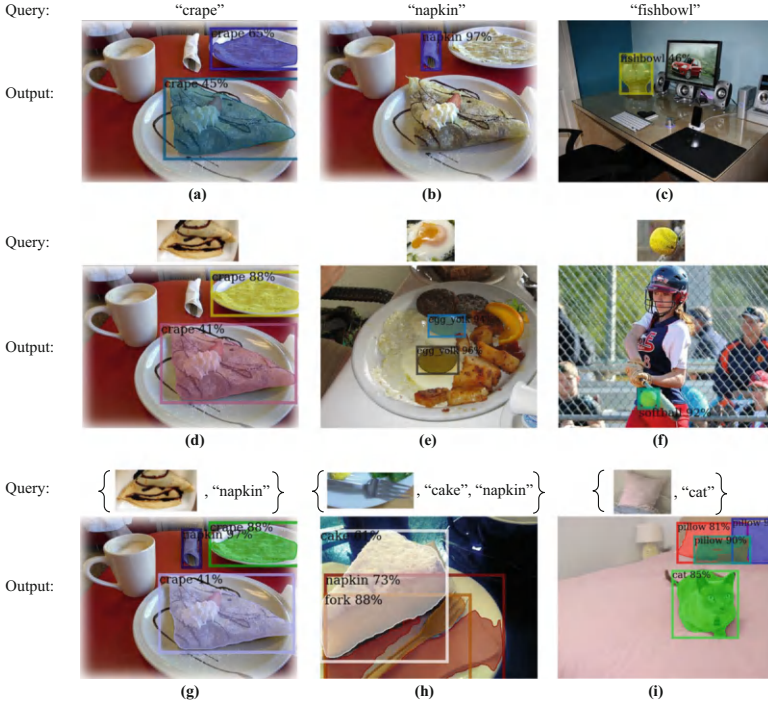


Fig. 10.6 Qualitative results on LVIS. OV-DETR can precisely detect and segment novel objects (e.g., ‘crape’, ‘fishbowl’, ‘softball’) given the conditional text query (top) or conditional image query (middle) or a mixture of them (bottom) (Image sourced from [39])

10.4.4 Inference Time Analysis

OV-DETR demonstrates significant potential for open-vocabulary detection, yet it is not without limitations. A major drawback is its slow inference speed, particularly pronounced when dealing with a large number of classes, such as the 1,203 classes found in the LVIS dataset. This performance bottleneck stems from the conditional design, which necessitates multiple forward passes through the Transformer decoder for each class, significantly increasing the overall processing time.

We show a detailed comparison of the inference time between Deformable DETR and OV-DETR in Table 10.3. Without using any tricks, the vanilla OV-DETR (#2), i.e., using a single forward pass for each class, is about $2\times$ slower than Deformable DETR (#1) on COCO (w/ 80 classes) while $16\times$ slower on LVIS (w/ 1,203 classes). As discussed in Sect. 10.3.2 and shown in Fig. 10.5b, we optimize the inference process by forwarding multiple conditional queries in parallel to the Transformer decoder. This optimization strategy results in a substantial reduction in inference time: approximately 12.5% on COCO and nearly 60% on LVIS (see

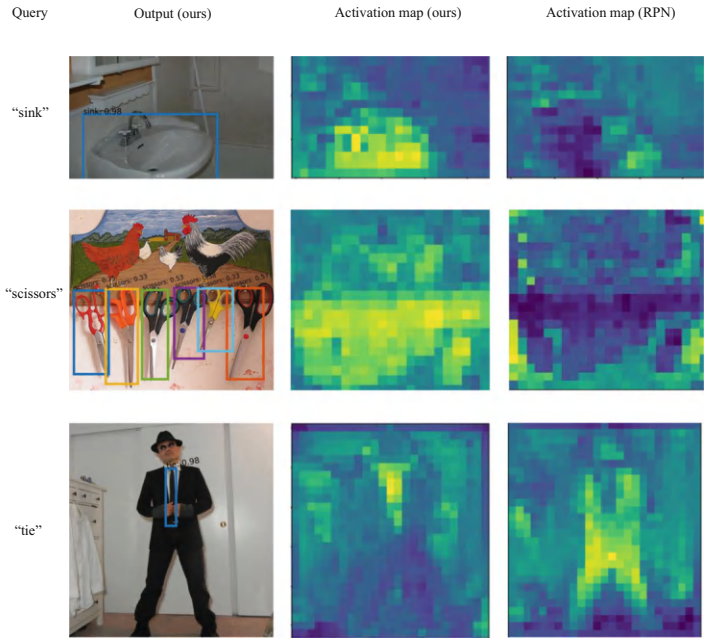


Fig. 10.7 Qualitative results on Open-Vocabulary COCO setting. We visualize the prediction results of OV-DETR on *novel* classes. We also provide a comparison of activation maps between ours and the RPN network (Image sourced from [39])

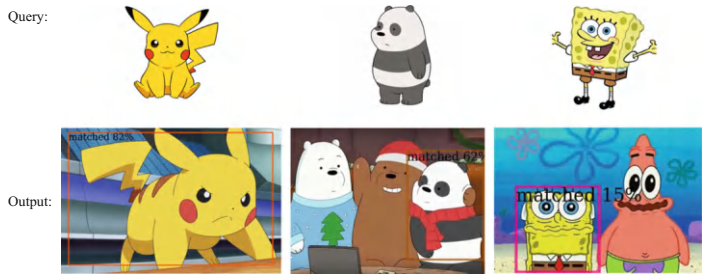


Fig. 10.8 Qualitative results on anime characters. These images of anime characters are sourced from the web. Notably, these images and their associated classes were not included in the training dataset. We employed the model trained on the LVIS dataset to assess matchability with the provided conditional image queries (Image sourced from [39])

#3 in Table 10.3). Despite this improvement, there remains ample room for further optimization.

It is important to note that the slow inference speed is not unique to our approach. Many instance-conditional models exhibit similar performance-speed trade-offs [19]. This often necessitates a compromise between accuracy and efficiency. The computational bottleneck in our method stems from the Transformer

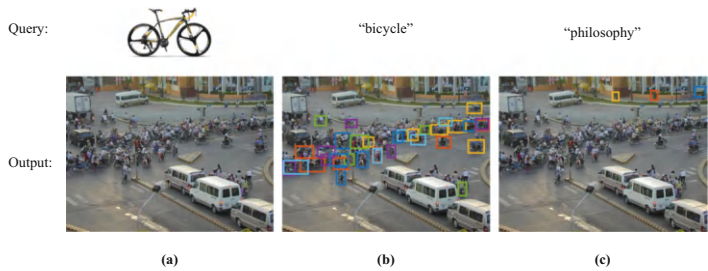


Fig. 10.9 Failure cases of OV-DETR. These images are collected from the web. We use the model trained on LVIS dataset to check the matchability with the given conditional image or text queries. (a) OV-DETR fails to detect these small and occluded objects with the conditional image query (“bicycle”). But as shown in (b), this issue can be solved to some extent by using text query. (c) Given the unrelated text queries (e.g., “philosophy”), OV-DETR will predict wrong false-positive detection results (Image sourced from [39])

Table 10.3 Comparison of the inference time (second per iteration) between Deformable DETR [46] and our OV-DETR before/after optimization

#	Method	COCO	LVIS
1	Deformable DETR	0.31	1.49
2	Ours	0.72	23.84
3	Ours (optimized)	0.63	9.57
4	(#3 vs #2)	↓ 12.5%	↓ 59.9%

decoder operations, as defined in Eq.(10.2). To address this, we can explore the integration of more efficient attention mechanisms, such as sparse attention [32, 33]. These advancements will be a focus of future research. Furthermore, in practical human-computer interaction scenarios, users typically have specific target objects in mind, such as a missing luggage item or a particular logo. In these cases, the number of conditional inputs remains relatively small and fixed, rendering the inference time negligible.

10.5 Conclusion

Open-vocabulary object detection presents a significant challenge due to the lack of training data for unseen classes. This chapter discusses how to adapt the Transformer-based detector for open-vocabulary scenarios through conditional matching and the integration of pretrained vision-language models. Despite its simplified training pipeline, OV-DETR demonstrates superior performance compared to existing baselines, all of which are based on two-stage detectors. These findings underscore the potential of Transformer-based architectures for advancing the field of open-vocabulary object detection.

Further Reading

Interested readers may refer to the following material:

- Wu et al. [36], Zang et al. [40] for combining the object detection task with Multimodal Large Language Models (MLLMs).
- Wang et al. [34] for 3D open-vocabulary object detection based on DETR.
- Wu et al. [35] for a comprehensive survey on open-vocabulary object detection.

References

1. Bansal, A., Sikka, K., Sharma, G., Chellappa, R., Divakaran, A.: Zero-shot object detection. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
3. Chen, Q., Chen, X., Zeng, G., Wang, J.: Group DETR: Fast training convergence with decoupled one-to-many label assignment. arXiv preprint arXiv:2207.13085 (2022)
4. Dai, Z., Cai, B., Lin, Y., Chen, J.: UP-DETR: Unsupervised pre-training for object detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (2019)
7. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) challenge. *Int J Comput Vision* **88**, 303–338 (2010)
8. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: DeViSE: A deep visual-semantic embedding model. In: Advances in Neural Information Processing Systems (2013)
9. Gao, P., Zheng, M., Wang, X., Dai, J., Li, H.: Fast convergence of detr with spatially modulated co-attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
10. Gao, M., Xing, C., Niebles, J.C., Li, J., Xu, R., Liu, W., Xiong, C.: Towards open vocabulary object detection without human-provided boxes. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (2012)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (2014)
13. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. In: International Conference on Learning Representations (2022)
14. Gupta, A., Dollar, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)

15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
16. Kang, K., Ouyang, W., Li, H., Wang, X.: Object detection from video tubelets with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
17. Kim, D., Lin, T.Y., Angelova, A., Kweon, I.S., Kuo, W.: Learning open-world object proposals without learning to classify. *IEEE Robot. Autom. Lett.* **7**(2), 5453–5460 (2022)
18. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval Res Logist Quart* **2**(1–2), 83–97 (1955)
19. Li, S., Xiao, T., Li, H., Zhou, B., Yue, D., Wang, X.: Person search with natural language description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
20. Li, Z., Yao, L., Zhang, X., Wang, X., Kanhere, S., Zhang, H.: Zero-shot object detection with textual descriptions. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)
21. 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: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
22. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
23. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional DETR for fast training convergence. In: Proceedings of the IEEE International Conference on Computer Vision (2021)
24. 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. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
25. Ouyang-Zhang, J., Cho, J.H., Zhou, X., Krähenbühl, P.: Nms strikes back. *arXiv preprint arXiv:2212.06137* (2022)
26. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014)
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021)
28. Rahman, T., Chou, S.H., Sigal, L., Carenini, G.: An improved attention for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (2015)
30. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
31. Shizhen, Z., Changxin, G., Yuanjie, S., Lerenhan, L., Changqian, Y., Zhong, J., Nong, S.: Gtnet: Generative transfer network for zero-shot object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence (2020)
32. Tay, Y., Bahri, D., Yang, L., Metzler, D., Juan, D.C.: Sparse sinkhorn attention. In: International Conference on Machine Learning (2020)
33. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768* (2020)
34. Wang, Z., Li, Y., Liu, T., Zhao, H., Wang, S.: OV-Uni3DETR: towards unified open-vocabulary 3d object detection via cycle-modality propagation. In: European Conference on Computer Vision, pp. 73–89. Springer, Berlin (2025)

35. Wu, J., Li, X., Xu, S., Yuan, H., Ding, H., Yang, Y., Li, X., Zhang, J., Tong, Y., Jiang, X., et al.: Towards open vocabulary learning: A survey. In: *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(7), 5092–5113 (2024)
36. Wu, S., Jin, S., Zhang, W., Xu, L., Liu, W., Li, W., Loy, C.C.: F-LMM: Grounding frozen large multimodal models. *arXiv preprint arXiv:2406.05821* (2024)
37. Xie, J., Zheng, S.: ZSD-YOLO: Zero-shot yolo detection using vision-language knowledge distillation. *arXiv preprint arXiv:2109.12066* (2021)
38. Zang, Y.: Real-world object detection. In: Nanyang Technological University, PhD thesis. (2023)
39. Zang, Y., Li, W., Zhou, K., Huang, C., Loy, C.C.: Open-vocabulary detr with conditional matching. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2022)
40. Zang, Y., Li, W., Han, J., Zhou, K., Loy, C.C.: Contextual object detection with multimodal large language models. *Int. J. Comput. Vision* **133**(2), 825–843 (2024)
41. Zareian, A., Rosa, K.D., Hu, D.H., Chang, S.F.: Open-vocabulary object detection using captions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2021)
42. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: DINO: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605* (2022)
43. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. *Int. J. Comput. Vision* **130**(9), 2337–2348 (2022)
44. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., Misra, I.: Detecting twenty-thousand classes using image-level supervision. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2022)
45. Zhu, P., Wang, H., Saligrama, V.: Don't Even Look Once: Synthesizing features for zero-shot detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2020)
46. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. In: *International Conference on Learning Representations* (2020)
47. Zong, Z., Song, G., Liu, Y.: Detsr with collaborative hybrid assignments training. *arXiv preprint arXiv:2211.12860* (2022)

Chapter 11

Unlocking CLIP for Zero-Shot Dense Segmentation



Chong Zhou, Chen Change Loy, and Bo Dai

Abstract Contrastive Language-Image Pre-training (CLIP) represents a ground-breaking advancement in open-vocabulary zero-shot image recognition, with its applications extending well beyond image-level tasks. This chapter discusses MaskCLIP, an early attempt to examine the intrinsic potential of CLIP for pixel-level dense prediction, specifically in semantic segmentation. With minimal modifications to CLIP, MaskCLIP shows impressive segmentation performance on open concepts across diverse datasets, all without requiring annotations or fine-tuning. In addition, through the incorporation of pseudo-labeling and self-training, the enhanced MaskCLIP+ model further improves MaskCLIP under the transductive zero-shot semantic segmentation setting. We also test the robustness of MaskCLIP under input corruption and evaluate its capability in discriminating fine-grained objects and novel concepts. Our finding suggests that MaskCLIP can serve as a reliable source of supervision for dense prediction tasks to achieve annotation-free segmentation.

Keywords Semantic segmentation · Open-vocabulary recognition · Pseudo-labeling · Self-training · Transductive zero-shot learning

Bo Dai completed this work when he was with S-Lab, NTU.

C. Zhou (✉) · C. C. Loy
S-Lab, Nanyang Technological University, Singapore, Singapore
e-mail: chong033@ntu.edu.sg; ccloy@ntu.edu.sg

B. Dai
Shanghai AI Laboratory, Shanghai, China
e-mail: daibo@pjlab.org.cn

11.1 Introduction

Large-scale vision-language pre-training models, such as CLIP [44], have demonstrated the ability to learn expressive visual and textual features from massive datasets. And the learned features have been successfully utilized across various downstream tasks, including text-driven image manipulation [41], image captioning [24], view synthesis [29], and object detection [18]. Notably, zero-shot image classification with raw CLIP features achieves competitive performance with fully supervised counterparts [44].

This chapter discusses the applicability of CLIP features for pixel-level dense prediction tasks, particularly semantic segmentation. While previous work leverages CLIP features for global image-level tasks, our investigation seeks to uncover whether these features also encode sufficient local and object-level semantics to support dense prediction. Unlike traditional image classification tasks that primarily involve iconic images, the pre-training of CLIP on diverse scene images paired with natural language descriptions offers three key advantages: (1) the embedding of local semantic information, (2) open-vocabulary concept learning, and (3) the incorporation of contextual knowledge, such as object co-occurrence/relation and spatial priors. These attributes suggest that CLIP’s dense features hold significant potential for semantic segmentation.

In this chapter, we share both our success and failure experience in leveraging CLIP features for dense prediction. A critical insight we uncovered is the importance of preserving the visual-language association inherent in CLIP’s original feature space. Our early attempts to fine-tune the image encoder of CLIP for the segmentation task, e.g., initializing DeepLab [5] with CLIP’s image encoder weights and fine-tuning the backbone on segmentation objectives, break this association and lead to suboptimal performance. Similarly, efforts to manipulate CLIP’s text embeddings fail in segmenting unseen classes.

Our successful model, MaskCLIP, builds on these lessons. By directly extracting dense patch-level features from the *value* features of CLIP’s final attention layer, MaskCLIP retains the integrity of the visual-language association. Classification weights for dense prediction, which are essentially 1×1 convolutions, are derived directly from CLIP’s text embeddings, requiring no additional mapping or modifications. This simple yet effective method allows MaskCLIP to produce reasonable predictions while maintaining compatibility with different variants of CLIP, including those using ResNet and ViT backbones. We also discuss two mask refinement techniques for MaskCLIP to further improve its performance, namely, *key smoothing* and *prompt denoising*; both require no training. Specifically, key smoothing computes the similarity between the *key* features (of the last attention layer) of different patches, which are used to smooth the predictions. Prompt denoising removes prompts with classes that unlikely exist in the image; thus, with fewer distractors, predictions become more accurate.

Despite these advances, MaskCLIP’s reliance on CLIP’s original image encoder imposes architectural limitations that constrain its segmentation capacity. To address

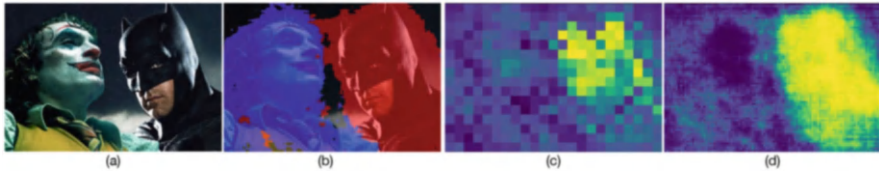


Fig. 11.1 Here, we show the original image in (a), the segmentation result of MaskCLIP+ in (b) and the confidence maps of MaskCLIP and MaskCLIP+ for *Batman* in (c) and (d), respectively. Through the adaptation of CLIP, MaskCLIP can be directly used for segmentation of fine-grained and novel concepts (e.g., *Batman* and *Joker*) without any training operations and annotations. Combined with pseudo-labeling and self-training, MaskCLIP+ further improves the segmentation result. Image sourced from [55]

this, we discuss an extended approach, MaskCLIP+, which uses MaskCLIP as a pseudo-label generator during training. By combining these pseudo-labels with a standard self-training strategy, MaskCLIP+ leverages the robustness and generalizability of CLIP’s features while enabling the use of more advanced segmentation architectures, such as PSPNet [54] and DeepLab [5].

Apart from annotation-free and open-vocabulary segmentation, MaskCLIP+ can also be applied to the transductive zero-shot semantic segmentation task, where MaskCLIP only generates pseudo-labels for the unseen classes. Thanks to the the generality and robustness of CLIP features, MaskCLIP+ can be readily applied to various extended settings of semantic segmentation, including the segmentation of fine-grained classes (e.g., attribute-conditioned classes like *white car* and *red bus*) or novel concepts (such as *Batman* and *Joker* as shown in Fig. 11.1), as well as the segmentation of moderately corrupted inputs. We show more interesting results in the experiment section.

Semantic segmentation is notorious for its high dependency on labeled training data. Many methods have been explored to get around such stringent requirement, e.g., through using weak labels like image tags, bounding boxes, and scribbles. Our study shows that features learned via large-scale visual-language pre-training can be readily used to facilitate open-vocabulary dense prediction. Our model, MaskCLIP, shows promising potential in providing rich and meaningful dense pseudo-labels for training existing methods.

11.2 Related Work

Transferable Representation Learning Pre-training has become a cornerstone of dense prediction tasks. Early studies, such as Yosinski et al. [51], demonstrate that ImageNet [10] pre-training significantly accelerates convergence in downstream tasks like object detection. Subsequent research has focused on removing the dependency on human-labeled data by employing self-supervised representation learning. These approaches include designing pretext tasks [12, 13, 38], contrastive

learning [6, 8, 21], clustering [3], and bootstrapping methods [17] to generate supervision signals. An alternative direction involves learning visual representations from natural language. Early works [11, 15, 16, 43, 47, 52] train models on image-caption pairs, while large-scale frameworks like CLIP [44] and ALIGN [30] leverage contrastive learning on extensive web-curated image-text datasets. These models exhibit strong zero-shot transferability and inspire numerous downstream applications, including text-driven image manipulation [41], image captioning [24], view synthesis [29], and object detection [18]. Different from these methods that typically apply CLIP right off the shelf for image encoding, we explore ways to adapt CLIP for pixel-level dense prediction. A concurrent work, DenseCLIP [45], aims to better fine-tune the CLIP pre-trained weights on target semantic segmentation datasets without keeping the zero-shot transferability, which are different from our setting. To examine the intrinsic potential of CLIP for dense prediction tasks, we refrain from any fine-tuning and major architectural modification.

Zero-Shot Visual Recognition Zero-shot learning targets classifying instances of categories not seen during training by leveraging clues such as shared attributes or visual-semantic mappings. The latter, which avoids the need for additional annotations, is particularly suitable for zero-shot dense prediction tasks. Prior works like Zhao et al. [53] map image pixel features and word concepts into a shared embedding space, while Kato et al. [31] integrate semantic features into visual features for guidance. Approaches such as ZS3Net [1] synthesize pixel-level features for unseen classes, and SPNet [49] learns to project visual features into semantic space. Other studies enhance these methods by improving uncertainty estimation [25], structural consistency [33], or contextual understanding [19]. STRICT [40] employs self-training to boost the performance of SPNet. The zero-shot segmentation setting can be categorized into inductive (unseen pixels are not observed during training) and transductive (unseen pixels are part of the training set). Our MaskCLIP not only performs well in traditional zero-shot segmentation tasks but also handles more challenging scenarios where all categories are unseen during training.

Self-Training Self-training is widely adopted in semi-supervised classification [28, 32, 34, 42], leveraging models trained on labeled data to generate pseudo-labels for unlabeled samples, which are then iteratively used to improve the model. This methodology has also been extended to semi-supervised and zero-shot semantic segmentation [4, 9, 26, 27, 35, 36, 39, 56]. In our approach, MaskCLIP+ builds on this philosophy. Pseudo-labels are generated by a frozen MaskCLIP model as well as by MaskCLIP+ itself in an iterative self-training loop. This iterative process allows MaskCLIP+ to refine its predictions and improve its performance, particularly in complex and open-vocabulary segmentation tasks.

11.3 Methodology

Our study serves as an early attempt that explores the applicability of CLIP features for pixel-level dense prediction tasks. We start with a brief introduction of CLIP and a naïve solution as the preliminary, followed by presenting our MaskCLIP in detail.

11.3.1 Preliminary on CLIP

CLIP [44] is a visual-language pre-training method that learns both visual and language representations from large-scale raw web-curated image-text pairs. It consists of an image encoder $\mathcal{V}(\cdot)$ and a text encoder $\mathcal{T}(\cdot)$, both jointly trained to, respectively, map the input image and text into a unified representation space. CLIP adopts contrastive learning as its training objective, where ground-truth image-text pairs are regarded as positive samples and mismatched image-text pairs are constructed as negative ones. In practice, the text encoder is implemented as a Transformer [48]. As for the image encoder, CLIP provides two alternative implementations, namely, a Transformer and a ResNet [22] with global attention pooling layer. Our method can be based on both encoder architectures.

We believe CLIP has inherently embedded local image semantics in its features as it learns to associate image content with natural language descriptions, the latter of which contain complex and dense semantic guidance across multiple granularities. For example, to correctly identify the image corresponds to the description *the man at bat readies to swing at the patch while the umpire looks on* [7], CLIP must divide image semantics into local segments and properly align image semantics with singular mentioned concepts like *man*, *bat*, *swing*, *patch*, *man at bat*, *man at patch*, and *man readies to swing*, instead of handling the image as a whole. Such uniqueness is absent from training with solely image labels.

11.3.2 Conventional Fine-Tuning Hinders Zero-Shot Ability

The current de facto pipeline of training a segmentation network is (1) initializing the backbone network with the ImageNet [10] pre-trained weights, (2) adding segmentation-specific network modules with randomly initialized weights, and (3) jointly fine-tuning the backbone and newly added modules.

It is natural to follow these standard steps to adapt CLIP for segmentation. Here, we start our exploration by applying this pipeline on DeepLab [5] with two CLIP-specific modifications. Specifically, we first replace the ImageNet pre-trained weights with weights of the image encoder of CLIP. Second, we adopt a mapper $\mathcal{M}$

that maps text embeddings of CLIP to the weights of DeepLab classifier (the last 1×1 convolutional layer). The modified model can be formulated as follows:

$$\text{DeepLab}(x) = C_\phi(\mathcal{H}(\mathcal{V}_{*l}(x))), \quad (11.1)$$

$$\phi = \mathcal{M}(t), \quad (11.2)$$

where $\mathcal{V}_{*l}(\cdot)$ denotes the DeepLab backbone, which is a ResNet dilated by a factor of l . $\mathcal{H}(\cdot)$ denotes the randomly initialized ASPP module [5], and $C_\phi(\cdot)$ is the DeepLab classifier, whose weights, denoted as ϕ , are determined by the text embedding of CLIP via the mapper $\mathcal{M}$. Ideally, by updating the classifier weights with the corresponding text embedding, the adapted DeepLab is able to segment different classes without retraining.

To evaluate the segmentation performance of this modified DeepLab on both seen and unseen classes, we train it on a subset of classes in the dataset, considering the remaining classes as unseen ones. We have tried a series of mapper architectures. Although they perform well on seen classes, in all these cases, the modified DeepLab fails to segment unseen classes with satisfying performance. We hypothesize that this is mainly because the original visual-language association of CLIP features has been broken: (1) the backbone is slightly different from the image encoder in terms of network architecture; (2) weights initialized from the image encoder have been updated during fine-tuning; (3) an extra mapper, which is trained only on data of seen classes, is introduced therefore leading to insufficient generality.

11.3.3 MaskCLIP

Failing the fine-tuning attempt, we turn to a solution that avoids introducing additional parameters and modifying the feature space of CLIP. To this end, we carefully revisit the image encoder of CLIP, especially its unique global attention pooling layer. As shown in Fig. 11.2b, different from conventional global averaged pooling, the image encoder of CLIP adopts a Transformer-style multi-head attention layer where globally average-pooled feature works as the query, and feature at each spatial location generates a key-value pair. Consequently, the output of this layer is a spatial weighted-sum of the incoming feature map followed by a linear layer $\mathcal{F}(\cdot)$:

$$\begin{aligned} \text{AttnPool}(\bar{q}, k, v) &= \mathcal{F}\left(\sum_i \text{softmax}\left(\frac{\bar{q}k_i^T}{C}\right)v_i\right) \\ &= \sum_i \text{softmax}\left(\frac{\bar{q}k_i^T}{C}\right)\mathcal{F}(v_i), \end{aligned} \quad (11.3)$$

$$\bar{q} = \text{Emb}_q(\bar{x}), k_i = \text{Emb}_k(x_i), v_i = \text{Emb}_v(x_i), \quad (11.4)$$

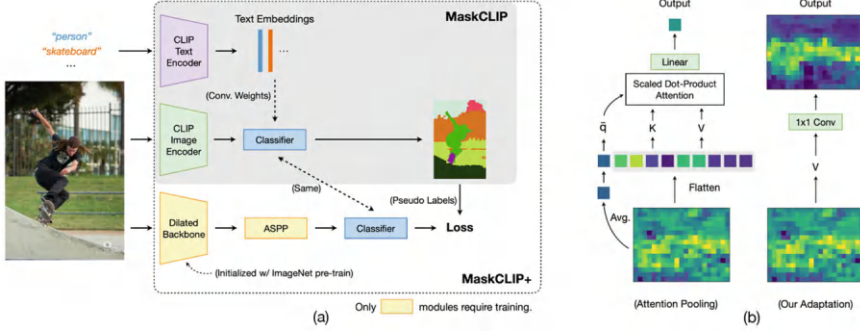


Fig. 11.2 Overview of MaskCLIP/MaskCLIP+. Compared to the conventional fine-tuning method, the key to the success of MaskCLIP is keeping the pre-trained weights frozen and making minimal adaptation to preserve the visual-language association. Besides, to compensate for the weakness of using the CLIP image encoder for segmentation, which is designed for classification, MaskCLIP+ uses the outputs of MaskCLIP as pseudo-labels and trains a more advanced segmentation network such as DeepLabv2 [5]. Image sourced from [55]

where C is a constant scaling factor and $\text{Emb}(\cdot)$ denotes a linear embedding layer.¹ x_i represents the input feature at spatial location i and $\bar{x}$ is the average of all x_i . The outputs of the Transformer layer serve as a comprehensive representation of the whole image. We believe that this is possible because $\mathcal{F}(v_i)$ computed at each spatial location already captures a rich response of local semantics that correspond well with tokens in the text embeddings of CLIP.

Based on such a hypothesis, as shown in Fig. 11.2b, we directly modify the image encoder of CLIP in our new attempt: (1) removing the query and key embedding layers and (2) reformulating the value-embedding layer and the last linear layer into two respective 1×1 convolutional layers. Moreover, we keep the text encoder unchanged, and it takes prompts with target classes as the input. The resulting text embedding of each class is used as the classifier. We name the resulting model as MaskCLIP since it yields pixel-level mask predictions instead of a global image-level prediction. We then evaluate MaskCLIP on various standard segmentation benchmarks as well as web-crawled images. As shown in Fig. 11.1, MaskCLIP can output reasonable results without any fine-tuning nor annotations. More qualitative results and quantitative results with respect to the mIoU metric are included in the experiment section.

One might argue that, since the global attention pooling is a self-attention layer, even without modification, it can also generate dense features. However, since query $\bar{q}$ is the only query trained during the CLIP pre-training, this naïve solution fails. We treat this solution as the baseline and compare its results with ours in the experiments. Moreover, the Transformer layer in ViT is very similar to the

¹ Here we have simplified the formula by ignoring the channel-wise splitting and concatenation.

global attention pooling. In fact, the only two differences are: (1) the global query is generated by a special [CLS] token instead of the average among all spatial locations; (2) Transformer layer has a residual connection. Therefore, by replacing $\bar{q}$ with $q_{[cls]}$ and adding input x to the output, MaskCLIP can work with the ViT backbone.

Despite the simplicity of MaskCLIP in comparison to existing segmentation approaches, our method enjoys multiple unique merits inherited from CLIP. First, MaskCLIP can be used as a free segmentation annotator to provide rich and novel supervision signals for segmentation methods working with limited labels. Second, since the visual-language association of CLIP is retained in MaskCLIP, it naturally possesses the ability to segment open-vocabulary classes, as well as fine-grained classes described by free-form phrases, such as *white car* and *red bus*. Third, since the CLIP is trained on raw web-curated images, CLIP demonstrates great robustness to natural distribution shift [44] and input corruptions [46]. We verify that MaskCLIP preserves such robustness to some extent.

Key Smoothing and Prompt Denoising To further improve the performance of MaskCLIP, we discuss two refinement strategies, namely, *key smoothing* and *prompt denoising*. Recall that, in Eq. (11.3), besides $\bar{q}$, key features k_i also get trained during CLIP pre-training. However, in the original MaskCLIP, k_i is simply discard. Hence, here we seek to utilize this information to refine the final output. Key features can be viewed as the descriptor of the corresponding patch; therefore, patches with similar key features should yield similar predictions. With this hypothesis, we smooth the predictions with

$$\text{pred}_i = \sum_j \cos\left(\frac{k_i}{\|k_i\|_2}, \frac{k_j}{\|k_j\|_2}\right) \text{pred}_j, \quad (11.5)$$

where k_i and pred_i are key features and class confidence predictions at spatial location i , while $\|\cdot\|_2$ and $\cos(\cdot)$ denote L2 normalization and cosine similarity. We name this strategy as key smoothing.

In addition, we also observe that when dealing with many target classes, since only a small proportion of the classes appear in a single image, the rest classes are in fact distractors and undermine the performance. Therefore, we adopt prompt denoising, which removes the prompt with target class if its class confidence at all spatial locations is all less than a threshold $t = 0.5$.

11.3.4 MaskCLIP+

While MaskCLIP does not require any training, its network architecture is rigid because it adopts the image encoder of CLIP. To relax it from this constraint and benefit from more advanced architectures tailored for segmentation, such as DeepLab [5] and PSPNet [54], we propose MaskCLIP+. Instead of directly applying

MaskCLIP for test-time prediction, MaskCLIP+ regard its predictions as training-time pseudo-ground-truth labels. Together with an adopted self-training strategy, MaskCLIP+ is thus free from the restriction on its backbone architecture. As shown in Fig. 11.2a, we take DeepLabv2 [5] as the backbone of MaskCLIP+ to ensure a fair comparison with previous segmentation methods.

MaskCLIP-Guided Learning In MaskCLIP+, we leverage the predictions of MaskCLIP to guide the training of another target network comprising an architecture tailored to segmentation task. In parallel to the target network, we feed the same preprocessed image input to the MaskCLIP and use the predictions of MaskCLIP as pseudo-ground-truth labels to train the target network. In addition, we replace the classifier of the target network with that of MaskCLIP, to preserve the network’s ability for open-vocabulary prediction.

MaskCLIP-guided learning is also applicable in the transductive zero-shot segmentation setting. Specifically, while pixels of both seen and unseen classes are observed, only annotations of seen classes are available. In this case, we only use MaskCLIP to generate pseudo-labels for the unlabeled pixels. Compared to SOTA methods, MaskCLIP+ obtains remarkably better results across three standard benchmarks, namely, PASCAL VOC 2012 [14], PASCAL Context [37], and COCO Stuff [2], where the results of MaskCLIP+ are even on par with that of fully supervised baselines.

We note that some related attempts [18, 50], targeting object detection, perform knowledge distillation between the image-level visual features of CLIP and the features of a target model. Different from such feature-level guidance, we adopt pseudo-labels in our case. This is because our target network, with a segmentation-tailored architecture, is structurally distinct from the image encoder of CLIP. Therefore, distillation by feature matching may be a suboptimal strategy. In fact, as reported by [18], under zero-shot setting, such feature-level guidance indeed results in conflicts between the performance of seen and unseen classes. On the contrary, by adopting pseudo-labels in MaskCLIP+, we do not observe any performance drop on seen classes.

Self-Training It is expected that after certain training iterations, the target network guided by MaskCLIP will outperform MaskCLIP, rendering the latter suboptimal for further guidance as it gradually becomes an inferior model over time. Empirically, we also find that MaskCLIP-guided learning reaches an upper bound at around 1/10 of the standard training schedule. To further improve the performance, we swap out MaskCLIP and let the target model generate pseudo-labels for itself. This is commonly referred to as self-training.

11.4 Experiments

Datasets We conduct experiments on three standard segmentation benchmarks, namely, PASCAL VOC 2012 [14], PASCAL Context [37], and COCO Stuff

[2]. PASCAL VOC 2012 contains 1,426 training images with 20 object classes plus a background class. Following common practice, we augment it with the Semantic Boundaries Dataset [20]. PASCAL Context labels PASCAL VOC 2010 (4,998/5,105 train/validation images) with segmentation annotations of 520 object/stuff classes, from which the most common 59 classes are treated as foreground while the rest are regarded as background. COCO Stuff extends the COCO dataset, which contains segmentation annotations of 80 object classes on 164K images, with additional 91 stuff classes.

Text Embedding We follow the same process to construct text embeddings as Gu et al. [18]. Specifically, we feed prompt engineered texts into the text encoder of CLIP with 85 prompt templates, such as *there is a {class name} in the scene*, and average the resulting 85 text embeddings of the same class.

Implementation Details We implement our method on the *MMSegmentation*² codebase and inherit its training configurations. Input resolutions are set as 512×512 . When using ViT, the pre-trained positional embeddings adopt bicubic interpolation. MaskCLIP requires no training and we train MaskCLIP+ on 4 Tesla V100 GPUs with a batch size of 16. For annotation-free segmentation, we perform MaskCLIP-guided learning for 4k/8k iterations on PASCAL Context/COCO Stuff with DeepLabv2-ResNet101 as the backbone segmentor. Self-training is not used in this setting. For zero-shot segmentation, we choose the lightest training schedule provided by *MMSegmentation*, which is 20k/40k/80k for PASCAL VOC/PASCAL Context/COCO Stuff. The first 1/10 training iterations adopt MaskCLIP-guided learning and the rest adopts self-training. For fair comparisons, we choose DeepLabv2 as the target model for PASCAL VOC and COCO Stuff and DeepLabv3+ for PASCAL Context. All use the ResNet-101 backbone initialized with the ImageNet pre-trained weights. Finally, we use the publicly available CLIP-ResNet-50 and CLIP-ViT-B/16 models.³

11.4.1 Annotation-Free Segmentation

In this challenging setting, no annotation is provided during training. We first evaluate the mIoU performance on two standard datasets, PASCAL Context and COCO-Stuff. Then, we collect images from Flickr to show interesting qualitative results on novel concepts, such as *Batman* and *Joker*. Finally, we test the robustness of MaskCLIP under various image corruptions.

Performance on Standard Datasets In Table 11.1a, we show mean Intersection over Union (mIoU) results on PASCAL Context and COCO-Stuff. The baseline in

² <https://github.com/open-mmlab/mmdetection>.

³ <https://github.com/openai/CLIP>.

Table 11.1 Annotation-free segmentation (mIoU). (a) We evaluate the performance of MaskCLIP(+) on two standard datasets. For Pascal Context, we ignore the evaluation on the background class. The target model of MaskCLIP+ is Deeplabv2-ResNet101. KS and PD denote key smoothing and prompt denoising, respectively. And they are not used in MaskCLIP+. (b) We test the robustness of MaskCLIP on Pascal Context under various types of corruption. Table sourced from [55]

(a)				(b)				
Method	CLIP	PASCAL	COCO	Corruption	level 1		level 5	
		Context	Stuff		r50	vit16	r50	vit16
Baseline	r50	8.3	4.6	None	18.5	21.7	18.5	21.7
	vit16	9.0	4.3	Gaussian noise	13.7	19.6	2.1	6.8
MaskCLIP	r50	18.5	10.2	Shot noise	14.0	19.6	2.4	7.5
	+KS	21.0	12.4	Impulse noise	9.9	17.3	2.1	7.2
	+PD	19.0	10.8	Speckle noise	15.1	20.0	5.6	11.4
	+KS+PD	21.8	12.8	Gaussian blur	17.4	21.6	4.3	14.1
	vit16	21.7	12.5	Defocus blur	15.7	20.8	6.6	15.5
	+KS	23.9	13.8	Spatter	17.1	20.5	7.8	12.2
	+PD	23.1	13.2	JPEG	15.7	20.8	7.6	14.5
	+KS+PD	25.5	14.6					
MaskCLIP+	r50	23.9	13.6					
	vit16	31.1	18.0					

the table refers to directly using dense features from the CLIP’s image encoder without any modification. As shown in the table, MaskCLIP outperforms the baseline by huge margins, indicating it is essential to avoid computing attention of the last attention layer and instead value features should be directly used. The results also show that key smoothing and prompt denoising are effective and are orthogonal to each other. Therefore, we empirically conclude that for each spatial location, the query features should be discarded and key/value features can be reorganized into final predictions. Furthermore, with the predictions of MaskCLIP as pseudo-labels, MaskCLIP+ significantly improves the performance, e.g., on PASCAL Context, without any human annotation, MaskCLIP+(ViT-B/16) obtains 31.1 mIoU. One may notice that ViT almost consistently surpasses ResNet. Apart from ViT-B/16 has more FLOPs than ResNet-50, another possible reason is that ViT only downsamples the input by 16 times, whereas ResNet downsamples 32 times, which particularly matters for dense prediction tasks. Besides quantitative results, in Fig. 11.3, we also visualize the outputs of each MaskCLIP variant.

Open-Vocabulary Segmentation on Web-Crawled Images MaskCLIP inherits the open-vocabulary ability from CLIP and does not require annotations. Therefore, we can deploy it on several interesting setups where the target classes are (1) more fine-grained, such as *red car*, *yellow car*; (2) of certain imagery properties, e.g., blurry; (3) novel concepts like *Batman*, *Joker*. To this end, we collect images from Flickr and then directly evaluate these images on MaskCLIP and train MaskCLIP+ with only MaskCLIP-guided learning. Note that, for the background, we enumerate

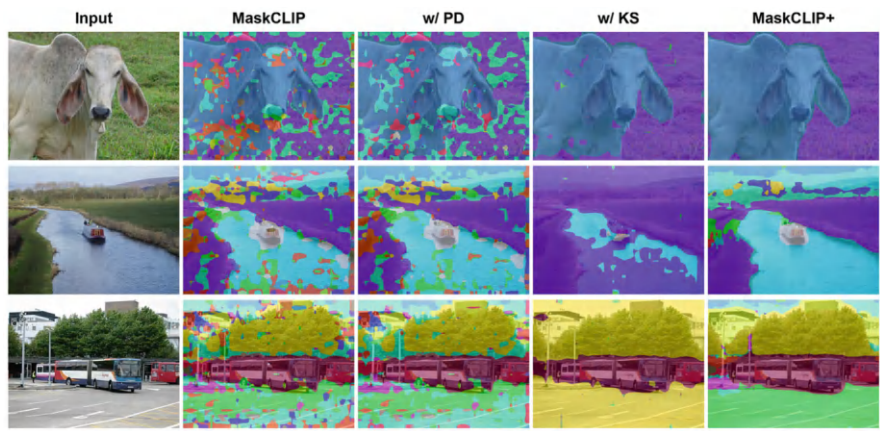


Fig. 11.3 Qualitative results on PASCAL Context. Here, all results are obtained **without** any annotation. PD and KS refer to prompt denoising and key smoothing, respectively. With PD, we can see some distraction classes are removed. KS is more aggressive. Its outputs are much less noisy but are dominated by a small number of classes. Finally, MaskCLIP+ yields the best results. Image sourced from [55]

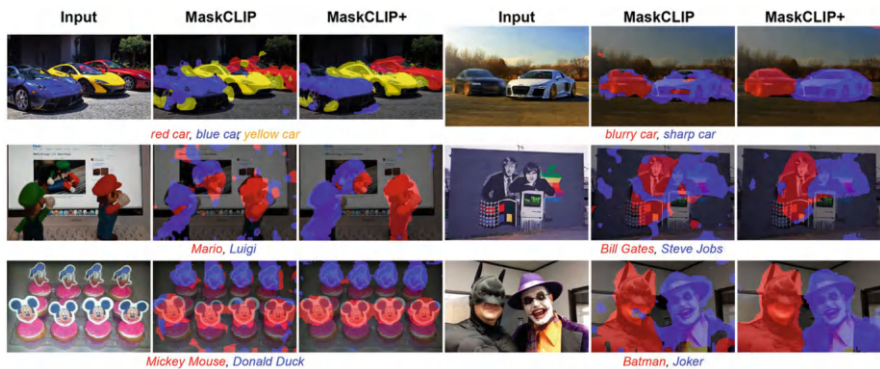


Fig. 11.4 Qualitative results on web images. Here, we show the segmentation results of MaskCLIP and MaskCLIP+ on various **unseen classes**, including fine-grained classes such as cars in different colors/imagery properties, celebrities, and animation characters. All results are obtained **without** any annotation. Image sourced from [55]

a set of classes that might appear in the background and regard them as a whole as the background class. Results in Fig. 11.4 are impressive given the open-vocabulary targets and being annotation-free. Besides, results from MaskCLIP+ are less noisy and more accurate than MaskCLIP, which is complementary to the quantitative results.

Robustness Under Corruption CLIP is trained on web-curated images, whose quality and distribution are more diverse than well-preprocessed datasets. Radford et

al. [44] and Ravula et al. [46] demonstrate the robustness of CLIP on natural distribution shift and artificial corruption, respectively. While these explorations are done for image classification, we benchmark its robustness for dense prediction tasks. Specifically, we impose various corruptions used in ImageNet-C [23] with different severity levels on images in PASCAL Context and evaluate on MaskCLIP. In Table 11.1b, MaskCLIP models based on CLIP-ViT-B/16 are much more robust than CLIP-ResNet-50. In particular, CLIP-ViT-B/16 rarely suffers from degradation across a wide range of corruptions with level 1 severity and is capable of generating reasonable labels even under the most severe corruptions (level 5⁴).

11.4.2 Zero-Shot Segmentation

Apart from annotation-free segmentation, MaskCLIP+ can also be applied to the zero-shot segmentation task with minor effort. Specifically, in the zero-shot setting, pixels of certain classes do not have annotations, to which MaskCLIP can assign reliable pseudo-labels.

Zero-Shot Setups Traditionally, zero-shot segmentation methods train on a subset of classes, named seen classes, with ground-truth annotations, and during inference, both seen and unseen classes are evaluated. Depending on whether the unlabeled pixels are observed during training, the setting can be split into inductive (not observed) and transductive (observed). Our method conforms to the transductive setting.

The selection of seen classes varies among previous works and we follow the most common setups, where for PASCAL VOC, the background class is ignored and *potted plant*, *sheep*, *sofa*, *train*, and *tv monitor* are chosen as the five unseen classes; for PASCAL Context, the background is not ignored and *cow*, *motorbike*, *sofa*, *cat*, *boat*, *fence*, *bird*, *tv monitor*, *keyboard*, *aeroplane* are unseen; and for COCO Stuff, *frisbee*, *skateboard*, *cardboard*, *carrot*, *scissors*, *suitcase*, *giraffe*, *cow*, *road*, *wall concrete*, *tree*, *grass*, *river*, *clouds*, and *playing field* are unseen. We report the mean Intersection over Union (mIoU) of seen, unseen, and all classes as well as the harmonic mean (hIoU) of seen and unseen mIoUs as evaluation metrics.

We compare MaskCLIP+ with SOTA methods including SPNet [49], ZS3Net [1], CaGNet [19], and STRICT [40]. ZS3Net and CaGNet are generative approaches, while SPNet is non-generative and more simple but requires post-possessing step of calibration (SPNet-C). STRICT improves SPNet by a self-training strategy and is free of calibration. Compared with these methods, our MaskCLIP+ does not rely on any particular network architecture nor post-possessing. Note that similar to CLIP, all methods, except for ZS3Net, do not exclude unseen classes during pre-training.

⁴ The severity level is controlled by certain coefficients, such as kernel size, specified in ImageNet-C [23].

Besides, MaskCLIP+ also follows the rule that pixel-level annotations of unseen classes are prohibited. Thus, the comparison is fair.

Despite being simple, MaskCLIP+ achieves a strikingly good result. As shown in Table 11.2, it surpasses all methods on all datasets with large margins. On PASCAL VOC, PASCAL Context, and COCO Stuff, in terms of unseen mIoUs, MaskCLIP+ improves the previous SOTA by 50.5, 24.4, and 46.0, respectively (on a scale of 100). Note that the overall mIoU of MaskCLIP+ is on par with that of fully supervised baselines. Please refer to Table 11.2 for more specific numbers.

Ablation Studies of MaskCLIP+ We perform ablation studies on the PASCAL VOC zero-shot segmentation setting. As shown in Table 11.3, we first examine the two discussed strategies in MaskCLIP+. Compared to the adapted DeepLabv2, whose classifier is replaced with the MaskCLIP classifier, MaskCLIP-guided learning improves the unseen mIoU from 3.7 to 72.8, and the result is further improved by self-training to 86.1. However, there is a slight degradation on seen classes when using self-training (from 89.5 to 88.8) partially due to model drifting. Overall, MaskCLIP+ performs better than MaskCLIP on unseen classes and surpasses the baseline DeepLabv2 on seen classes in the same time.

Table 11.2 Zero-shot segmentation performances. ST stands for self-training. mIoU(U) denotes mIoU of the unseen classes. SPNet-C is the SPNet with calibration. On PASCAL Context, all methods use DeepLabv3+-ResNet101 as the backbone segmentation model and the rest two datasets use DeepLabv2-ResNet101. For MaskCLIP+, CLIP-ResNet-50 is used to generate pseudo-labels. Table sourced from [55]

Method	PASCAL-VOC			COCO-Stuff			PASCAL-Context		
	mIoU(U)	mIoU	hIoU	mIoU(U)	mIoU	hIoU	mIoU(U)	mIoU	hIoU
Inductive									
SPNet	0.0	56.9	0.0	0.7	31.6	1.4	·	·	·
SPNet-C	15.6	63.2	26.1	8.7	32.8	14.0	·	·	·
ZS3Net	17.7	61.6	28.7	9.5	33.3	15.0	12.7	19.4	15.8
CaGNet	26.6	65.5	39.7	12.2	33.5	18.2	18.5	23.2	21.2
Transductive									
SPNet+ST	25.8	64.8	38.8	26.9	34.0	30.3	·	·	·
ZS3Net+ST	21.2	63.0	33.3	10.6	33.7	16.2	20.7	26.0	23.4
CaGNet+ST	30.3	65.8	43.7	13.4	33.7	19.5	·	·	·
STRICT	35.6	70.9	49.8	30.3	34.9	32.6	·	·	·
MaskCLIP+	86.1	88.1	87.4	54.7	39.6	45.0	66.7	48.1	53.3
	+50.5	+17.2	+37.6	+24.4	+4.7	+12.4	+46.0	+22.1	+29.9
Fully Sup.	·	88.2	·	·	39.9	·	·	48.2	·

Table 11.3 Ablations of MaskCLIP+. Experiments are performed on the PASCAL VOC dataset under the zero-shot setting. Table sourced from [55]

Method	mIoU(S)	mIoU(U)	mIoU	hIoU
Adapted DeepLabv2	83.4	3.7	63.5	7.0
+ MaskCLIP-Guided	89.5	72.8	85.3	80.3
+ Self-Training	88.8	86.1	88.1	87.4

11.5 Conclusion

This chapter presents our investigation into applying CLIP for semantic segmentation, marking an early exploration of how pre-trained vision-language models can be utilized for pixel-level dense prediction tasks. While traditional fine-tuning approaches fail to benefit from CLIP, we discover that the image encoder of CLIP inherently possesses the capability to function as a segmentation model without requiring additional training. The resulting model, MaskCLIP, can be seamlessly applied across various semantic segmentation tasks, maintaining its zero-shot transferability. Building on the success of MaskCLIP, we discuss MaskCLIP+, which utilizes MaskCLIP to generate high-quality pseudo-labels during training. This approach enables the use of more segmentation-tailored architectures beyond CLIP’s original image encoder, significantly broadening its applicability. On standard transductive zero-shot segmentation benchmarks, MaskCLIP+ achieves substantial improvements over previous state-of-the-art methods. More critically, its generality allows it to tackle challenging scenarios, such as segmenting unseen categories like celebrities and animated characters, demonstrating its versatility and robustness. This study highlights the potential of leveraging pre-trained vision-language models for dense prediction tasks and paves the way for further research into annotation-free and open-vocabulary segmentation methods.

References

1. Bucher, M., Vu, T.H., Cord, M., Pérez, P.: Zero-shot semantic segmentation. In: NeurIPS (2019)
2. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: thing and stuff classes in context. In: CVPR (2018)
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NeurIPS (2020)
4. Chen, L.C., Lopes, R.G., Cheng, B., Collins, M.D., Cubuk, E.D., Zoph, B., Adam, H., Shlens, J.: Naive-student: leveraging semi-supervised learning in video sequences for urban scene segmentation. In: ECCV (2020)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)

7. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: data collection and evaluation server. arXiv preprint (2015)
8. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR (2021)
9. Chen, X., Yuan, Y., Zeng, G., Wang, J.: Semi-supervised semantic segmentation with cross pseudo supervision. In: CVPR (2021)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
11. Desai, K., Johnson, J.: Virtex: Learning visual representations from textual annotations. In: CVPR (2021)
12. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015)
13. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: NeurIPS (2014)
14. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. *IJCV* **111**, 98–136 (2015)
15. Gomez, L., Patel, Y., Rusiñol, M., Karatzas, D., Jawahar, C.: Self-supervised learning of visual features through embedding images into text topic spaces. In: CVPR (2017)
16. Gordo, A., Larlus, D.: Beyond instance-level image retrieval: leveraging captions to learn a global visual representation for semantic retrieval. In: CVPR (2017)
17. Grill, J.B., Strub, F., Althé, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: a new approach to self-supervised learning. In: NeurIPS (2020)
18. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. arXiv preprint (2021)
19. Gu, Z., Zhou, S., Niu, L., Zhao, Z., Zhang, L.: Context-aware feature generation for zero-shot semantic segmentation. In: ACM MM (2020)
20. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV (2011)
21. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
23. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: ICLR (2019)
24. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: a reference-free evaluation metric for image captioning. In: EMNLP (2021)
25. Hu, P., Sclaroff, S., Saenko, K.: Uncertainty-aware learning for zero-shot semantic segmentation. In: NeurIPS (2020)
26. Hung, W.C., Tsai, Y.H., Liou, Y.T., Lin, Y.Y., Yang, M.H.: Adversarial learning for semi-supervised semantic segmentation. In: BMVC (2018)
27. Ibrahim, M.S., Vahdat, A., Ranjbar, M., Macready, W.G.: Semi-supervised semantic image segmentation with self-correcting networks. In: CVPR (2020)
28. Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: CVPR (2019)
29. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: semantically consistent few-shot view synthesis. In: ICCV (2021)
30. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)
31. Kato, N., Yamasaki, T., Aizawa, K.: Zero-shot semantic segmentation via variational mapping. In: ICCVW (2019)
32. Lee, D.H.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: ICMLW (2013)

33. Li, P., Wei, Y., Yang, Y.: Consistent structural relation learning for zero-shot segmentation. In: NeurIPS (2020)
34. Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.S., Schiele, B.: Learning to self-train for semi-supervised few-shot classification. In: NeurIPS (2019)
35. Mendel, R., De Souza, L.A., Rauber, D., Papa, J.P., Palm, C.: Semi-supervised segmentation based on error-correcting supervision. In: ECCV (2020)
36. Mittal, S., Tatarchenko, M., Brox, T.: Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE TPAMI* **43**(4), 1369–1379 (2019)
37. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR (2014)
38. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016)
39. Ouali, Y., Hudelot, C., Tami, M.: Semi-supervised semantic segmentation with cross-consistency training. In: CVPR (2020)
40. Pastore, G., Cermelli, F., Xian, Y., Mancini, M., Akata, Z., Caputo, B.: A closer look at self-training for zero-label semantic segmentation. In: CVPRW (2021)
41. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: text-driven manipulation of stylegan imagery. In: ICCV (2021)
42. Pham, H., Dai, Z., Xie, Q., Le, Q.V.: Meta pseudo labels. In: CVPR (2021)
43. Quattoni, A., Collins, M., Darrell, T.: Learning visual representations using images with captions. In: CVPR (2007)
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
45. Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., Zhou, J., Lu, J.: Denseclip: language-guided dense prediction with context-aware prompting. *arXiv preprint* (2021)
46. Ravula, S., Smyrnis, G., Jordan, M., Dimakis, A.G.: Inverse problems leveraging pre-trained contrastive representations. In: NeurIPS (2021)
47. Sariyildiz, M.B., Perez, J., Larlus, D.: Learning visual representations with caption annotations. In: ECCV (2020)
48. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
49. Xian, Y., Choudhury, S., He, Y., Schiele, B., Akata, Z.: Semantic projection network for zero-and few-label semantic segmentation. In: CVPR (2019)
50. Xie, J., Zheng, S.: Zsd-yolo: Zero-shot yolo detection using vision-language knowledge distillation. *arXiv preprint* (2021)
51. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: NeurIPS (2014)
52. Yuan, X., Lin, Z., Kuen, J., Zhang, J., Wang, Y., Maire, M., Kale, A., Faieta, B.: Multimodal contrastive training for visual representation learning. In: CVPR (2021)
53. Zhao, H., Puig, X., Zhou, B., Fidler, S., Torralba, A.: Open vocabulary scene parsing. In: ICCV (2017)
54. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
55. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: ECCV (2022)
56. Zou, Y., Zhang, Z., Zhang, H., Li, C.L., Bian, X., Huang, J.B., Pfister, T.: Pseudoseg: designing pseudo labels for semantic segmentation. In: ICLR (2021)

Chapter 12

Adapting CLIP for 3D Understanding



Xiangyang Zhu, Renrui Zhang, and Peng Gao

Abstract Contrastive Vision-Language Pre-training (CLIP) has shown ground-breaking results in zero-shot and few-shot learning. Its success in the 2D domain sparks the curiosity to explore if CLIP, pre-trained with large-scale image-text pairs, can be adapted for 3D recognition. This chapter discusses PointCLIP and PointCLIP V2 and shows that it is indeed possible to align CLIP-encoded point clouds with 3D category text representations. PointCLIP projects a point cloud into multi-view 2D images, which helps in revealing geometric information from 3D to 2D. For better extraction of global features, PointCLIP features an interview adapter, significantly boosting the performance by just fine-tuning the lightweight adapter. After that, PointCLIP V2 is discussed to further improve the performance and enhance the generalization ability. V2 employs a shape projection module for the visual end, helping to synthesize more realistic depth maps and narrowing the domain gap between projected point clouds and natural images. Simultaneously, V2 combines CLIP and large language models (LLMs) to generate 3D-specific text, improving the feature extraction of CLIP's textual encoder. PointCLIP V1 and V2 demonstrate remarkable generalization capabilities in unified 3D open-world learning, paving the way for better in-depth understanding and possible future improvements and extensions.

Keywords Vision-language model · 3D point clouds · 3D object recognition · Open-world learning

X. Zhu · P. Gao

Shanghai Artificial Intelligence Laboratory, Shanghai, China

e-mail: zhuxiangyang@pjlab.org.cn; gaopeng@pjlab.org.cn

R. Zhang (✉)

The Chinese University of Hong Kong, Hong Kong, China

e-mail: renruizhang@link.cuhk.edu.hk

12.1 Introduction

Deep learning has been pivotal in undertaking tasks in 3D computer vision field, such as classification [29, 41, 52], segmentation [30, 38, 44, 46], and detection [24, 47]. With the accelerated growth of 3D sensing technology, the demand for processing 3D point cloud data has stimulated the progress of advanced deep models with enhanced feature aggregators [20, 34] and projection-based processing [33]. Unlike structured 2D image data, 3D point clouds grapple with the challenge of space sparsity and random distribution, posing difficulty in directly using method transfers from the 2D domain. Added to that is the drawback of large-scale point cloud data containing numerous “unseen” category objects to the pre-trained classifier, which may lead to failures in object recognition.

The difficulty of identifying unseen categories in 2D vision has been significantly alleviated by pre-trained models like CLIP [31], which learns text-aligned visual features from large-scale web data. To categorize unseen objects, CLIP uses its learned correlation between vision and language to execute open-vocabulary recognition, contributing to impressive results. However, whether such CLIP-based models could be migrated to the 3D domain to enable unseen 3D object classification remains unexplored.

In this chapter, we discuss two frameworks, PointCLIP and PointCLIP V2, both of which transfer learned knowledge of CLIP from 2D pre-training to 3D point cloud understanding. We observe that the 2D knowledge from CLIP, guided by contrastive training, can complement 3D learning. PointCLIP first establishes a connection between grid-based images and unordered point clouds. Considering the necessity of real-time projections in fields such as autonomous driving [4, 19] and indoor navigation [57], PointCLIP suggests the use of online perspective projection [10] without any additional rendering [33]. This means projecting each point onto several predetermined image planes to create scatter depth maps. The projection process preserves the shape originality of point clouds from multiple angles, and its cost in terms of time and computation is negligible. Following the steps of CLIP, we build 3D category prompts with a handcrafted template. Using the pre-trained CLIP visual encoder, we extract multi-view point cloud features, and each view’s text-matching prediction is independently made through the similarity between the view feature and prompts’ features. Considering that different views play different roles in recognizing a whole object, we calculate the final prediction by using a weighted aggregation of view predictions.

Despite the ability to perform cross-modality zero-shot classification without 3D training, PointCLIP still lags behind traditional point cloud networks that have been well trained on comprehensive datasets. To overcome this issue, PointCLIP incorporates a learnable interview adapter with bottleneck linear layers to enhance feature extraction from multiple views in a few-shot setting. Concretely, we concatenate the features of all views and extract the compact global point cloud feature by interacting with cross-view information. Based on this global representation, the adapted feature is added to their original CLIP-encoded feature through a residual

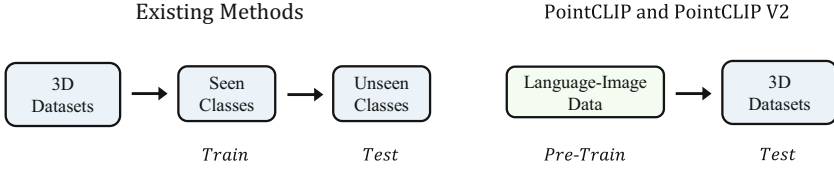


Fig. 12.1 Pipeline comparison between CLIP-based methods and other approaches. Other methods still depend on prerequisite 3D training to recognize the “unseen” point clouds. In PointCLIP and V2, the usage of CLIP eliminates the requirement of 3D pre-training

connection. Therefore, each view not only features the specific view angle but also merges the newly adapted feature from the 3D few-shot learning with the 2D pre-trained CLIP encoding. During training, we solely fine-tune this lightweight adapter while keeping both CLIP’s visual and textual encoders frozen to avoid overfitting, given the limited number of samples per class. Impressively, PointCLIP, equipped with an interview adapter and few-shot training, performs on par with preceding well-trained models on rich datasets, striking an ideal balance between performance and cost (Fig. 12.1).

PointCLIP achieves cross-modality zero-shot classification without any 3D training. Based on this, PointCLIP V2 makes improvements in two aspects. On the one hand, V2 introduces a new projection method to generate CLIP-friendly images from 3D point clouds, where the sparse projection of point clouds is replaced with more visually realistic depth maps to fully exploit the understanding capabilities of CLIP’s visual encoder. To do this, we convert the irregular point cloud into grid-based voxels and then apply nonparametric 3D local filtering. Consequently, the projected 3D shapes consist of denser points with smoother depth values, resulting in maps more closely resembling real-world images. On the other hand, we utilize LLMs, e.g., the GPT-3 model [2], to generate text with rich 3D semantics for CLIP’s textual encoder. By inputting heuristic 3D-oriented commands into GPT-3, we derive a series of 3D-specific texts possessing diverse 3D shape information, substantially enhancing CLIP’s image-text alignment for 3D point clouds.

With both prompting from visual and textual perspectives, V2 exhibits promising performance for zero-shot 3D classification. In addition, V2 introduces a trainable 3D smoothing process to better extract features from multiple views in a few-shot setting. The zero-shot framework equipped with these learnable modules can be fine-tuned with very limited training samples. Moreover, V2 can be adjusted for more complex 3D open-world tasks with minimal alterations. These include the addition of a back-projection block for zero-shot segmentation and a 3D region proposal network (RPN) for zero-shot detection. The success of these tasks fully demonstrates the capability of the discussed model for broad 3D open-world understanding.

12.2 Related Work

In this section, we first review the basic knowledge of CLIP, and then, we discuss classic deep learning models developed for point cloud analysis, as well as 3D open-world understanding.

12.2.1 Revisiting CLIP

CLIP [31] is trained to align the representation of images and corresponding textual descriptions. It contains two independent encoders: one for visual data and the other for textual input. In the learning stage, CLIP encodes texts and images, extracts global features, and aligns them in the embedding space with a contrastive loss. A collection of 400 million image-text pairs from online resources is used as training data. This allows CLIP to analyze images with a wide range of semantic concepts and exhibit the potential for identifying categories from an open vocabulary.

Specifically, when dealing with a random dataset with K classes, CLIP organizes the textual inputs by injecting all class names into a predefined template, also referred to as a prompt. The zero-shot classifier, denoted as $W_t \in \mathbb{R}^{K \times C}$, is yielded from the C -dimensional textual feature of K category prompts. Every row vector in W_t represents the preestablished category weights. During inference, the test image feature, $f \in \mathbb{R}^{1 \times C}$, encoded by CLIP's visual encoder is matched with K class features. The predicted classification logits can be calculated as

$$\text{logits} = f W_t^\top. \quad (12.1)$$

The whole process does not require new training samples but achieves promising zero-shot classification performance only by frozen pre-trained encoders.

12.2.2 Deep Learning for Point Clouds

Current deep learning networks for point cloud data can be categorized into point-based and projection-based methods. Point-based models function directly on raw points without any prior transformation. Early models like PointNet [29] and PointNet++ [30] encode points using multilayer perceptron (MLP) layers and apply a max pooling operation for permutation invariance. After that, point-based methods introduce advanced local aggregators and architectural designs [20, 34, 38, 46]. Some efforts also apply transformers [36] for point cloud analysis and achieve satisfactory results [12, 42, 43, 53]. Projection-based methods, on the other hand, process point clouds by converting them into grid-based multi-view or volumetric data. Multi-view methods project point clouds into images from various perspectives

[1, 32, 37, 40, 50] and then process them using 2D convolution neural networks (CNN) that are pre-trained on ImageNet [17], like MVCNN [33]. Typically, these methods rely on offline-generated images that are projected from point-converted 3D meshes or require post-rendering for shades and textures, making them time-consuming and unsuitable for real-time applications. Contrarily, we simply project raw points onto image planes and set their pixel values based on the vertical distance. The generated depth maps require minimal time and computational resources, facilitating efficient end-to-end zero-shot recognition.

12.2.3 3D Open-World Learning

Traditional approaches to 3D open-world learning still necessitate 3D data during the pre-training phase. A series of studies by Cheraghian et al. pre-train zero-shot classifiers on “seen” categories by maximizing interclass divergence in latent space, before testing on “unseen” categories [5–7]. Existing investigations have also delved into open-world semantic segmentation and 3D object detection for more intricate 3D scenes [21, 22, 25, 27]. CLIP2Point enhances the adaptation performance of CLIP on point clouds with additional 3D pre-training [16]. In this chapter, we conduct experiments under open-world settings, which is more challenging. Without any 3D training, we investigate zero-shot 3D classification, part segmentation, and object detection tasks and achieve a comprehensive understanding of the 3D open world.

12.3 PointCLIP for 3D Understanding

Large-scale 2D datasets provide abundant samples to effectively pre-train models for high-quality and robust feature extraction. However, most existing 3D datasets are relatively small with limited categories, making it challenging to acquire powerful pre-trained 3D networks for transfer learning. To address this issue, PointCLIP utilizes a simple projection manner to bridge the modality gap and fully leverage the pre-trained knowledge of CLIP. The whole framework is presented in Fig. 12.2.

12.3.1 Bridging the Modality Gap

Point clouds could be described as points scattered in 3D space, whose sparsity and distribution significantly differ from grid-based 2D images. To transform point clouds into 2D images, we employ multiple-view projections to bridge the gap between 3D and 2D modalities. Specifically, given a point’s coordinates in 3D

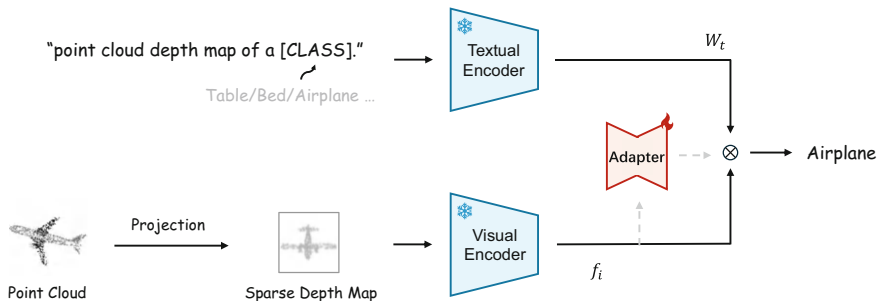


Fig. 12.2 The whole framework of PointCLIP, where CLIP [31] is used as the backbone. The point cloud is projected into sparse depth maps to bridge the modality gap



Fig. 12.3 Examples of projection views. Top: Sparse projections used in PointCLIP. Bottom: Corresponding realistic depth maps generated in PointCLIP V2

space denoted as (x, y, z) , the point's location on a 2D image plane—taking the bottom projection view as an example—would be $(\lceil x/z \rceil, \lceil y/z \rceil)$, following the approach mentioned in SimpleView [10]. Examples can be found in Fig. 12.3. This method results in projections that appear foreshortened, meaning that they appear smaller from a distance and larger when closer, mirroring the appearance one typically observes in real photos. Unlike the previous approach [10] which applies convolution layers to preprocess one-channel depth maps into three, PointCLIP skips convolution and directly sets the pixel value identical to “ z ” across all three channels. Contrary to other offline projection techniques, which generate projected images from meshes or CAD models [33], PointCLIP's scatter projections derive directly from raw points, displaying depth values devoid of color information. This method minimizes time and computation costs. By utilizing this efficient cross-modality integration, we can harness CLIP's pre-trained knowledge for point cloud understanding.

12.3.2 Zero-Shot Classification

Using projected images from M views, we employ the CLIP model to extract their visual features, denoted as $\{f_i\}_{i=1}^M$. For the textual input, we insert K category

names into the class token position of a predetermined template: “point cloud depth map of a [CLASS].” We encode these textual features to form the zero-shot classifier $W_t \in \mathbb{R}^{K \times C}$. Additionally, we calculate the classification logits for each view individually and derive the final logits for the point cloud through their weighted summation:

$$\text{logits} = \sum_{i=1}^M \alpha_i f_i W_t^\top, \quad (12.2)$$

where α_i is a hyperparameter that assigns the level of importance to view i . Each view f_i encodes individual point cloud features from a distinct perspective, which enables independent zero-shot classification. The summation of these views further consolidates information from different angles to generate a comprehensive understanding. The entire process does not require parameterization for the “unseen” 3D dataset. This pairs each point cloud with its corresponding category by leveraging the pre-trained 2D knowledge of the CLIP model, which eliminates the requirement for 3D training.

12.3.3 The Interview Adapter

The zero-shot scheme falls short compared to fully trained 3D neural networks. Therefore, we further consider the situation where “unseen” categories are represented by a few objects in freshly collected data, and our task is to identify them within these modest settings. Attempting to fine-tune the entire model is impractical due to the risk of overfitting caused by a lack of training samples along with an overwhelming number of parameters. Therefore, PointCLIP adds a three-layer MLP on top of CLIP. This additional component, called the interview adapter, is designed to effectively bolster performance in few-shot settings. For the training process, we freeze CLIP’s visual and textual encoders and fine-tune the adaptable component through cross-entropy loss. Specifically, we integrate M -view features of a point cloud encoded by CLIP across the channel dimension. This allows us to obtain the point cloud’s compact global feature via the first two layers of the interview adapter. Once this interview aggregation occurs, the features from multiple viewpoints are combined into a comprehensive representation. Subsequently, the adjusted feature is generated from the global feature and then merged with its initial CLIP-encoded feature by way of a residual connection. This allows for a comprehensive understanding of the point cloud and, subsequently, an improved prediction.

Remarkably, merely refining this lightweight adapter with few-shot samples leads to a remarkable performance improvement, for instance, enhancing the classification accuracy from 20.18% to 87.20% on the commonly used ModelNet40 [45] dataset with 16 samples available per category. This significant advancement underlines the effectiveness and crucial role of feature adaptation on 3D few-shot data, thereby facilitating the transition of knowledge from 2D to 3D. As a result,

PointCLIP appended with an interview adapter showcases a promising solution for point cloud understanding. In certain applications, whole model training might be unfeasible due to a lack of fully annotated data, and here, merely fine-tuning the lightweight adapter with few-shot data yields a comparable performance with fully trained models.

12.4 PointCLIP V2 for 3D Open-World Understanding

After PointCLIP, PointCLIP V2 extends CLIP to more 3D tasks, including classification, part segmentation, and detection. Compared to V1, V2 not only prompts CLIP with realistic depth maps but also prompts GPT-3 [2] with 3D-specific commands to ensure congruity between depth maps and languages. V2 holds the potential for extensive 3D tasks and unified 3D open-world learning. Figure 12.4 illustrates the whole framework of PointCLIP V2.

12.4.1 Prompting CLIP with Realistic Projection

To create a more realistic 2D representation from 3D data for CLIP while also optimizing time efficiency, we translate 3D point clouds into depth maps through four steps: quantize, densify, smooth, and squeeze.

Quantize We build a zero-initialized 3D grid $G \in \mathbb{R}^{H \times W \times D}$ for different M views to be projected, where H , W , and D represent its spatial resolutions with D exceptionally denoting the depth dimension perpendicular to the image plane. Taking a single view as an example, we standardize the 3D coordinates of the

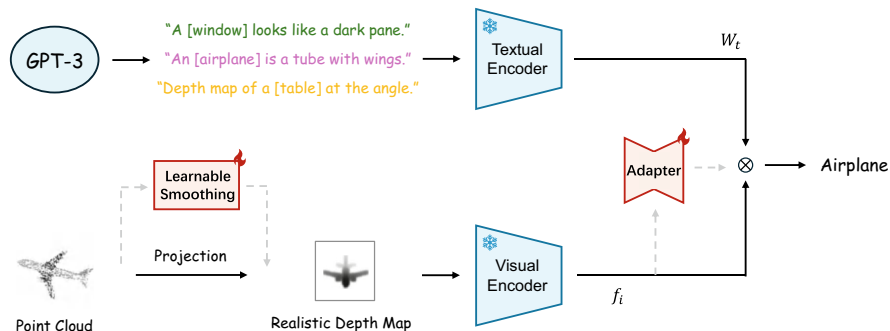


Fig. 12.4 The framework of PointCLIP V2. We first generate high-quality depth maps via a realistic projection to prompt CLIP's [31] visual encoder. Then, we design 3D language commands to prompt GPT-3 [2] for 3D-specific text into CLIP's textual encoder

inputted point cloud into $[0, 1]$ and arrange a point $p = (x, y, z)$ into a cell in the grid as $G(\lceil sHx \rceil, \lceil sWy \rceil, \lceil Dz \rceil) = z$. Here, the cells are assigned varying depth values, and $s \in (0, 1]$ stands for a scale factor modifying the projected shape size. In scenarios where multiple points are projected into the same cell, the cell is simply designated with the smallest depth value. This is because in the context of the target view plane, points with smaller depth value z tend to occlude the larger ones. Consequent to this, we acquire a 3D grid G encompassing sparse depth values, with the majority of cells empty due to the sparsity of point clouds.

Densify To address this sparsity, we enlarge the density of the grid through local minimum pooling operations to ensure visual consistency. Each voxel in G is reassigned by the least voxel value within a local spatial window. Similarly, compared with average and maximum pooling, maintaining the smallest depth values aligns with visually occluded appearances on the projected maps. In this manner, previously unoccupied voxels between sparse points can effectively be populated with logical depth values, and background voxels remain empty, leading to dense and solid spatial shapes.

Smooth The local pooling operation may result in artifacts on certain 3D surfaces. This necessitates the use of a nonparametric Gaussian kernel for shape smoothing and denoising. With suitable kernel size and variance, the filtering approach can eliminate the spatial disruptions caused by densification yet retain the sharpness of edges and corners in original 3D forms. This step results in a more consolidated and realistic shape contained in the 3D grid.

Squeeze Lastly, we concisely squeeze the depth dimension of G to attain the final projected depth map $V \in \mathbb{R}^{H \times W}$. We employ the least values of each depth channel as the value for every pixel location and replicate it three times to simulate RGB intensity. Our grid-centered projection can simply be enacted via minimum pooling along G 's depth channel, proving advantageous for hardware application.

After the four steps, a point cloud can be projected into a realistic depth map. We synthesize projections from M different view angles like in PointCLIP. Examples of generated depth maps are presented in Fig. 12.3.

12.4.2 Prompting GPT with 3D Command

To better align CLIP's textual encoder with our depth maps, we use 3D-specific descriptions which contain rich and vivid shape details as the textual input for CLIP, instead of the generic "a photo of a [CLASS]." Given the robust descriptive capabilities of LLMs, we harness GPT-3 [2] to generate 3D-specific text with ample 3D semantics for the textual encoder in CLIP. Typically, GPT-3 processes a language command and delivers a response leveraging pre-trained knowledge. To fully acclimate GPT-3 to 3D domains, we suggest the following four series of heuristic commands:

Caption Generation Under a descriptive command, GPT-3 creates general captions for the target projected depth map. For example, Input: “Describe a depth map of a [window]”; GPT-3: “It portrays the [window] as a dark pane.”

Question Answering GPT-3 yields descriptive answers to 3D-related questions. For example, Input: “How to describe a depth map of a [table]?” GPT-3: “The [table] possesses a rectangular or circular flat top with legs.”

Paraphrase Generation GPT-3 is commanded to generate a synonymous phrase for a depth map description. For example, Input: “Formulate a synonym for the sentence: A grayscale depth map of an inclined [bed].” GPT-3: “A monochromatic depth map of a slanted [bed].”

Words to Sentence Based on a set of keywords, GPT-3 is tasked to weave them into a comprehensive sentence and embellish it with additional shape-related details. For example, Input: “Compose a sentence using these words: a [table], depth map, smooth.” GPT-3: “This sleek depth map illustrates a [table] at the angle.” The adjective “smooth” here characterizes the natural look triggered by the smoothing operation.

For a K -category 3D dataset, we allocate K category names at each command’s “[CLASS]” location and feed them into GPT-3, subsequently generating 3D-specific descriptions rich in category-specific semantics. Eventually, we integrate the descriptions of each category and treat them as input for the textual encoder in CLIP. From Fig. 12.5, we observe that our realistic depth maps and the generated 3D prompts achieve better alignment.

12.4.3 3D Open-World Learning

Thanks to the realistic projection and 3D-specific description, PointCLIP V2 not only achieves superior performance on 3D zero-shot classification task but also

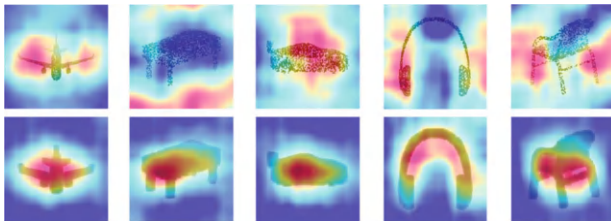


Fig. 12.5 Similarity maps between projections and textual prompts. We calculate and visualize the cosine similarity between fine-grained visual features and textual features. Top: Similarity maps of PointCLIP. Bottom: Similarity maps of PointCLIP V2

proves robust generalization capacity and can be tailored for various 3D open-world tasks.

12.4.3.1 3D Few-Shot Classification

With a small set of point cloud samples, PointCLIP V2 can be adapted to be a few-shot learner via two alterations. On the one hand, we can adapt the realistic projection pipeline into a learnable module. Specifically, irregular point clouds, once reshaped into grid-based voxels, are put through two 3D convolutional layers following the Gaussian filter. In this way, the projection module learns to generate CLIP-preferred depth maps. On the other hand, we also inherit the interview adapter used in PointCLIP. Both learnable modules capture 3D geometric insights from the training samples, thus improving their compatibility with CLIP.

12.4.3.2 3D Zero-Shot Part Segmentation

Beyond classification, V2 also investigates the zero-shot part segmentation task. Instead of global representation $\{f_i\}_{i=1}^M$, we utilize CLIP’s visual encoder to extract dense features $\{F_i\}_{i=1}^M$ from different projection views, where $F_i \in \mathbb{R}^{H \times W \times C}$. Specifically, we output the feature maps from the visual encoder before its final pooling operation and upsample the features into the original depth map size. For our 3D-specific text, we utilize GPT-3 to generate the descriptions for different part categories. As an example, for a part category “[PART]” within object “[CLASS],” we construct the command as “Describe the [PART] part of a [CLASS] in a depth map:”. Then, for view i , we conduct dense alignment between each pixel and the textual feature W_i , i.e., segmenting different parts of the shape on multi-view depth maps, formulated as

$$\text{logits}_i = F_i W_i^\top \in \mathbb{R}^{H \times W \times K}. \quad (12.3)$$

Each entry in logits_i denotes pixel-specific classification logits which are then back-projected into the 3D space following the 2D-3D mapping. Considering occlusion often causes only partial points visible in one view, we average the prediction across various views for each point. Through geometric back projection, we can accomplish the segmentation task in a zero-shot manner in the 3D space.

12.4.3.3 3D Zero-Shot Object Detection

For 3D object detection, we follow the technique of 2D open-world detection [11, 54], functioning as a zero-shot classification head atop a pre-trained region proposal network (RPN). We first engage 3DETR [26] as the 3D RPN to generate a series of 3D bounding boxes. Subsequently, we extract the raw points within each box and

input them into the classification pipeline. In this way, the CLIP-enhanced 3DETR can detect unrecognized objects without explicit training.

12.5 Experiments

In this section, we detail the evaluation benchmarks and highlight the performance of various 3D open-world tasks.

12.5.1 Zero-Shot Classification

Settings We assess zero-shot classification performance utilizing three widely used benchmarks: ModelNet10 [45], ModelNet40 [45], and ScanObjectNN [35]. We analyze three divisions of the ScanObjectNN dataset: OBJ_ONLY, OBJ_BG, and PB_T50_RS. Adhering to the zero-shot settings, we directly evaluate the classification results on the test set of each benchmark, without any learning from the training set. We assess existing methods under their optimal settings. Specifically, ViT-B/16 [9] applies to both PointCLIP V2 and CLIP2Point [16]. For PointCLIP, we employ ResNet-101 [15], ResNet-50-4 [31], and ViT-B/16 for ModelNet10, ModelNet40, and ScanObjectNN datasets to maximize its performance, respectively.

Results Table 12.1 presents the zero-shot classification performance in comparison with existing methods. Certain models necessitate additional pre-training on 3D point cloud datasets. CLIP2Point trains a depth map encoder on ShapeNet dataset [3] before applying it for the 3D zero-shot classification task. Cheraghian et al. [6], on the other hand, directly extract point cloud features with a 3D encoder, using “seen” categories in the dataset for model pre-training while testing on the “unseen” categories. In contrast, PointCLIP and V2 eliminate any 3D training and can directly conduct inference. We observe that both PointCLIP and V2 achieve satisfactory results. For all three benchmarks, V2 notably surpasses existing works, achieving 73.13% and 64.22% accuracy on ModelNet10 and ModelNet40, respectively. V2 also achieves 35.36% on the ScanObjectNN dataset’s PB_T50_RS split, proving its efficacy in real-world noisy scenes.

Ablation Study Table 12.2 presents our ablation results with regard to the four steps of the realistic projection module. When directly projecting the point cloud into 2D images using orthogonal projection, the zero-shot accuracy falls by -6.87% to 57.35% . If the quantization step is introduced, both the densifying and smoothing operations can boost zero-shot performance by $+15.14\%$ and $+5.7\%$, respectively, highlighting their importance. We also compare alternative pooling operations for the densifying step, such as maximum, minimum, and average pooling. The minimum pooling yields optimal results, in line with occlusion effects in the real world. In Table 12.3, we show the effect of four command types in the GPT

Table 12.1 Zero-shot 3D classification (%) of ModelNet10 [45], ModelNet40 [45] and ScanObjectNN [35]. We report the performance of other methods with their best-performing settings, e.g., visual encoder, projected view number, and textual input. “2D Pretrain” denotes the pre-training of CLIP on image-language pairs, and “3D Pretrain” denotes the training on 3D datasets. Table adapted from [56]

Method	2D Pretrain	3D Pretrain	ModelNet10	ModelNet40	OBJ_ONLY	OBJ_BG	PB_T50_RS
CLIP2Point [16]	✓	✓	66.63	49.38	35.46	30.46	23.32
Cheraghian [6]	-	✓	68.50	-	-	-	-
PointCLIP	✓	-	30.23	23.78	21.34	19.28	15.38
PointCLIP V2	✓	-	73.13	64.22	50.09	41.22	35.36

Table 12.2 Ablation study of realistic shape projection on ModelNet40 [45] zero-shot classification (%). We compare the four steps in our projection module. Table adapted from [56]

Quantize	Densify	Smooth	Squeeze	Zero-shot
–	Min	✓	–	57.35
✓	–	–	✓	44.50
✓	Min	–	✓	59.64
✓	–	✓	✓	50.20
✓	Max	✓	✓	57.35
✓	Avg	✓	✓	60.71
✓	Min	✓	✓	64.22

Table 12.3 Ablation study of GPT prompting on ModelNet40 [45] classification (%). We compare four types of commands to generate the 3D-specific text. Table adapted from [56]

Caption	Question	Paraphrase	Words	Zero-shot
–	–	–	–	39.11
✓	–	–	–	61.67
✓	✓	–	–	60.86
✓	–	✓	–	61.12
✓	✓	✓	–	63.29
✓	✓	–	✓	61.26
✓	✓	✓	✓	64.22

prompting module. With all four types of commands, the 3D-specific text improves the zero-shot performance by + 25.11%, indicating the significance of better language-image alignment.

12.5.2 Few-Shot Classification

Settings We investigate k -shot classification results on ModelNet40 [45] and ScanObjectNN [35] datasets, where $k \in \{1, 2, 4, 8, 16\}$. We employ the same 3D-specific text from the zero-shot task as the textual input. For PointCLIP, we only train the interview adapter. For V2, we jointly train the learnable smoothing and interview adapter.

Results Figure 12.6 illustrates the few-shot classification results. We compare PointCLIP and V2 to four other 3D models: PointNet [29], PointNet++ [30], SimpleView [10], and CurveNet [46]. Notably, both PointCLIP V1 and V2 outperform all other methodologies and show a marked improvement in one-shot classification, which validates the efficacy of V1 and V2. Furthermore, V1 and V2 attain a 16-shot accuracy of higher than 87.60% on ModelNet40, nearing the fully supervised results.

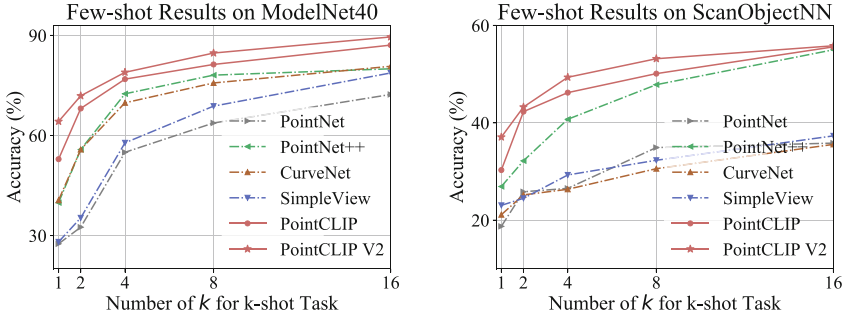


Fig. 12.6 Few-shot 3D classification on ModelNet40 [45] and ScanObjectNN [35]. We adopt the PB_T50_RS split of ScanObjectNN for comparison

12.5.3 Zero-Shot Part Segmentation

Settings Our evaluation of zero-shot part segmentation utilizes the ShapeNetPart dataset [51], which possesses 16 classes and 50 corresponding parts. Adhering to previous evaluation settings [23, 30, 38], we select 2048 data points from every point cloud of the test set. To enable a comparison, we list the highest-performing results.

Results The Mean Intersection of Union score per instance ($mIoU_I$) of V1 and V2 is exhibited in Table 12.4. We observe that PointCLIP achieves satisfactory results and V2 yields consistently superior results across diverse object categories, suggesting its capacity to analyze intricate 3D geometrics in a zero-shot manner.

12.5.4 3D Object Detection

Settings We evaluate the detection performance on ScanNet V2 dataset [8], which contains 18 categories. To generate bounding boxes, a pre-trained 3DETR-m [26] model is utilized. Then, we extract 1024 points within each 3D box. We report the zero-shot detection performance on the validation set using mean average precision (mAP) at two different IoU thresholds of 0.25 and 0.5, denoted as AP_{25} and AP_{50} .

Results Table 12.5 shows the 3D detection results of PointCLIP and V2. This verifies that both V1 and V2 have the potential to recognize 3D open-world objects in real-world scenes

Table 12.4 Zero-shot part segmentation (%) on ShapeNetPart [51]. Besides overall mIoU_I, we provide the class-wise mIoU_I of 11 categories, respectively

	mIoU _I	Airplane	Bag	Cap	Chair	Earphone	Guitar	Knife	Laptop	Mug	Rocket	Skate
# Shapes	2874	341	14	11	704	14	159	80	83	38	12	31
PointCLIP	31.0	22.0	44.8	13.4	18.7	28.3	22.7	24.8	22.9	48.6	22.7	42.7
PointCLIP V2	49.5	33.5	60.4	52.8	51.5	56.5	71.5	66.7	61.6	48.0	49.6	43.9

Table 12.5 3D object detection results (%) on ScanNet V2 [8]. Besides the mAP, the average precisions of 11 categories are shown. Table adapted from [56]

	Method	Mean	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Counter	Desk	Sink
AP ₂₅	PointCLIP	6.00	3.99	4.82	45.16	4.82	7.36	4.62	2.19	1.02	4.00	13.40
	PointCLIP V2	18.97	19.32	20.98	61.89	15.55	23.78	13.22	17.42	12.43	21.43	14.54
AP ₅₀	PointCLIP	4.76	1.67	4.33	39.53	3.65	5.97	2.61	0.52	0.42	2.45	5.27
	PointCLIP V2	11.53	10.43	13.54	41.23	6.60	15.21	6.23	11.35	6.23	10.84	11.43

12.6 Conclusion

In this chapter, we discuss PointCLIP and PointCLIP V2 which effectively transfer CLIP's pre-trained 2D knowledge to the 3D domain for cross-modality zero-shot recognition and other challenging tasks without any 3D training. These are achieved through multi-view projection and fine-tuned lightweight interview adapter designed to aggregate multi-view representations. PointCLIP first applies CLIP to conduct point cloud analysis and achieve satisfactory results. Based on this, PointCLIP V2 builds a powerful 3D open-world learner. V2 designs a realistic projection module for high-quality depth map synthesis and prompts the GPT-3 model for 3D-specific descriptions, which can achieve better visual-language alignment. The exploration of other 3D open-world tasks forms part of our future objectives, alongside the aim of developing 3D large models for wider practical applications such as outdoor 3D detection and visual grounding.

Further Reading

Interested readers may refer to the following material:

- [14, 48, 49] for 3D pre-training methods guided by the aligned multi-modality embedding space.
- [13, 18] for a comprehensive study into the deep learning-based point cloud analysis methods and tasks.
- [55] for large-scale 3D representation learning, which builds a unified and scalable 3D pre-training framework.
- [39] for point-to-pixel prompting approach to transfer pre-trained image models for point cloud analysis.
- [28] for prompt engineering via GPT-3.

References

1. Ahmed, S.M., Liang, P., Chew, C.M.: EPN: Edge-aware PointNet for object recognition from multi-view 2.5D point clouds. In: IEEE International Conference on Intelligent Robots and Systems, pp. 3445–3450 (2019)
2. 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. Proces. Syst.* **33**, 1877–1901 (2020)
3. 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* (2015)
4. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3D object detection network for autonomous driving. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1907–1915 (2017)

5. Cheraghian, A., Rahman, S., Campbell, D., Petersson, L.: Mitigating the hubness problem for zero-shot learning of 3D objects. *arXiv preprint arXiv:1907.06371* (2019)
6. Cheraghian, A., Rahman, S., Chowdhury, T.F., Campbell, D., Petersson, L.: Zero-shot learning on 3d point cloud objects and beyond. *Int. J. Comput. Vis.* **130**(10), 2364–2384 (2022)
7. Cheraghian, A., Rahman, S., Petersson, L.: Zero-shot learning of 3D point cloud objects. In: *IEEE International Conference on Machine Vision Applications*, pp. 1–6 (2019)
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: richly-annotated 3D reconstructions of indoor scenes. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839 (2017)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020)
10. Goyal, A., Law, H., Liu, B., Newell, A., Deng, J.: Revisiting point cloud shape classification with a simple and effective baseline. In: *International Conference on Machine Learning* (2021)
11. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921* (2021)
12. 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)
13. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M.: Deep learning for 3D point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(12), 4338–4364 (2020)
14. Guo, Z., Zhang, R., Zhu, X., Tang, Y., Ma, X., Han, J., Chen, K., Gao, P., Li, X., Li, H., et al.: Point-bind & point-llm: aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615* (2023)
15. 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)
16. Huang, T., Dong, B., Yang, Y., Huang, X., Lau, R.W., Ouyang, W., Zuo, W.: CLIP2Point: transfer CLIP to point cloud classification with image-depth pre-training. *arXiv preprint arXiv:2210.01055* (2022)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Proces. Syst.* **25** (2012)
18. Lahoud, J., Cao, J., Khan, F.S., Cholakkal, H., Anwer, R.M., Khan, S., Yang, M.H.: 3d Vision with Transformers: A Survey (2022)
19. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: fast encoders for object detection from point clouds. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705 (2019)
20. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: convolution on x-transformed points. *Adv. Neural Inf. Proces. Syst.* **31** (2018)
21. Liu, B., Deng, S., Dong, Q., Hu, Z.: Segmenting 3D hybrid scenes via zero-shot learning. *arXiv preprint arXiv:2107.00430* (2021)
22. Lu, Y., Xu, C., Wei, X., Xie, X., Tomizuka, M., Keutzer, K., Zhang, S.: Open-vocabulary 3D detection via image-level class and debiased cross-modal contrastive learning (2022)
23. Ma, X., Qin, C., You, H., Ran, H., Fu, Y.: Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123* (2022)
24. Meng, Q., Wang, W., Zhou, T., Shen, J., Jia, Y., Van Gool, L.: Towards a weakly supervised framework for 3D point cloud object detection and annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(8), 4454–4468 (2021)
25. Michele, B., Boulch, A., Puy, G., Bucher, M., Marlet, R.: Generative zero-shot learning for semantic segmentation of 3D point clouds. In: *IEEE International Conference on 3D Vision*, pp. 992–1002 (2021)
26. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3d object detection. In: *International Conference on Computer Vision*, pp. 2906–2917 (2021)

27. Naeem, M.F., Örnek, E.P., Xian, Y., Van Gool, L., Tombari, F.: 3D compositional zero-shot learning with decompositional consensus. In: European Conference on Computer Vision, pp. 713–730 (2022)
28. Pratt, S., Covert, I., Liu, R., Farhadi, A.: What does a platypus look like? generating customized prompts for zero-shot image classification. In: IEEE International Conference on Computer Vision, pp. 15691–15701 (2023)
29. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3D classification and segmentation. arXiv preprint arXiv:1612.00593 (2016)
30. 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. Proces. Syst.* **30** (2017)
31. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021)
32. Shi, B., Bai, S., Zhou, Z., Bai, X.: DeepPano: deep panoramic representation for 3-D shape recognition. *IEEE Signal Process Lett.* **22**(12), 2339–2343 (2015)
33. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3D shape recognition. In: International Conference on Computer Vision, pp. 945–953 (2015)
34. 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)
35. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: Revisiting point cloud classification: a new benchmark dataset and classification model on real-world data. In: International Conference on Computer Vision, pp. 1588–1597 (2019)
36. 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. Proces. Syst.* **30** (2017)
37. Wang, C., Pelillo, M., Siddiqi, K.: Dominant set clustering and pooling for multi-view 3D object recognition. arXiv preprint arXiv:1906.01592 (2019)
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 Trans. Graph.* **38**(5), 1–12 (2019)
39. Wang, Z., Yu, X., Rao, Y., Zhou, J., Lu, J.: P2P: tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. arXiv preprint arXiv:2208.02812 (2022)
40. Wei, X., Yu, R., Sun, J.: View-GCN: view-based graph convolutional network for 3D shape analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1850–1859 (2020)
41. Wu, W., Qi, Z., Fuxin, L.: PointConv: deep convolutional networks on 3D point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 9621–9630 (2019)
42. Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: simpler faster stronger. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4840–4851 (2024)
43. Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H.: Point transformer v2: grouped vector attention and partition-based pooling. In: NeurIPS (2022)
44. Wu, Y., Cheng, X., Zhang, R., Cheng, Z., Zhang, J.: Eda: explicit text-decoupling and dense alignment for 3d visual and language learning. arXiv preprint arXiv:2209.14941 (2022)
45. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: a deep representation for volumetric shapes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920 (2015)
46. Xiang, T., Zhang, C., Song, Y., Yu, J., Cai, W.: Walk in the cloud: learning curves for point clouds shape analysis. In: International Conference on Computer Vision, pp. 915–924 (2021)
47. Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: SqueezeSegV3: spatially-adaptive convolution for efficient point-cloud segmentation. In: European Conference on Computer Vision, pp. 1–19 (2020)

48. Xue, L., Gao, M., Xing, C., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S.: Ulip: Learning unified representation of language, image and point cloud for 3d understanding. *arXiv preprint arXiv:2212.05171* (2022)
49. Xue, L., Yu, N., Zhang, S., Li, J., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S.: Ulip-2: towards scalable multimodal pre-training for 3d understanding (2023)
50. Yang, Z., Wang, L.: Learning relationships for multi-view 3D object recognition. In: *International Conference on Computer Vision*, pp. 7505–7514 (2019)
51. Yi, L., Kim, V.G., Ceylan, D., Shen, I.C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L.: A scalable active framework for region annotation in 3D shape collections. *ACM Trans. Graph.* **35**(6), 1–12 (2016)
52. Zhang, R., Wang, L., Wang, Y., Gao, P., Li, H., Shi, J.: Parameter is not all you need: starting from non-parametric networks for 3d point cloud analysis. *arXiv preprint arXiv:2303.08134* (2023)
53. 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)
54. Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L.H., Zhou, L., Dai, X., Yuan, L., Li, Y., et al.: Regionclip: region-based language-image pretraining. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16793–16803 (2022)
55. Zhou, J., Wang, J., Ma, B., Liu, Y.S., Huang, T., Wang, X.: Uni3d: exploring unified 3d representation at scale. In: *International Conference on Learning Representations* (2024)
56. Zhu, X., Zhang, R., He, B., Guo, Z., Zeng, Z., Qin, Z., Zhang, S., Gao, P.: Pointclip v2: prompting clip and GPT for powerful 3d open-world learning. *arXiv preprint arXiv:2211.11682* (2022)
57. 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. In: *IEEE International Conference on Robotics and Automation*, pp. 3357–3364 (2017)

Chapter 13

Multimodal Face Generation and Manipulation with Collaborative Diffusion Models



Ziqi Huang, Kelvin C. K. Chan, Yuming Jiang, and Ziwei Liu

Abstract Diffusion models arise as a powerful generative tool recently. Despite the great progress, existing diffusion models mainly focus on unimodal control, i.e., the diffusion process is driven by only one modality of condition. To further unleash the users' creativity, it is desirable for the model to be controllable by multiple modalities simultaneously, e.g., generating and editing faces by describing the age (text driven) while drawing the face shape (mask driven). In this chapter, we discuss Collaborative Diffusion, where pre-trained unimodal diffusion models collaborate to achieve multimodal face generation and editing without retraining. The key insight is that diffusion models driven by different modalities are inherently complementary regarding the latent denoising steps, where bilateral connections can be established upon. Specifically, we discuss dynamic diffuser, a meta-network that adaptively hallucinates multimodal denoising steps by predicting the spatial-temporal influence functions for each pre-trained unimodal model. Collaborative diffusion not only collaborates generation capabilities from unimodal diffusion models but also integrates multiple unimodal manipulations to perform multimodal editing. Extensive qualitative and quantitative experiments demonstrate the superiority of this framework in both image quality and condition consistency.

Keywords Multimodal learning · Multimodal content generation · Diffusion model · Text-driven image generation

13.1 Introduction

Recent years have witnessed substantial progress in image synthesis and editing with the surge of diffusion models [5, 9, 31, 33]. In addition to the remarkable synthesis quality, one appealing property of diffusion models is the flexibility of

Z. Huang · K. C. K. Chan · Y. Jiang · Z. Liu (✉)
S-Lab, Nanyang Technological University, Singapore, Singapore
e-mail: ziqi002@ntu.edu.sg; chan0899@ntu.edu.sg; yuming002@ntu.edu.sg;
ziwei.liu@ntu.edu.sg

conditioning on various modalities, such as texts [1, 7, 16, 17, 22, 26, 29], segmentation masks [26, 36, 37], and sketches [3, 36]. However, existing explorations are largely confined to the use of a single modality at a time. The exploitation of multiple conditions remains under-explored. As a generative tool, its controllability is still limited (Fig. 13.1).

To unleash users' creativity, it is desirable that the model is simultaneously controllable by multiple modalities. While it is trivial to extend the current supervised framework with multiple modalities, training a large-scale model from scratch is computationally expensive, especially when extensive hyperparameter tuning and delicate architecture designs are needed. More importantly, each trained model could only accept a fixed combination of modalities, and hence, retraining is necessary when a subset of modalities are absent or when additional modalities become available. The above demonstrates the necessity of a unified framework that effectively exploits pre-trained models and integrate them for multimodal synthesis and editing.

In this chapter, we discuss *collaborative diffusion*, a framework that synergizes pre-trained unimodal diffusion models for multimodal face generation and editing without the need of retraining. Motivated by the fact that different modalities are complementary to each other (e.g., *text* for age and *mask* for hair shape), we explore the possibility of establishing lateral connections between models driven by different modalities. We discuss *dynamic diffuser* which adaptively predicts the spatial-temporal *influence function* for each pre-trained model. The *dynamic diffuser* dynamically determines spatial-varying and temporal-varying influences of each model, suppressing contributions from irrelevant modalities while enhancing contributions from admissible modalities. In addition to multimodal synthesis, the simplicity and flexibility of our framework enable extension to multimodal face editing with minimal modifications. In particular, the *dynamic diffuser* is first trained for collaborative synthesis. It is then fixed and combined with existing face editing approaches [16, 28] for multimodal editing. It is worth mentioning that users can select the best editing approaches based on their needs without the need of altering the *dynamic diffusers*.

13.2 Methodology

We discuss *collaborative diffusion*, which exploits multiple pre-trained unimodal diffusion models (Sect. 13.2.1) for multimodal generation and editing. The key of our framework is the *dynamic diffuser*, which adaptively predicts the *influence functions* to enhance or suppress the contributions of the pre-trained models based on the spatial-temporal influences of the modalities. Our framework is compatible with most existing approaches for both multimodal guided synthesis (Sect. 13.2.2) and multimodal editing (Sect. 13.2.3) (Fig. 13.2).

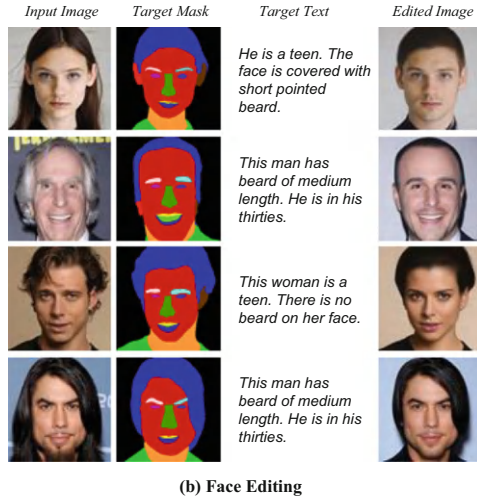
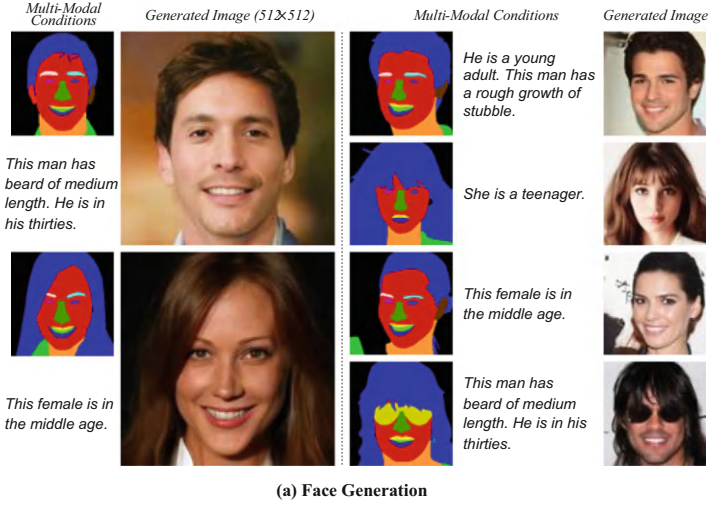


Fig. 13.1 Examples of collaborative diffusion. We discuss *collaborative diffusion*, where users can use multiple modalities to control face generation and editing. **(a) Face Generation.** Given multimodal controls, our framework synthesizes high-quality images consistent with the input conditions. **(b) Face Editing.** Collaborative diffusion also supports multimodal editing of real images with promising identity preservation capability. Figure sourced from collaborative diffusion [11]

13.2.1 Unimodal Conditional Diffusion Models

Diffusion models are a class of generative models that model the data distribution in the form of $p_{\theta}(\mathbf{x}_0) := \int p_{\theta}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$. The *diffusion process* (a.k.a. *forward*

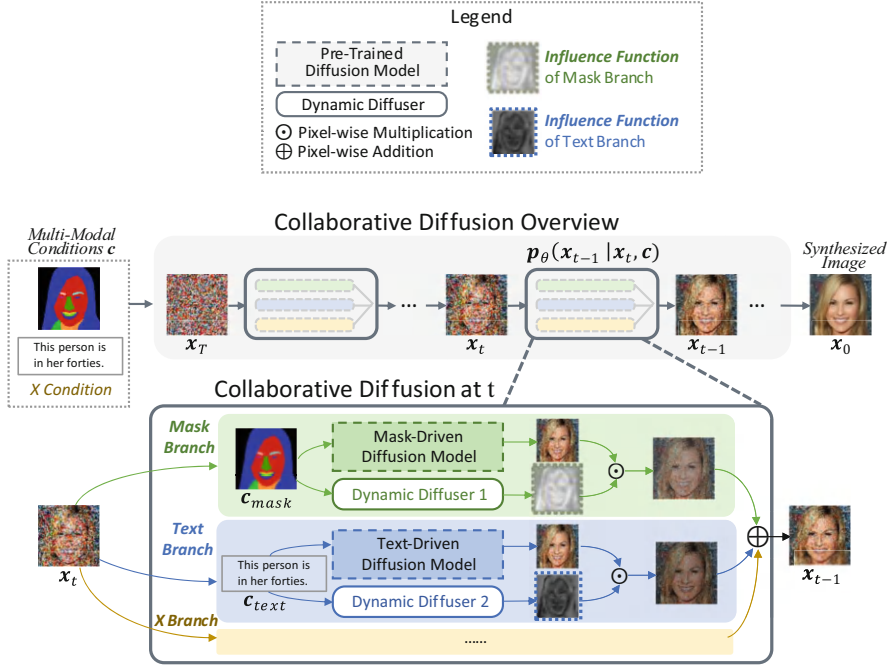


Fig. 13.2 Overview of collaborative diffusion. We use pre-trained unimodal diffusion models to perform multimodal guided face generation and editing. At each step of the reverse process (i.e., from timestep t to $t - 1$), the *dynamic diffuser* predicts the spatial-varying and temporal-varying *influence function* to selectively enhance or suppress the contributions of the given modality. Figure sourced from collaborative diffusion [11]

process) gradually adds Gaussian noise to the data and eventually corrupts the data $\mathbf{x}_0$ into an approximately pure Gaussian noise $\mathbf{x}_T$ using a variance schedule $\beta_1, \dots, \beta_T$:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad (13.1)$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}).$$

Reversing the *forward process* allows sampling new data $\mathbf{x}_0$ by starting from $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$. The *reverse process* is defined as a Markov chain where each step is a learned Gaussian transition $(\mu_\theta, \Sigma_\theta)$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad (13.2)$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)).$$

Training diffusion models relies on minimizing the variational bound on $p(x)$'s negative log-likelihood. The commonly used optimization objective L_{DM} [9] reparameterizes the learnable Gaussian transition as $\epsilon_\theta(\cdot)$ and temporally reweights the variational bound to trade for better sample quality:

$$L_{\text{DM}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2 \right], \quad (13.3)$$

where $\mathbf{x}_t$ can be directly approximated by $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$, with $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$ and $\alpha_t := 1 - \beta_t$.

To sample data $\mathbf{x}_0$ from a trained diffusion model $\epsilon_\theta(\cdot)$, we iteratively denoise $\mathbf{x}_t$ from $t = T$ to $t = 1$ with noise $\mathbf{z}$:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z} \quad (13.4)$$

The unconditional diffusion models can be extended to model conditional distributions $p_\theta(\mathbf{x}_0|c)$, where $\mathbf{x}_0$ is the image corresponding to the condition c such as class labels, segmentation masks, and text descriptions. The conditional diffusion model receives an additional input $\tau(c)$ and is trained by minimizing $\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \tau(c))\|^2$, where $\tau(\cdot)$ is an encoder that projects the condition c to an embedding $\tau(c)$. For brevity, we will use c to represent $\tau(c)$ in our subsequent discussions.

Algorithm 1: Dynamic diffuser training

```

1: repeat
2:    $\mathbf{x}_0, c_1, c_2, \dots, c_M \sim q(\mathbf{x}_0, c_1, c_2, \dots, c_M)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   for  $m = 1, \dots, M$  do
6:      $\epsilon_{\text{pred}, m, t} = \epsilon_{\theta_m}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t, c_m)$ 
7:      $\mathbf{I}_{m, t} = \mathbf{D}_{\phi_m}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t, c_m)$ 
8:   end for
9:    $\hat{\mathbf{I}}_{m, t, p} = \frac{\exp(\mathbf{I}_{m, t, p})}{\sum_{j=1}^M \exp(\mathbf{I}_{j, t, p})}$ , softmax at each pixel  $p$ 
10:   $\epsilon_{\text{pred}, t} = \sum_{m=1}^M \hat{\mathbf{I}}_{m, t} \odot \epsilon_{\text{pred}, m, t}$ 
11:  Take gradient descent step on
     $\nabla_\phi \|\epsilon - \epsilon_{\text{pred}, t}\|^2$  where  $\phi = \{\phi_m | m = 1, \dots, M\}$ 
12: until converged

```

Algorithm 2: Collaborative sampling

```

1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
4:   for  $m = 1, \dots, M$  do
5:      $\boldsymbol{\epsilon}_{pred,m,t} = \boldsymbol{\epsilon}_{\theta_m}(\mathbf{x}_t, t, c_m)$ 
6:      $\mathbf{I}_{m,t} = \mathbf{D}_{\phi_m}(\mathbf{x}_t, t, c_m)$ 
7:   end for
8:    $\hat{\mathbf{I}}_{m,t,p} = \frac{\exp(\mathbf{I}_{m,t,p})}{\sum_{j=1}^M \exp(\mathbf{I}_{j,t,p})}$ , softmax at each pixel  $p$ 
9:    $\boldsymbol{\epsilon}_{pred,t} = \sum_{m=1}^M \hat{\mathbf{I}}_{m,t} \odot \boldsymbol{\epsilon}_{pred,m,t}$ 
10:   $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \boldsymbol{\epsilon}_{pred,t} \right) + \sigma_t \mathbf{z}$ 
11: end for
12: return  $\mathbf{x}_0$ 

```

13.2.2 Multimodal Collaborative Synthesis

In our *collaborative diffusion*, multiple unimodal diffusion models collaborate at each step of the denoising process for multimodal guided synthesis. The core of our framework is the *dynamic diffuser*, which determines the extent of contribution from each collaborator by predicting the spatial-temporal *influence functions*.

Problem Formulation Given M pre-trained unimodal conditional diffusion models $\{\epsilon_{\theta_m}\}$ (each models the distribution $p(\mathbf{x}_0|c_m)$), where the modality index $m = 1, \dots, M$, our objective is to sample from $p(\mathbf{x}_0|\mathbf{c})$, where $\mathbf{c} = \{c_1, c_2, \dots, c_M\}$, without altering pre-trained models.

Dynamic Diffuser In diffusion models, each step of the *reverse process* requires predicting the noise $\boldsymbol{\epsilon}$. With multiple diffusion models collaborating, we need to carefully determine *when*, *where*, and *how much* each diffusion model contributes to the prediction $\boldsymbol{\epsilon}$. At each diffusion timestep $t = T, \dots, 1$, the influence $\mathbf{I}_{m,t}$ from each pre-trained diffusion model ϵ_{θ_m} is adaptively determined by a *dynamic diffuser* $\mathbf{D}_{\phi_m}$:

$$\mathbf{I}_{m,t} = \mathbf{D}_{\phi_m}(\mathbf{x}_t, t, c_m), \quad (13.5)$$

where $m = 1, \dots, M$ is the index of the modalities, $\mathbf{I}_{m,t} \in \mathbb{R}^{h \times w}$, $\mathbf{x}_t$ is the noisy image at time t , c_m is the condition of the m th modality, and $\mathbf{D}_{\phi_m}$ is the *dynamic diffuser* implemented by a UNet [27]. To regularize the overall influence strength,

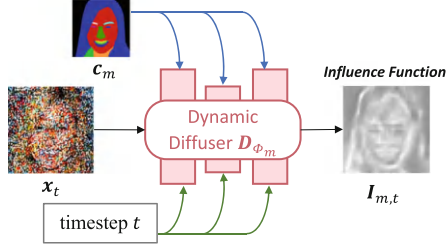


Fig. 13.3 Dynamic diffuser. It predicts the *influence function* that represents the desired level of contribution from each pre-trained diffusion model. The *influence function* varies at different timestep t and different spatial locations and is dependent on the state of diffusion $\mathbf{x}_t$ and the condition input c_m . Figure sourced from collaborative diffusion [11]

we perform softmax across all modalities' $\mathbf{I}_{m,t}$ at each pixel p to obtain the final *influence function* $\hat{\mathbf{I}}_{m,t}$ (Fig. 13.3):

$$\hat{\mathbf{I}}_{m,t,p} = \frac{\exp(\mathbf{I}_{m,t,p})}{\sum_{j=1}^M \exp(\mathbf{I}_{j,t,p})}. \quad (13.6)$$

Multimodal Collaboration We use the M learned *influence functions* $\hat{\mathbf{I}}_{m,t}$ to control the contribution from each pre-trained diffusion model at each denoising step:

$$\epsilon_{pred,t} = \sum_{m=1}^M \hat{\mathbf{I}}_{m,t} \odot \epsilon_{\theta_m}(x_t, t, c_m) \quad (13.7)$$

where ϵ_{θ_m} is the m th collaborator and $\odot$ denotes pixel-wise multiplication. The complete training procedure of *dynamic diffusers* and the image sampling strategy in our *collaborative diffusion* framework are detailed in Algorithm 1 and Algorithm 2, respectively.

Relation with Composable Diffusion Composable diffusion [20] combines two instances of the same text-to-image diffusion model to achieve compositional visual generation via intermediate results addition. Our framework is related to composable diffusion in terms of composing diffusion models for image synthesis but is substantially different in terms of the task nature and methodology. Composable diffusion aims to decompose the text condition into elementary segments to factorize the conditional synthesis problem, while we aim to integrate unimodal collaborators to achieve multimodal controls. Furthermore, different from composable diffusion, which can only compose instances of the same text-based synthesis model (i.e., the weights of the diffusion models being composed are the same), our framework possesses the flexibility in integrating models with different weights, architectures, and modalities through the learned *dynamic diffusers*.

13.2.3 Multimodal Collaborative Editing

In addition to face synthesis, our framework is also capable of combining multiple facial manipulations, each of which is guided by a different modality, into a collaborative edit. Our collaborative framework in theory can integrate any existing unimodal diffusion-based editing approach for collaborative editing.

In this chapter, we demonstrate such possibility by extending Imagic [16] to the multimodal paradigm. We first follow Imagic to fine-tune the embeddings and models to better capture the input face identity during editing. The trained *dynamic diffusers* discussed in Sect. 13.2.2 are then used to combine the fine-tuned models. Algorithm 3 displays the complete procedure of collaborative editing. Note that fine-tuning the pre-trained models is for identity preservation proposed in Imagic, which is independent to our framework. The extension of our framework to editing requires no further training of *dynamic diffusers* (Figs. 13.4 and 13.5).

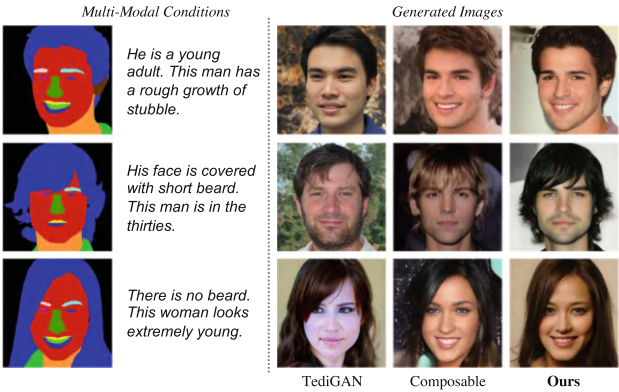


Fig. 13.4 Qualitative comparison of face generation. In the second example, TediGAN and Composable fails to follow mask, while ours generates results highly consistent to both conditions. Figure sourced from collaborative diffusion [11]

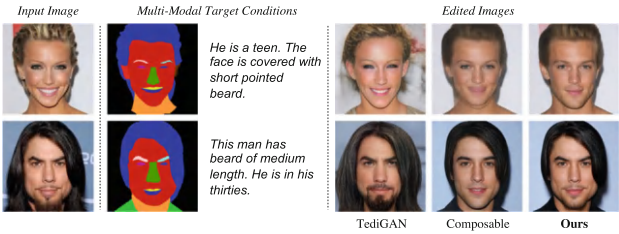


Fig. 13.5 Qualitative comparison of face editing. While TediGAN’s hair shape is inconsistent with mask, and Composable fails to generate beard according to text, our framework produces results highly consistent with both conditions while maintaining identity. Figure sourced from collaborative diffusion [11]

Algorithm 3: Collaborative editing**Require:**

input image $\mathbf{x}_{input}$, target conditions $c_{m,target}$,
 diffusion models ϵ_{θ_m} , dynamic diffusers $\mathbf{D}_{\phi_m}$, ($m = 1, \dots, M$),
 interpolation scale α

```

1: for  $m = 1, \dots, M$  do                                     ▷ Uni-Modal Editing
2:    $c_m = c_{m,target}$ 
3:    $\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_{input} + \sqrt{1 - \alpha_t} \epsilon$ 
4:    $c_{m,opt} = \operatorname{argmin}_{c_m} \mathbb{E}_{\epsilon, t} \|\epsilon - \epsilon_{\theta_m}(\mathbf{x}_t, t, c_m)\|^2$ 
5:    $\theta_{m,opt} = \operatorname{argmin}_{\theta_m} \mathbb{E}_{\epsilon, t} \|\epsilon - \epsilon_{\theta_m}(\mathbf{x}_t, t, c_{m,opt})\|^2$ 
6:    $c_{m,int} = \alpha \cdot c_{m,target} + (1 - \alpha) \cdot c_{m,opt}$ 
7: end for
8:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$                                        ▷ Collaborate the Uni-Modal Edits
9: for  $t = T, \dots, 1$  do
10:   $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
11:  for  $m = 0, \dots, M$  do
12:     $\epsilon_{pred, m, t} = \epsilon_{\theta_{m,opt}}(\mathbf{x}_t, t, c_{m,int})$ 
13:     $\mathbf{I}_{m, t} = \mathbf{D}_{\phi_m}(\mathbf{x}_t, t, c_{m,int})$ 
14:  end for
15:   $\hat{\mathbf{I}}_{m, t, p} = \frac{\exp(\mathbf{I}_{m, t, p})}{\sum_{j=1}^M \exp(\mathbf{I}_{j, t, p})}$ , softmax at each pixel  $p$ 
16:   $\epsilon_{pred, t} = \sum_{m=1}^M \hat{\mathbf{I}}_{m, t} \odot \epsilon_{pred, m, t}$ 
17:   $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon_{pred, t} \right) + \sigma_t \mathbf{z}$ 
18: end for
19: return  $\mathbf{x}_0$ 

```

13.2.4 Implementation Details

In this section, we describe the implementation details of our *collaborative diffusion* framework. We adopt LDM [26] for our unimodal diffusion models since it achieves a good balance between quality and speed. Our *dynamic diffuser* is a time-conditional UNet [27], with conditional embeddings injected via cross-attention [35]. The *dynamic diffuser* is $\times 30$ smaller than the pre-trained diffusion model.

13.2.4.1 Multimodal Collaborative Synthesis Implementation

We adopt LDM [26] as our unimodal diffusion models for its good balance between quality and speed. LDM [26] applies diffusion models in the latent space of autoencoders to reduce the computation overhead of training and sampling. Our framework supports both 256×256 and 512×512 resolution, and we will use the 256×256 version in subsequent discussions for simplicity.

We train a variational autoencoder (VAE) [18], where the encoder compresses $256 \times 256 \times 3$ resolution images into the $64 \times 64 \times 3$ latent space, and the decoder reconstructs the $256 \times 256 \times 3$ images from the $64 \times 64 \times 3$ latent codes. The VAE is trained on the CelebA-HQ [14] Dataset by minimizing the following objective:

$$L_{VAE} = 1.0 \cdot L_{rec} + 1.0 \cdot L_{vgg} + 10^{-6} \cdot L_{kl}, \quad (13.8)$$

where L_{rec} is the L_1 distance between the reconstructed image and the input image, L_{vgg} is the perceptual loss [13] using VGG-16 [30], and L_{kl} is the Kullback-Leibler divergence term which regularizes the VAE latent space toward the Gaussian distribution. The KL term is largely scaled down by a factor of 10^{-6} for two reasons: (1) KL regularization was required in the original VAE for directly sampling latent codes from the Gaussian prior. In this chapter, we simply use VAE as an image compression tool instead of a generative model, so that we do not need strong regularization of VAE latent space. Diffusion models will take care of sampling meaningful latent codes from the weakly regularized VAE latent space. (2) Weaker KL regularization allows relatively stronger focus on image reconstruction and thus potentially less distortion during VAE’s compression-reconstruction process. All our *dynamic diffusers* and unimodal diffusion models are applied in the $64 \times 64 \times 3$ latent space of the pre-trained VAE. The *reverse process* of diffusion models gradually denoises the Gaussian $\mathbf{x}_T \in \mathbb{R}^{64 \times 64 \times 3}$ to $\mathbf{x}_0 \in \mathbb{R}^{64 \times 64 \times 3}$ which will then be decoded to a synthesized image of size $256 \times 256 \times 3$ using VAE’s decoder. In subsequent discussions, we will term $\mathbf{x}_T$ as the “latent code” and $\mathbf{x}_0$ as the “image” to avoid confusion between diffusion models’ latent space and VAE’s latent space.

The text conditions are converted to a sequence of 77 tokens using BERT-tokenizer [4] and are then embedded using 32 transformer encoder layers to obtain the 77×640 text condition embedding. The segmentation masks are downsampled to 32×32 resolution, and each pixel is expanded to a 1×19 one-hot vector to encode the 19 classes of facial components. The unimodal diffusion models are trained with learning rate of 2×10^{-6} and batch size of 32 on CelebA-HQ [14]’s 256×256 images and corresponding condition annotations.

The *dynamic diffuser* $\mathbf{D}_{\theta_m}$ takes the noisy image $\mathbf{x}_t$, timestep t , and the condition c_m as input and predicts the *influence function* $\mathbf{I}_{m,t}$. Since the input noisy image $\mathbf{x}_t \in \mathbb{R}^{64 \times 64 \times 3}$ and the output *influence function* $\mathbf{I}_{m,t} \in \mathbb{R}^{64 \times 64 \times 1}$ has the same spatial resolution, we implement *dynamic diffuser* as a UNet [27].

The timestep t is injected to the *dynamic diffuser* using adaptive layer normalization (AdaLN) [2]:

$$h_{out} = (1 + s(t))\text{LayerNorm}(h_{in}) + b(t), \quad (13.9)$$

where $s(\cdot)$ and $b(\cdot)$ are linear layers that project the timestep t to the scale and bias, respectively, and h_{in} and h_{out} are the intermediate activations before and after timestep injection.

Table 13.1 Hyperparameters of dynamic diffusers

	Dynamic diffuser for text branch	Dynamic diffuser for mask branch
VAE latent space shape	$64 \times 64 \times 3$	$64 \times 64 \times 3$
Number of parameters	13.1M	13.1M
Diffusion steps	1000	1000
Channels	32	32
Attention resolutions	8,4,2	8,4,2
Batch size	8 samples $\times$ 4 GPUs	8 samples $\times$ 4 GPUs
Number of iterations	187k	187k
Learning rate	2×10^{-6}	2×10^{-6}

Table 13.2 Comparison of model size. A *dynamic diffuser* is much smaller than a unimodal conditional diffusion model

Model name	Number of parameters
Mask-driven pre-trained diffusion model	403.6M
Text-driven pre-trained diffusion model	403.6M
Dynamic diffuser for mask branch	13.1M
Dynamic diffuser for text branch	13.1M

The condition c_m is fed into the *dynamic diffuser* via cross-attention [35] with the intermediate activations h :

$$h_{out} = \text{CrossAttention}(h_{in}, c_m) = \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \right) \cdot V, \quad (13.10)$$

$$Q = W_Q \cdot h_{in}, \quad K = W_K \cdot c_m, \quad V = W_V \cdot c_m, \quad (13.11)$$

where W_Q , W_K , and W_V are learnable projection matrices. We provide an overview of the hyperparameters of *dynamic diffusers* in Table 13.1.

Our *dynamic diffuser* has a much smaller model size than the conditional diffusion model, as shown in Table 13.2.

13.2.4.2 Collaborative Editing Implementation

In this chapter, all face editing results, including user study and the qualitative results, are applied on *real images* in the validation split of the CelebA-HQ Dataset.

We use Imagic [16] to demonstrate that our *collaborative diffusion* framework can be extended from synthesis to editing. Imagic is a text-based image editing method using diffusion models and involves three steps to complete an edit. Given the input image $\mathbf{x}_{input}$ and target text $c_{text,target}$, Imagic first optimizes the text

condition so that the diffusion model $\epsilon_{\theta_{text}}$ can reconstruct the input image:

$$c_{text,opt} = \underset{c_{text}}{\operatorname{argmin}} \mathbb{E}_{\epsilon,t} \left\| \epsilon - \epsilon_{\theta_{text}}(\mathbf{x}_t, t, c_{text}) \right\|^2, \quad (13.12)$$

where c_{text} is initialized as $c_{text,target}$ before optimization and $\mathbf{x}_t$ is constructed using the *diffusion process* via $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_{input} + \sqrt{1 - \bar{\alpha}_t} \epsilon$. To further improve the fidelity of the input image, the diffusion model $\epsilon_{\theta_{text}}$ is then fine-tuned with the optimized condition $c_{text,opt}$ being fixed:

$$\theta_{text,opt} = \underset{\theta_{text}}{\operatorname{argmin}} \mathbb{E}_{\epsilon,t} \left\| \epsilon - \epsilon_{\theta_{text}}(\mathbf{x}_t, t, c_{text,opt}) \right\|^2. \quad (13.13)$$

Finally, Imagic interpolates between $c_{text,target}$ and $c_{text,opt}$ to obtain the interpolated condition $c_{text,int}$:

$$c_{text,int} = \alpha \cdot c_{text,target} + (1 - \alpha) \cdot c_{text,opt}. \quad (13.14)$$

The edited image is synthesized using the interpolated text condition $c_{text,int}$ and the fine-tuned diffusion model $\epsilon_{\theta_{text}}$.

We generalize Imagic to achieve mask-driven editing by optimizing the mask condition embedding $c_{mask,target}$ and fine-tuning the pre-trained mask-driven model $\epsilon_{\theta_{mask}}$. We then use *collaborative diffusion* to integrate any text-driven edit and mask-driven edit on the same input image into a collaborative edit.

13.3 Experiments

13.3.1 Dataset in Multimodal-Driven Face Generation and Editing

The CelebA-HQ Dataset [14] consists of 30,000 high-resolution images. We use the multimodal annotations for these images in the CelebAMask-HQ [19] Dataset and the CelebA-Dialog Dataset [12]. The 30,000 images are split into the training set (27,000 images) and validation set (3,000 images). The training of unimodal diffusion models and the *dynamic diffusers* are conducted on the training set, and all the results reported and shown in this chapter are using multimodal conditions from the validation set. The segmentation masks in the CelebAMask-HQ Dataset have 19 classes including facial components and accessories: “background,” “skin,” “nose,” “left eye,” “right eye,” “left eyebrow,” “right eyebrow,” “left ear,” “right ear,” “mouth,” “upper lip,” “lower lip,” “hair,” “hat,” “eyeglass,” “earring,” “necklace,” “neck,” and “cloth.” The texts in the CelebA-Dialog Dataset provide fine-grained natural language descriptions of the five attributes: “bangs,” “eyeglasses,” “beard,” “smiling,” and “age.” To avoid conflict between segmentation masks and texts, we

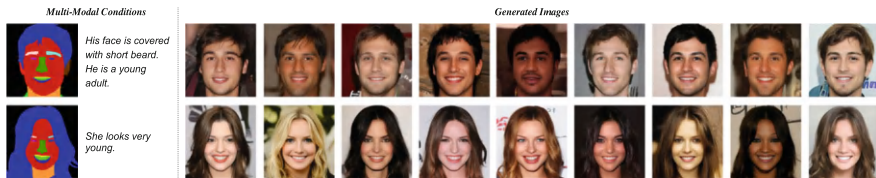


Fig. 13.6 Diversity of face generation. Given input conditions, our method demonstrates promising synthesis diversity in facial attributes that are not constrained by a specific combination of multimodal conditions, such as hair color and skin tone. Figure sourced from collaborative diffusion [11]

trimmed the descriptions regarding “bangs,” “eyeglasses,” and “smiling” from the natural language descriptions as they are described by segmentation masks as well. In this chapter, we train the mask-driven diffusion models on CelebAMask-HQ and the text-driven models on CelebA-Dialog. For *dynamic diffuser*, we simply combine the mask and text for each image without further processing (Fig. 13.6).

13.3.2 Existing Methods in Multi-Model-Driven Image Generation

TediGAN [38, 39] *TediGAN* is a StyleGAN-based method for text-driven face generation and manipulation. It can be extended to support other modality’s guidance by projecting the conditions into StyleGAN’s $\mathcal{W}+$ latent space and performing style mixing to achieve multimodal control. We use TediGAN [38, 39]’s official implementation for text-driven and mask-driven generation and editing. For multimodal-driven generation, we mix the style codes of text and mask using TediGAN’s style mixing control mechanism. For editing, the style codes are initialized using the inverted $\mathcal{W}+$ codes of the input image, and the remaining steps are the same as generation.

Composable diffusion [20] parallelly implements two instances of the same text-to-image diffusion model to perform compositional scene generation. We generalize it by replacing the two text-to-image model instances by the text-driven model and the mask-driven model trained using LDM [26]. The two models are composed by averaging the intermediate diffusion results at each step of the *reverse process*. Both *Composable* and *Ours* use the same set of pre-trained unimodal conditional diffusion models described in Sect. 13.2.4. To accelerate the time-consuming sampling process of diffusion models while maintaining fair comparisons, we use DDIM [32] with 50 steps in all experiments (i.e., quantitative, qualitative, and user study) involving *Composable* or *Ours*.

13.3.3 *Evaluation Metrics in Multimodal-Driven Image Generation and Editing*

FID We use Frechet Inception Distance (FID) [8] to measure the quality of images synthesized by different methods. FID computes the feature representation’s distance between generated images and real images. A lower FID implies better sample quality.

CLIP Score CLIP [25] is a vision-language model trained on large-scale datasets. It uses an image encoder and a text encoder to project images and texts to a common feature space, respectively. The CLIP score is computed as the cosine similarity between the normalized image and text embeddings. A higher score usually indicates higher consistency between the output image and the text caption.

Mask Accuracy For each output image, we predict the segmentation mask using the face parsing network provided by CelebAMask-HQ [19]. Mask accuracy is the pixel-wise accuracy against the ground-truth segmentation. A higher average accuracy indicates better consistency between the output image and the segmentation mask.

User Study We conduct user study with 25 human evaluators to measure the effectiveness of the methods perceptually. For *face generation*, we randomly sample multimodal conditions in the validation split of CelebA-HQ Dataset and then synthesize output images given the conditions. Evaluators are provided with the input conditions and the output images, and they are asked to choose the best image based on (1) image photo-realism, (2) image-text consistency, (3) image-mask consistency. User study is conducted similarly for *face editing*, except that (1) evaluators are also provided with the input image for editing, and (2) they are also asked to assess the level of identity preservation between the edited image and input image (Fig. 13.7).

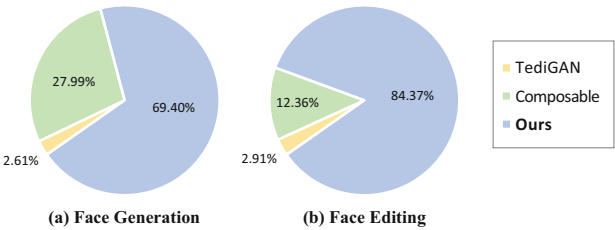


Fig. 13.7 User study. Among the three methods, the majority of users vote for our results as the best for both generation (69.40%) and editing (84.37%), in terms of image quality, condition consistency, and identity preservation. Figure sourced from collaborative diffusion [11]



Fig. 13.8 Face generation results. Our method generates realistic images under different combinations of multimodal conditions, even for relatively rare combinations in the training distribution, such as a man with long hair. Figure sourced from collaborative diffusion [11]

13.3.4 Face Generation and Face Editing Results

Qualitative Comparison In Fig. 13.8, we provide the generation results of our *collaborative diffusion* under different combinations of texts and masks. It is observed that our framework is capable of synthesizing realistic outputs consistent with the multimodal inputs, even for relatively rare conditions, such as a man with long hair. We then compare our method with TediGAN [38, 39] and composable diffusion [20] in Fig. 13.4. For example, in the second example, TediGAN and composable diffusion’s results are inconsistent with the mask, while ours is able to maintain both mask and text consistency. We also show in Fig. 13.6 that our method synthesizes images with high diversity without losing condition consistency.

In addition to image generation, it is also observed in Fig. 13.9 that our framework is able to edit images based on multimodal conditions while preserving identity. The comparison to existing works in Fig. 13.5 also verifies our effectiveness. For example, in the second example, while TediGAN is unable to synthesize hair consistent to the mask and composable diffusion fails to generate beard according to the text, our framework is able to generate results highly consistent to both conditions while maintaining identity.

Quantitative Comparison As shown in Table 13.3, our *collaborative diffusion* outperforms TediGAN and Composable Diffusion in all three objective metrics. In addition, as depicted in Fig. 13.7, our framework achieves the best result in 69.40%

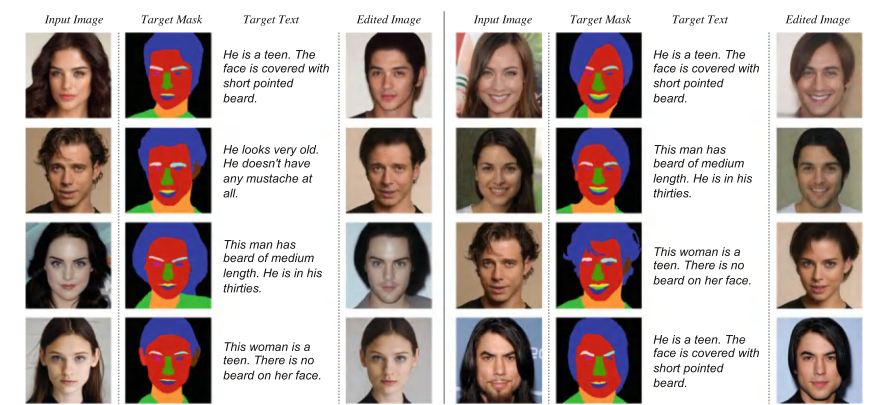


Fig. 13.9 Face editing results. Given the input real image and target conditions, we display the edited image using our method. Figure sourced from collaborative diffusion [11]

Table 13.3 Quantitative results of face generation. Compared with TediGAN and composable diffusion, our method synthesizes images with better quality (lower FID) and higher consistency with the text and mask conditions

Method	FID ↓	Text (%) ↑	Mask (%) ↑
TediGAN [38, 39]	157.81	24.27	72.19
Composable [20]	124.62	23.94	76.11
Ours	111.36	24.51	80.25

and 84.37% of the time for face generation and editing, respectively. These results verify our effectiveness in both image quality and image-condition consistency.

13.3.5 Analysis of Influence Functions

In this section, we visualize the *influence functions* and show that it is necessary for the *influence function* to be both spatially and temporally varying to facilitate effective collaboration.

13.3.5.1 Effects of Spatial Variations in Influence Functions

As shown in Fig. 13.10, the *influence functions* behave differently at different spatial regions. For instance, the influence for the mask-driven model mainly lies on the contours of facial regions, such as the outline of hair, face, and eyes, as these regions are crucial in defining facial layout. In contrast, the influence for the text-driven model is stronger at skin regions including cheeks and chin. This is because the

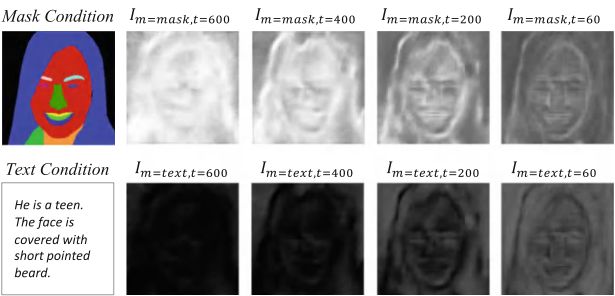


Fig. 13.10 Visualization of influence functions. The *influence functions* vary spatially at different face regions and temporally at different diffusion timesteps. Figure sourced from collaborative diffusion [11]

Table 13.4 Analysis of influence functions. Temporal or spatial suppression in influence variation introduces performance drops, which shows the *necessity of influence functions’ spatial-temporal adaptivity*

Method	FID ↓	Text (%) ↑	Mask (%) ↑
Ours w/o spatial	117.81	24.36	80.08
Ours w/o temporal	117.34	24.48	77.07
Ours	111.36	24.51	80.25

attributes related skin texture, such as age and beard length, are better described by text.

To verify its necessity, we remove the spatial variation of *influence functions*. From Table 13.4, we observe that removing the spatial variance results in deterioration in both output quality and condition consistency. This corroborates our hypothesis that it is important to assign different weights to different modalities at different spatial regions.

13.3.5.2 Effects of Temporal Variations in Influence Functions

In addition to the spatial variation, it is also observed in Fig. 13.10 that the influence from the mask-driven model is stronger at earlier diffusion stages (i.e., larger t), since early stages focus on initializing the facial layout using the mask-driven model’s predictions. At later stages, the influence from the text-driven model increases as the textural details (e.g., skin wrinkles and beard length) are instantiated using information from the text.

We remove the temporal variation in the *influence functions* to demonstrate its importance. It is shown in Table 13.4 that both the quality and condition consistency drop without the temporal variation.

13.3.6 Additional Qualitative Results

We show various qualitative results in Figs. 13.11, 13.12, 13.13, 13.14 and 13.15, which are located at the end of this appendix chapter.

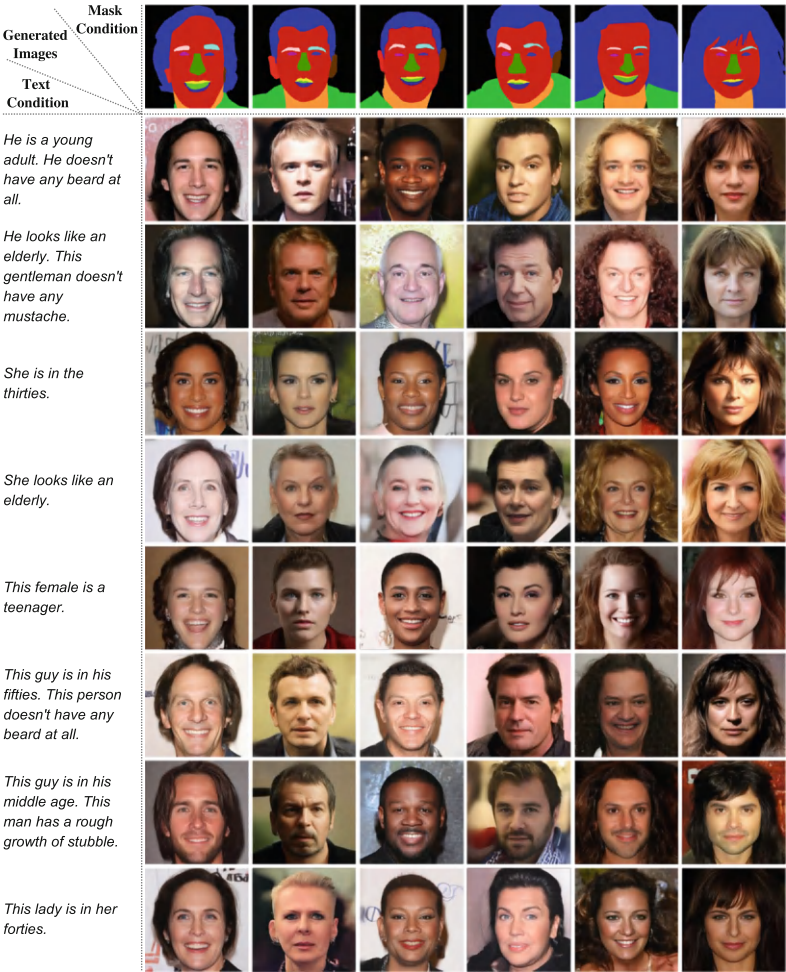


Fig. 13.11 More face generation results (A). Our method generates realistic images under different combinations of multimodal conditions, even for relatively rare combinations in the training distribution, such as a man with long hair. Figure sourced from collaborative diffusion [11]

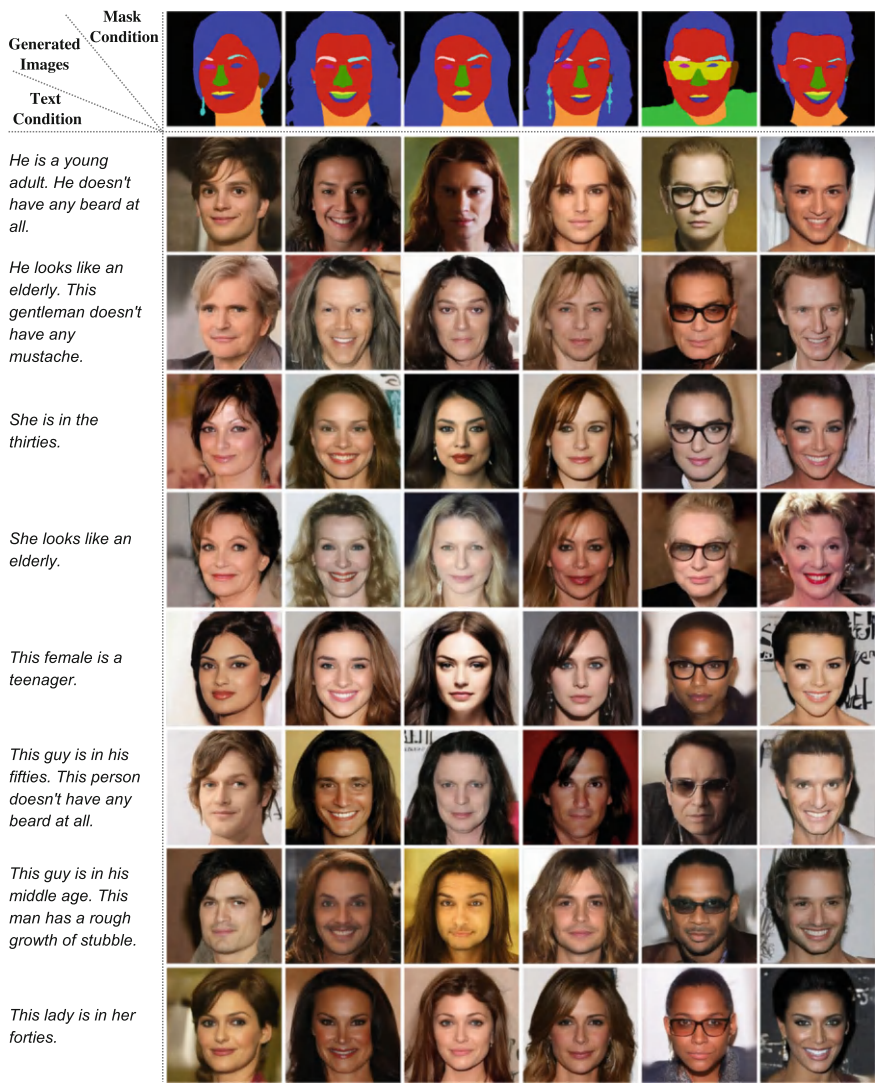


Fig. 13.12 More face generation results (B). Our method generates realistic images under different combinations of multimodal conditions, even for relatively rare combinations in the training distribution, such as a man with long hair. Figure sourced from collaborative diffusion [11]

13.3.6.1 Generation and Editing

We provide more face generation results in Figs. 13.11 and 13.12, and face editing results in Fig. 13.13.

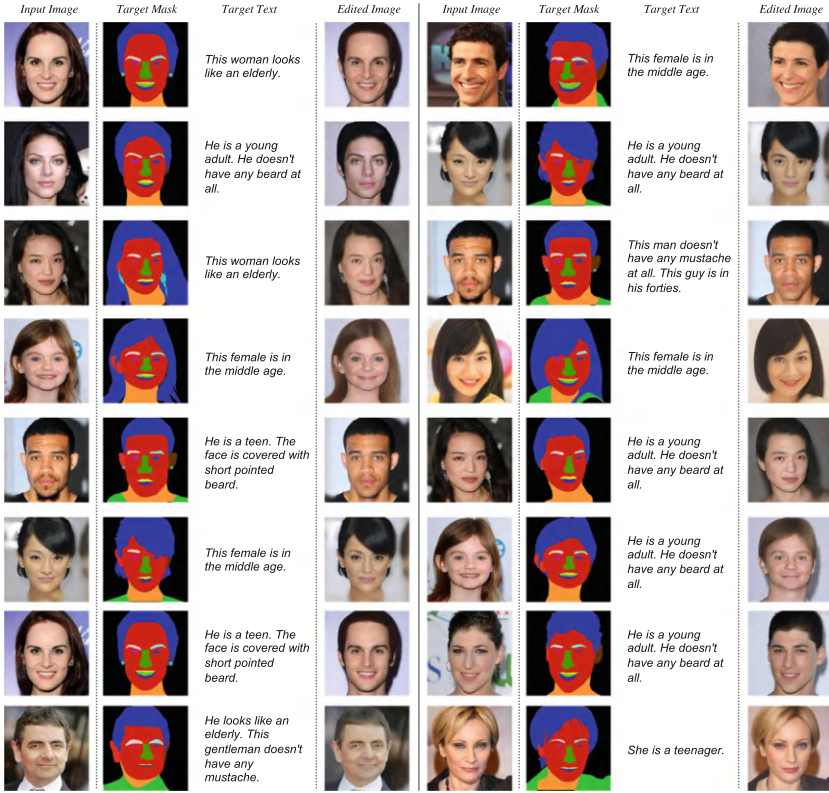


Fig. 13.13 Face editing results. Given the input real image and target conditions, we display the edited image using our method. Figure sourced from collaborative diffusion [11]

13.3.6.2 Visualization of Influence Functions

In Figs. 13.14 and 13.15, we visualize the *influence functions* to show their spatial-temporal variation. Given the mask condition in Fig. 13.14a, b displays the *influence functions* of the mask-driven collaborator at each DDIM sampling step $t = 980, 960, \dots, 20, 0$, from the left to right, top to down. The text branches' *influence functions* are displayed similarly. In Fig. 13.14f, we show the intermediate diffusion results $\mathbf{x}_t$ for $t = 980, 960, \dots, 20, 0$ by decoding them to the image space using the VAE decoder. The final synthesized image is displayed in Fig. 13.14e. Figure 13.15 displays the intermediate results using a different set of multimodal conditions and is arranged in the same way as Fig. 13.14.

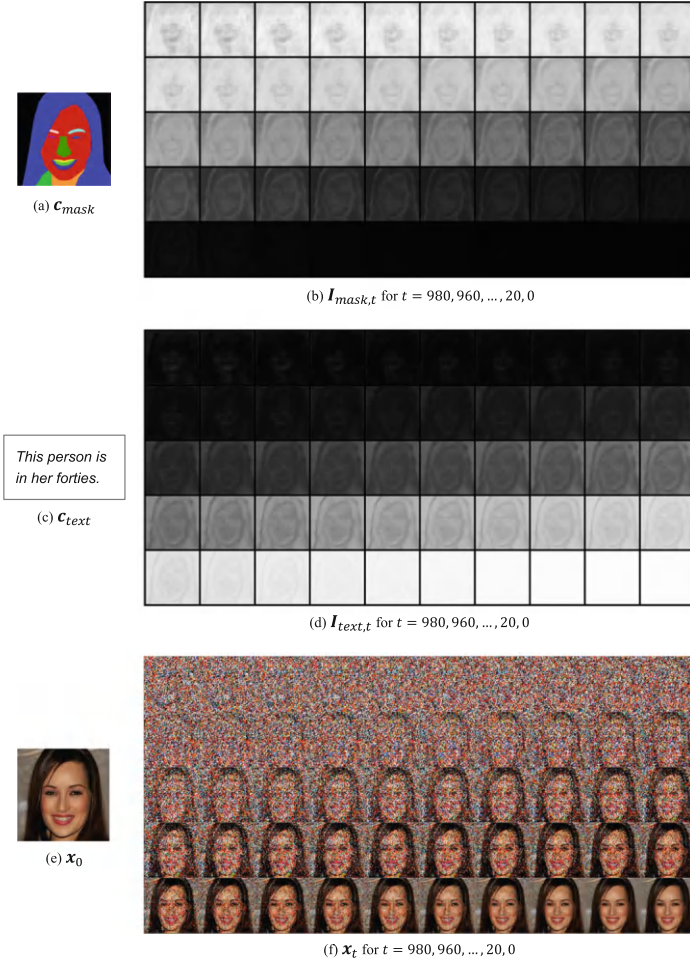


Fig. 13.14 Visualization of influence functions (A). The *influence function* varies spatially at different face regions and temporally at different diffusion timesteps. The spatial-temporal adaptivity of *influence functions* facilitates effective collaboration. Figure sourced from collaborative diffusion [11]

13.3.7 Potential Societal Impacts

Collaborative diffusion can achieve high-quality real image editing driven by different modalities. However, such capabilities could be applied to maliciously manipulate real human faces. Therefore, we advise users to use *collaborative diffusion* only for proper recreational purposes.

The rapid progress in generative models unleashes creativity but inevitably introduces various societal concerns. First, it becomes easier to create false imagery or

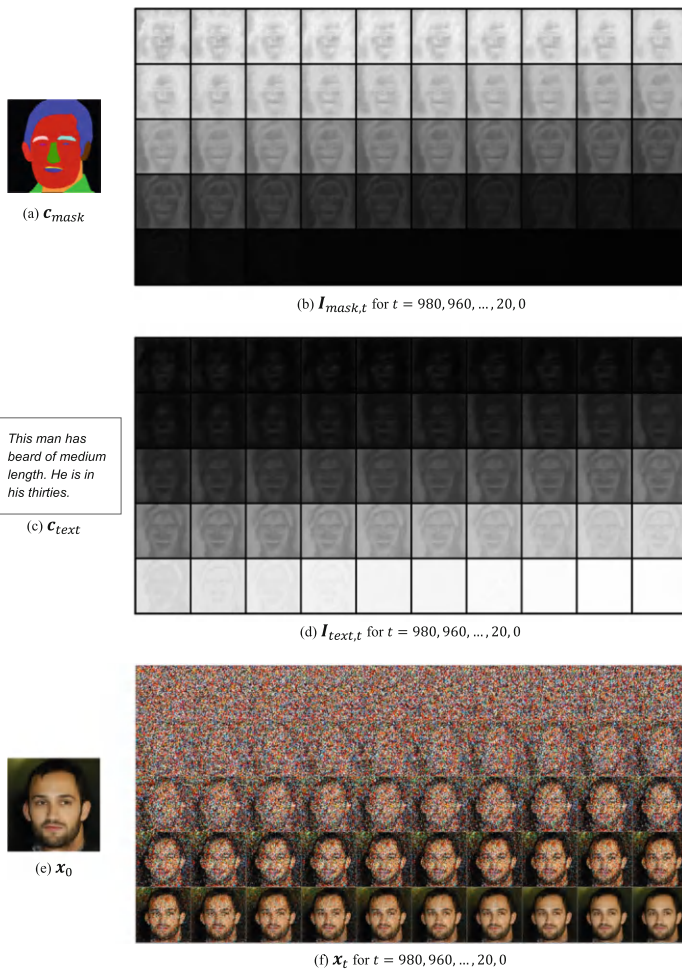


Fig. 13.15 Visualization of influence functions (B). The *influence function* varies spatially at different face regions and temporally at different diffusion timesteps. The spatial-temporal adaptivity of *influence functions* facilitates effective collaboration. Figure sourced from collaborative diffusion [11]

maliciously manipulate the data, which could lead to the spread of misinformation. Second, training data might be revealed during the sampling process without explicit consent from data owner [34]. Third, generative models potentially suffer from the biases present in the training data [6]. For *collaborative diffusion*, we conduct training on CelebA-HQ [14]’s faces of various celebrities, which could potentially deviate from the looks of the general population. We hope to see more research to alleviate the risks and biases of generative models, and we advise all to apply generative models with discretion.

13.4 Conclusion

With generative AI gaining increasing attention, multimodal conditioning becomes an indispensable direction to unleash creativity and enable comprehensive controls from human creators. In this chapter, we discuss *collaborative diffusion*, where pre-trained unimodal diffusion models collaboratively achieve multimodal face generation and editing without being retrained. With our *dynamic diffuser*, this framework could be used to extend arbitrary unimodal approach to the multimodal paradigm through predicting the relative influence of different modalities. We believe our idea of synergizing unimodal models for multimodal tasks would be a good inspiration for future works in various domains such as motion and 3D generation.

Further Reading

Interested readers may refer to the following material:

- [12, 15, 23, 38] for earlier GAN-based face generation and editing methods
- [5, 9, 24, 26] for an in-depth understanding of diffusion models and diffusion-based approaches for image generation
- [10, 21, 40] for more control techniques in diffusion-based image generation

References

1. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: CVPR (2022)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Cheng, S.I., Chen, Y.J., Chiu, W.C., Tseng, H.Y., Lee, H.Y.: Adaptively-realistic image generation from stroke and sketch with diffusion model. In: WACV (2023)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. In: NeurIPS (2021)
6. Esser, P., Rombach, R., Ommer, B.: A note on data biases in generative models. In: NeurIPS Workshop (2020)
7. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: CVPR (2022)
8. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
10. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
11. Huang, Z., Chan, K.C., Jiang, Y., Liu, Z.: Collaborative diffusion for multi-modal face generation and editing. In: CVPR (2023)

12. Jiang, Y., Huang, Z., Pan, X., Loy, C.C., Liu, Z.: Talk-to-Edit: fine-grained facial editing via dialog. In: ICCV (2021)
13. Johnson, J., Alahi, A., Li, F.F.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
14. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: ICLR (2018)
15. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: CVPR (2020)
16. Kavar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. arXiv preprint arXiv:2210.09276 (2022)
17. Kim, G., Kwon, T., Ye, J.C.: DiffusionCLIP: text-guided diffusion models for robust image manipulation. In: CVPR (2022)
18. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
19. Lee, C.H., Liu, Z., Wu, L., Luo, P.: MaskGAN: towards diverse and interactive facial image manipulation. In: CVPR (2020)
20. Liu, N., Li, S., Du, Y., Torralba, A., Tenenbaum, J.B.: Compositional visual generation with composable diffusion models. arXiv preprint arXiv:2206.01714 (2022)
21. Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)
22. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
23. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: StyleCLIP: text-driven manipulation of stylegan imagery. In: ICCV (2021)
24. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
25. 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)
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
27. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: MICCAI (2015)
28. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: fine tuning text-to-image diffusion models for subject-driven generation. In: CVPR (2023)
29. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., et al.: Photorealistic text-to-image diffusion models with deep language understanding. arXiv preprint arXiv:2205.11487 (2022)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
31. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015)
32. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
33. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021)
34. Tinsley, P., Czajka, A., Flynn, P.: This face does not exist...but it might be yours! identity leakage in generative models. In: WACV (2021)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
36. Wang, T., Zhang, T., Zhang, B., Ouyang, H., Chen, D., Chen, Q., Wen, F.: Pretraining is all you need for image-to-image translation. arXiv preprint arXiv:2205.12952 (2022)

37. Wang, W., Bao, J., Zhou, W., Chen, D., Chen, D., Yuan, L., Li, H.: Semantic image synthesis via diffusion models. arXiv preprint arXiv:2207.00050 (2022)
38. Xia, W., Yang, Y., Xue, J.H., Wu, B.: TediGAN: Text-guided diverse face image generation and manipulation. In: CVPR (2021)
39. Xia, W., Yang, Y., Xue, J.H., Wu, B.: Towards open-world text-guided face image generation and manipulation. arXiv preprint arXiv:2104.08910 (2021)
40. Zhang, L., Agrawala, M.: Adding conditional control to text-to-image diffusion models. arXiv preprint arXiv:2302.05543 (2023)

Chapter 14

Boosting Diffusion U-Net with Free Lunch for Text-to-Image and Text-to-Video Generation



Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu

Abstract Diffusion models have emerged as a powerful tool for generative tasks, showing impressive capabilities in various domains. This chapter explores the potential of the diffusion U-Net architecture, highlighting how it can serve as an effective and straightforward approach to enhancing generation quality in generative models. Through an analysis of the U-Net structure, it becomes clear that the backbone is primarily responsible for denoising, while the skip connections play a crucial role in introducing high-frequency details into the decoder. However, these skip connections can sometimes overshadow the backbone's key functions. By reconsidering the interaction between the skip connections and the backbone, it is possible to significantly improve generation quality without the need for additional training or fine-tuning. This approach, referred to as FreeU, involves re-weighting the contributions of these two components to leverage their strengths. Promising results on tasks such as image and video generation demonstrate how FreeU can be easily integrated into existing diffusion models like Stable Diffusion, DreamBooth, and ControlNet, requiring only a few adjustments during inference. The simplicity of the method—adjusting two scaling factors—makes it a powerful tool for improving generation quality with minimal changes to the underlying model.

Keywords Diffusion model · Text-to-image generation · Text-to-video generation · Multimodal learning · Free lunch

C. Si · Z. Huang · Y. Jiang · Z. Liu (✉)
Nanyang Technological University, Singapore, Singapore
e-mail: chenyang.si@ntu.edu.sg; ziqi002@ntu.edu.sg; yuming002@ntu.edu.sg;
ziwei.liu@ntu.edu.sg

14.1 Introduction

Diffusion probabilistic models, a cutting-edge category of generative models, have garnered significant attention, particularly for tasks related to computer vision [7, 8, 11, 18, 33, 41, 45, 46, 49]. These diffusion models are composed of two key processes: *diffusion process* and the *denoising process*. In the *diffusion process*, Gaussian noise is gradually added to the input data and eventually corrupts it into approximately pure Gaussian noise. During the *denoising process*, the original input data is recovered from its noise state through a learned sequence of inverse diffusion operations. Usually, a U-Net is employed to iteratively predict the noise to be removed at each denoising step. Existing works [3, 47, 59, 67] primarily focus on utilizing pre-trained diffusion U-Nets for downstream applications, while the internal properties of the diffusion U-Net remain largely under-explored.

This chapter delves into the denoising process of the diffusion U-Net. For a comprehensive analysis, our first objective is to explore the mechanics behind ***how images are generated from noise during the denoising process***. To understand what’s going on, we conduct an investigation within the Fourier domain, focusing on the generative evolution during the denoising process. Our meticulous analysis reveals a subtle modulation of the low-frequency components, which demonstrate a gentle rate of change. In contrast, the high-frequency components showcase more pronounced dynamics throughout the denoising process. Fundamentally, low-frequency components bestow upon an image its foundational structure and color attributes. Excessive adjustments during iterative denoising risk undermining the image’s intrinsic semantic integrity. High-frequency components, which represent details like edges and textures, are more affected by noise. Hence, the goal of the denoising process is to reduce this noise while ensuring the preservation of critical details.

Building on this foundational understanding, our analysis scope is expanded to ***how diffusion U-Net implements denoising process***, thereby ascertaining the specific contributions of the U-Net architecture within the diffusion framework. Structurally, the U-Net architecture comprises a primary backbone network, encompassing both an encoder and a decoder, as well as the skip connections that bridge information transfer between the encoder and decoder, as shown in Fig. 14.1. This investigation reveals that the main backbone of the U-Net primarily contributes to denoising. Conversely, the skip connections are observed to introduce high-frequency features into the decoder module. These connections propagate high-frequency information to make U-Net easier to recover the input data during training. Yet, an unintended consequence of this propagation is the potential weakening of the backbone’s inherent denoising capabilities during the inference. This can lead to a reduction in the generation quality, e.g., abnormal image details, as illustrated in Fig. 14.2.

With these insights in mind, a new approach—referred to as “***FreeU***”—emerges as a promising solution for enhancing sample quality without the need for additional training or fine-tuning. The core idea of FreeU is to adjust the contributions from

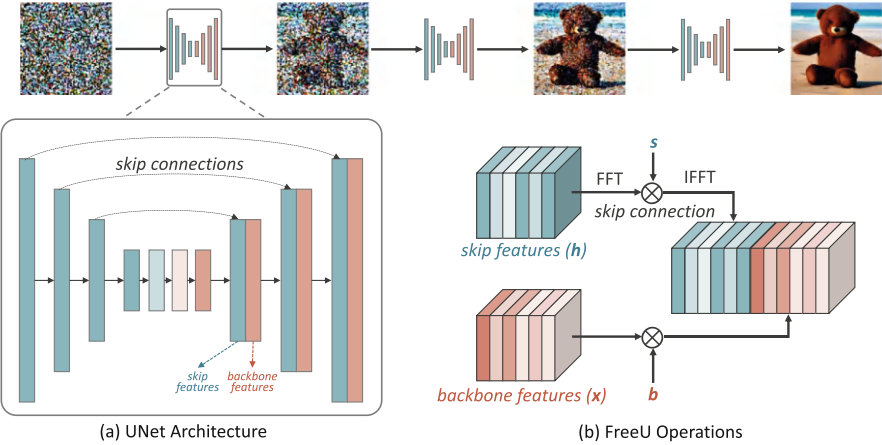


Fig. 14.1 FreeU framework. **(a)** U-Net Skip features and backbone features. In U-Net, the skip features and backbone features are concatenated together at each decoding stage. We apply the FreeU operations during concatenation. **(b)** FreeU operations. Two modulation factors (b and s) are employed to balance the feature contributions from the backbone and skip connections (Image sourced from FreeU [52])

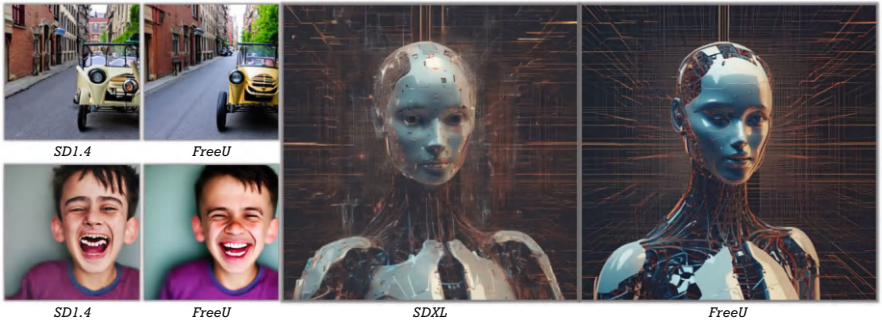


Fig. 14.2 FreeU substantially improves diffusion model sample quality at no costs: no training, no additional learnable parameter introduced, and no increase in memory or sampling time (Image sourced from FreeU [52])

the two main components of the U-Net architecture—the backbone and the skip connections—during inference. This is achieved through the use of two modulation factors that help balance the feature maps coming from each of these components. The first, termed the backbone feature factors, aims to amplify the feature maps of the main backbone, thereby bolstering the denoising process. However, we find that while the inclusion of backbone feature scaling factors yields significant improvements, it can occasionally lead to an undesirable oversmoothing of textures. To mitigate this issue, we introduce the second factor, skip feature scaling factors, aiming to alleviate the problem of texture oversmoothing.

The FreeU method exhibits seamless adaptability when integrated with existing diffusion models. A thorough experimental evaluation of this approach is conducted, employing Stable Diffusion [43, 46], ModelScope [37], Dreambooth [47], ReVersion [23], Rerender [63], ScaleCrafter [16], Animatediff [14], and ControlNet [67] as the foundational models for benchmark comparisons. By employing FreeU during the inference phase, these models indicate a discernible enhancement in the quality of generated samples, as shown in Fig. 14.2.

The rest of this chapter is organized as follows. Section 14.2 gives a brief description of related work. The analysis and proposed approach is described in Sect. 14.3. Section 14.4 shows the experimental results. Conclusions and discussions are given in Sect. 14.5.

14.2 Related Work

14.2.1 Diffusion Probabilistic Models

Diffusion models have achieved great success in generation tasks [7, 8, 11, 13, 18, 24, 29, 33, 41, 45, 46, 49]. Distinct from other classes of generative models [4, 9, 12, 25–28, 32, 39, 54, 56] such as variational autoencoder (VAE) [32], generative adversarial networks (GANs) [4, 12, 25–28, 39], and vector-quantized approaches [9, 54], diffusion models introduce a novel generative paradigm. These models employ a fixed Markov chain to map the latent space, facilitating intricate mappings that capture latent structural complexities within a dataset. Its impressive generative capabilities, ranging from the high level of details to the diversity of the generated examples, have fueled groundbreaking advancements in a variety of computer vision applications such as image synthesis [18, 46, 49], image editing [1, 6, 21, 38], image-to-image translation [6, 48, 57], and text-to-video generation [3, 15, 19, 37, 53, 58, 59, 66]. Though successful, these studies mainly focus on utilizing pre-trained diffusion models for downstream applications, while the internal properties of the diffusion models remain largely under-explored. In this chapter, we conduct a pioneering exploration of the potential of diffusion models.

14.2.2 Frequency Analysis

Frequency analysis is commonly used to understand and enhance the performance of deep neural networks [2, 42, 51, 55, 60, 61, 65]. Some studies such as [5, 10, 31, 50] have closely examined the frequency biases present in GANs models. Furthermore, the frequency characteristics of trained small diffusion models have been studied in [64]. In this chapter, we explore the denoising process in the Fourier domain for diffusion models and pioneer an investigation into the denoising potential of diffusion U-Net.

14.3 Methodology

In this section, we commence our investigation by analyzing the process of image generation from noise. Subsequently, we delve into a detailed analysis of how the U-Net architecture implements the denoising process. Finally, we then discuss FreeU, an approach that enhances the denoising capabilities of the U-Net without the need for additional training or fine-tuning.

14.3.1 Preliminaries

Generating samples from a diffusion model is initiated by sourcing from a Gaussian noise distribution and subsequently following the inverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. This results in a trajectory sequence $\mathbf{x}_T, \mathbf{x}_{T-1}, \dots, \mathbf{x}_0$ ending with the generated sample $\mathbf{x}_0$. Crucially, the sampling process depends on the denoising model ϵ_θ to eliminate noise. The optimization objective of denoising model is as follows:

$$\mathcal{L}_{DM} = \mathbb{E}_{\mathbf{x}, \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2 \right] \quad (14.1)$$

In most implementations, the denoising model is realized using a time-conditional U-Net architecture. Hence, its denoising ability plays a pivotal role in determining the quality of the data generated.

14.3.2 How to Generate Images from Noise During Denoising Process?

To better understand the denoising process, we conduct an investigation within the Fourier domain to perspective the generated process of diffusion models. As illustrated in Fig. 14.3, the uppermost row provides the progressive denoising process, showcasing the generated images across successive iterations. The subsequent two rows exhibit the associated low-frequency and high-frequency spatial domain information after the inverse Fourier transform, aligning with each respective step.

Evident from Fig. 14.3 is the gradual modulation of low-frequency components, showing a soft rate of change, while their high-frequency components show more obvious changes throughout the entire denoising process. These findings are further corroborated in Fig. 14.4. This can be intuitively explained: (1) low-frequency components inherently embody the global structure and characteristics of an image, encompassing global layouts and smooth color. These components encapsulate the foundational global elements that constitute the image's essence and representation. Its rapid alterations are generally unreasonable in denoising processes. Drastic

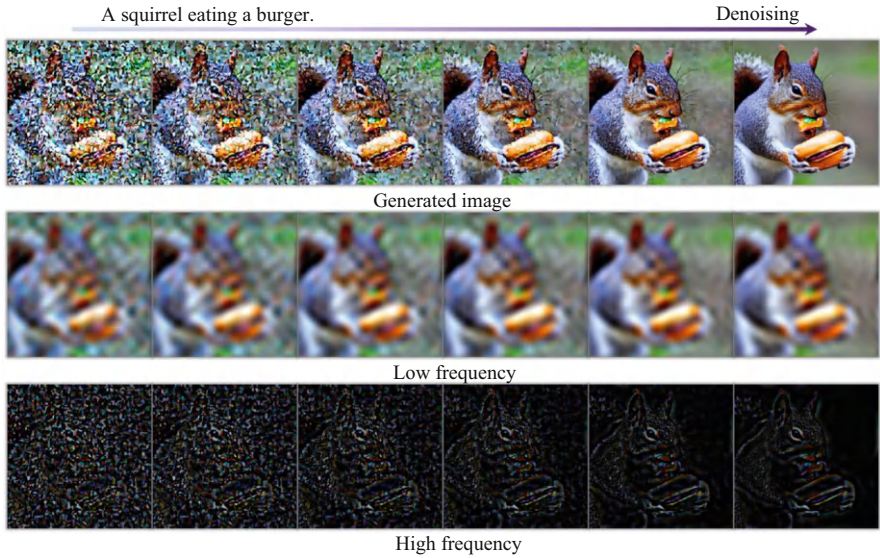
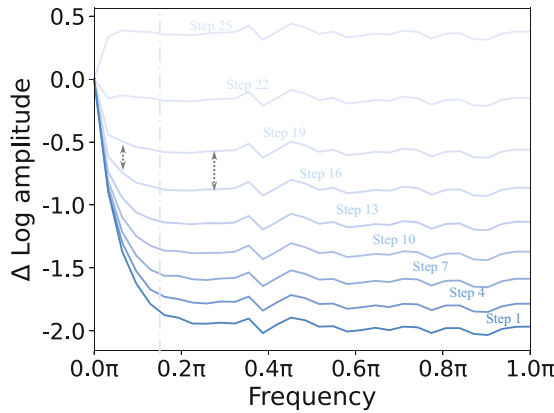


Fig. 14.3 Denoising process visualization: The top row shows the generated images of the denoising process. The next two rows display low-frequency and high-frequency components after the inverse Fourier transform. Low-frequency components change slowly, whereas high-frequency components exhibit more significant variations during the denoising process (Image sourced from FreeU [52])

Fig. 14.4 Relative log amplitudes of Fourier for denoising process. At each denoising step t , we visualize the relative log amplitudes of Fourier of recovered data x_t . We observe that the high-frequency components of x_t drop drastically during the denoising process (Image sourced from FreeU [52])



changes to these components could fundamentally reshape the image’s essence, an outcome typically incompatible with the objectives of denoising processes. (2) Conversely, high-frequency components contain rapid changes in the images, such as edges and textures. These finer details are markedly sensitive to noise, often manifesting as random high-frequency information when noise is introduced to an image. Consequently, denoising processes need to expunge noise while upholding indispensable intricate details.

14.3.3 How Does Diffusion U-Net Perform Denoising?

Building on this foundational understanding throughout the denoising process, we extend the investigation to delineate the specific contributions of the U-Net architecture within the denoising process, to explore the internal properties of the denoising network. As illustrated in Fig. 14.1, the U-Net architecture comprises a primary backbone network, as well as the skip connections that facilitate information transfer between the encoder and decoder.

To evaluate the role of the backbone and lateral skip connections in the denoising process, we conduct a controlled experiment wherein we introduce two multiplicative scaling factors—denoted as b and s —to modulate the feature maps generated by the backbone and skip connections, respectively, prior to their concatenation. As shown in Fig. 14.5, it is evident that elevating the scale factor b of the backbone distinctly enhances the quality of generated images. Conversely, variations in the scaling factor s , which modulates the impact of the lateral skip connections, appear to exert a limited influence on the quality of the generated images.

The backbone of U-Net Building upon these observations, we subsequently probed the underlying mechanisms for the enhancement in image generation quality when the scaling factor b associated with the backbone feature map increases. The analysis reveals that this quality improvement is fundamentally linked to an amplified denoising capability imparted by the U-Net architecture’s backbone. As delineated in Fig. 14.6, a commensurate increase in b correspondingly results in a suppression of high-frequency components in the images generated by the diffusion model. Therefore, in Fig. 14.5, when $b = 0.6$, the generated images exhibit a significant amount of noise that adversely affects image quality. In contrast, when $b = 1.4$, highly clear images are generated. This indicates that the primary role

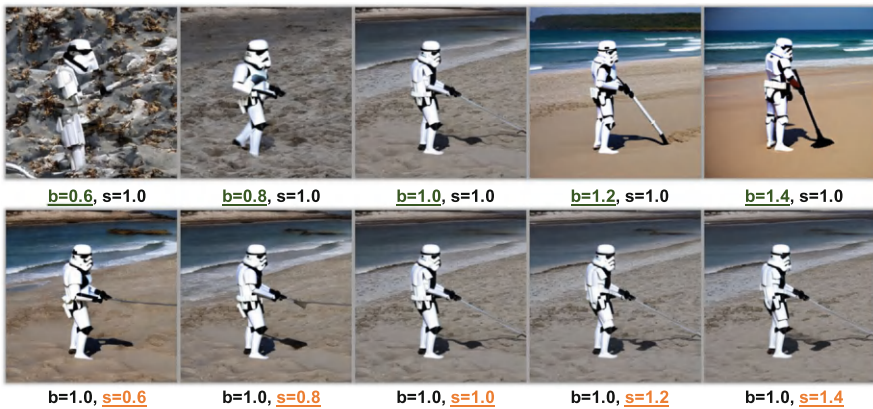


Fig. 14.5 Effect of backbone and skip connection scaling factors (b and s). Increasing the backbone scaling factor b significantly enhances image quality, while directly scaling s in the skip features has a limited influence on image synthesis quality (Image sourced from FreeU [52])

Fig. 14.6 Relative log amplitudes of Fourier with variations of the backbone scaling factor b . Increasing in b correspondingly results in a suppression of high-frequency components in the images generated by the diffusion model (Image sourced from FreeU [52])

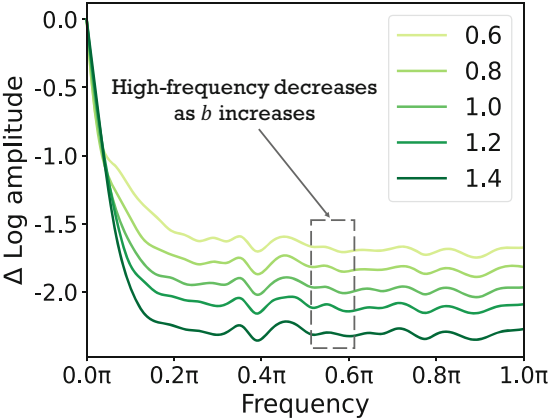
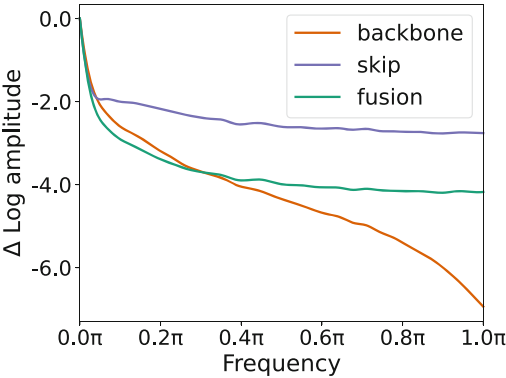


Fig. 14.7 Fourier relative log amplitudes of backbone, skip, and their fused feature maps. The skip features contain a large amount of high-frequency information (Image sourced from FreeU [52])



of the U-Net backbone network is to filter out high-frequency noise. Enhancing the backbone features effectively boosts the denoising capability of the U-Net architecture, thereby contributing to superior output in terms of fidelity and detail preservation.

The Skip Connections of U-Net Conversely, the skip connections serve to forward features from the earlier layers of encoder blocks directly to the decoder. Intriguingly, as evidenced in Fig. 14.7, these features primarily constitute high-frequency information. Our conjecture, grounded in this observation, posits that during the training of the U-Net architecture, the presence of these high-frequency features may inadvertently accelerate the convergence toward noise prediction with the optimization objective of Eq. 14.1, making it easier to reconstruct the input data. This phenomenon, in turn, could result in an unintended attenuation of the efficacy of the backbone’s intrinsic denoising capabilities. However, unlike the training process where the goal is to reconstruct input data, the inference process aims to generate data from Gaussian noise. The generative capacity of diffusion models manifests in their denoising capabilities. Therefore, during inference, it is

essential to enhance the denoising capabilities of the U-Net to ensure high-quality data generation.

14.3.4 Free Lunch in Diffusion U-Net

Capitalizing on the above discovery, this chapter propels forward with the introduction of a simple yet effective method, denoted as “**FreeU**,” which effectively bolsters the denoising capability of the U-Net architecture by leveraging the strengths of both components of the U-Net architecture. It substantially improves the generation quality without requiring additional training or fine-tuning.

The Backbone Factors To enhance the denoising capabilities of the U-Net, a method called structure-aware scaling for the backbone features is introduced, which dynamically adjusts the scaling of backbone features for each sample. Unlike a fixed scaling factor applied uniformly to all samples or positions within the same channel, structure-aware scaling adjusts the scaling factor adaptively based on the specific characteristics of the sample features. The process begins by computing the average feature map along the channel dimension:

$$\bar{x}_l = \frac{1}{C} \sum_{i=1}^C x_{l,i} \quad (14.2)$$

where $x_{l,i}$ represents the i -th channel of the backbone feature map x_l in the l -th block of the U-Net decoder. C denotes the total number of channels in x_l . As illustrated in Fig. 14.8, the average feature map $\bar{x}_l$ inherently contains valuable structural information. Consequently, the backbone factor map α_l amplifies the backbone feature map x_l in a manner that aligns with its structural characteristics. Subsequently, the backbone factor map is determined as follows:

$$\alpha_l = (b_l - 1) \cdot \frac{\bar{x}_l - \text{Min}(\bar{x}_l)}{\text{Max}(\bar{x}_l) - \text{Min}(\bar{x}_l)} + 1, \quad (14.3)$$

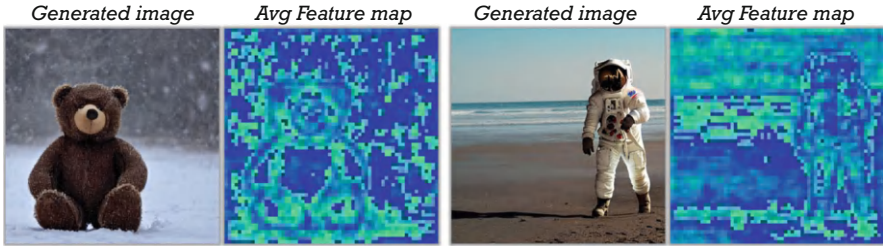


Fig. 14.8 Visualization of average feature maps: This visualization displays the average feature maps along the channel dimension of backbone features (Image sourced from FreeU [52])



Fig. 14.9 Generated images with different backbone scaling operations: (a) without backbone scaling, (b) scaling all channels, and (c) scaling half channels (Image sourced from FreeU [52])

where α_l represents the backbone factor map. b_l is a scalar constant and $b_l > 1$. Then, upon experimental investigation, we discern that indiscriminately amplifying all channels of $\mathbf{x}_l$ through multiplication with α_l engenders an oversmoothed texture in the resulting synthesized images, as shown in Fig. 14.9b. The reason is that U-Net’s strong denoising ability can damage the high-frequency details of the image during denoising. Consequently, we confine the scaling operation to the half channels of $\mathbf{x}_l$ as follows:

$$\mathbf{x}'_{l,i} = \begin{cases} \mathbf{x}_{l,i} \odot \alpha_l & \text{if } i < C/2 \\ \mathbf{x}_{l,i} & \text{otherwise} \end{cases} \quad (14.4)$$

Hence, the backbone factors can effectively enhance the denoising capabilities of the U-Net and generate better image quality, as shown in Fig. 14.9c.

The Skip Factors To further mitigate the issue of oversmoothed texture due to enhancing denoising, FreeU further employs spectral modulation in the Fourier domain to selectively diminish low-frequency components for the skip features. Mathematically, this operation is performed as follows:

$$\mathcal{F}(\mathbf{h}_{l,i}) = \text{FFT}(\mathbf{h}_{l,i}) \quad (14.5)$$

$$\mathcal{F}'(\mathbf{h}_{l,i}) = \mathcal{F}(\mathbf{h}_{l,i}) \odot \beta_{l,i} \quad (14.6)$$

$$\mathbf{h}'_{l,i} = \text{IFFT}(\mathcal{F}'(\mathbf{h}_{l,i})) \quad (14.7)$$

where $\mathbf{h}_{l,i}$ denotes the i -th channel of the skip feature map in the l -th block of the U-Net decoder. $\text{FFT}(\cdot)$ and $\text{IFFT}(\cdot)$ are Fourier transform and inverse Fourier transform. $\odot$ denotes element-wise multiplication, and $\beta_{l,i}$ is a Fourier mask,



Fig. 14.10 Generated images of FreeU without skip scaling (*w/o s*) and with skip scaling (*w/s*) (Image sourced from FreeU [52])

designed as a function of the magnitude of the Fourier coefficients, serving to implement the frequency-dependent scaling factor s_l :

$$\beta_{l,i}(r) = \begin{cases} s_l & \text{if } r < r_{\text{thresh}}, \\ 1 & \text{otherwise.} \end{cases} \quad (14.8)$$

where r is the radius. r_{thresh} is the threshold frequency, set to 1 in our experiments. As shown in Fig. 14.10, reducing low-frequency components of the skip features can generate better details.

Notably, the FreeU framework operates without the need for task-specific training or fine-tuning. Implementing the backbone and skip scaling factors is straightforward, requiring only a few lines of code. This provides a flexible and effective approach to enhancing denoising operations without introducing additional computational overhead. As a result, FreeU can be effortlessly integrated into existing diffusion models to enhance their generation quality.

14.4 Experiments

14.4.1 Implementation Details

To assess the effectiveness of the proposed FreeU, this chapter systematically conduct a series of experiments, aligning the benchmarks with state-of-the-art methods such as Stable Diffusion [43, 46], ModelScope [37], Dreambooth [47], ReVersion [23], Rerender [63], ScaleCrafter [16], Animatediff [14], and ControlNet [67]. Importantly, FreeU seamlessly integrates with these methods without imposing any additional computational overhead associated with training or fine-tuning. The experiments strictly follow the prescribed settings of these methods, incorporating only the backbone feature factors and skip feature factors during inference.

14.4.2 Image Generation

Stable Diffusion [43, 46] is a latent text-to-image diffusion model renowned for its capability to generate photorealistic images based on textual input. It has consistently demonstrated exceptional performance in various image synthesis tasks. With the integration of the FreeU augmentation into Stable Diffusion-XL [43], the results, as exemplified in Fig. 14.11, exhibit a notable enhancement in the model’s generative capacity. It becomes evident that FreeU consistently excels in generating realistic images, especially in detail generation. More results of SD [46] and SD-XL [43] are provided in the supplementary material. These compelling results serve as a testament to the substantial qualitative enhancements engendered by the synergy of FreeU with the SD[46] or SDXL[43] frameworks.

The section provides FID [17] and CLIP-score [44] and follow VBench [22] to use MUSIQ image quality predictors [30] and the LAION aesthetic predictor [34] (LAION-Aes) for quantitative assessments. MUSIQ image quality predictors have been trained on KonIQ [20], AVA [40] datasets, encompassing two evaluation metrics: MUSIQ-KonIQ and MUSIQ-AVA. The section follows prior work and evaluates text-to-image generation on the MS-COCO [35] validation set. Table 14.1 presents the results comparing FreeU with SD1.4 [46], SD2.1 [46], and SD-XL [43]. The section also provides FID [17] and CLIP-score [44] results in Table 14.2. Notably, FreeU demonstrates improvements over these powerful models. These results highlight the effectiveness of FreeU in enhancing image quality.

ScaleCrafter [16] is a training-free method designed to adapt a pre-trained diffusion model to generate images of much higher resolution than the image size used during training. As shown in Fig. 14.12, when FreeU is combined with ScaleCrafter [16],



Fig. 14.11 Text-to-image generation results of SD-XL [43] with or without FreeU. Images generated by SD-XL+FreeU show significantly improved detail and quality compared to SD-XL (Image sourced from FreeU [52])

Table 14.1 Quantitative evaluation of MUSIQ-AVA and LAION-Aes for text-to-image generation

Method	MUSIQ-AVA $\uparrow$	LAION-Aes $\uparrow$
SD 1.4 [46]	5.231	5.365
SD 1.4 + FreeU	5.563	5.532
SD 2.1 [46]	5.432	5.503
SD 2.1 + FreeU	5.686	5.612
SD-XL [43]	5.675	5.538
SD-XL + FreeU	5.994	5.776

Table 14.2 Quantitative results of FID and CLIP-score for text-to-image generation

Method	FID $\downarrow$	CLIP-sc. $\uparrow$
SD-XL [43]	43.82	0.31
SD-XL + FreeU	40.79	0.33
LCM [36]	62.03	0.30
LCM + FreeU	50.88	0.32



Fig. 14.12 4096×4096 SD-XL [43] Images generated by ScaleCrafter [16] with or without FreeU (Image sourced from FreeU [52])

the resulting combination can generate 4K images using SD-XL [43]. These images exhibit superior fine-grained details and texture quality compared to those produced solely by ScaleCrafter [16]. Consequently, FreeU serves as an effective tool for enhancing the capability of ScaleCrafter [16] in generating high-quality, higher-resolution images.

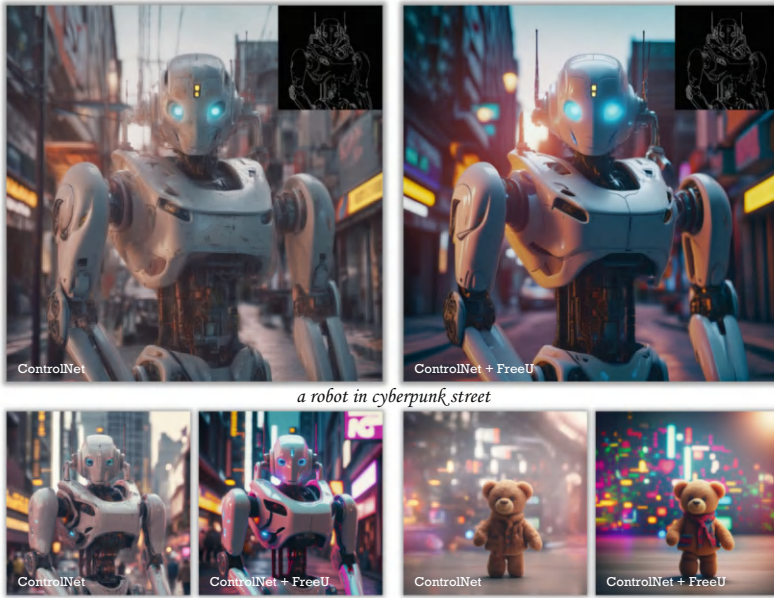


Fig. 14.13 Generated images from ControlNet [67] with and without FreeU enhancement (Image sourced from FreeU [52])

ControlNet [67] is a framework designed to introduce conditional controls to pre-trained text-to-image diffusion models. In this work, FreeU is integrated into ControlNet [67]. Figure 14.13 presents a comparison of the results. We observe a notable improvement in image quality and the presence of more detailed features in both the background and foreground when FreeU is employed alongside ControlNet [67]. These enhancements are particularly impressive given that the conditional image itself already contains a substantial amount of detailed information. This proves the effectiveness of FreeU in further enhancing the generative capabilities of ControlNet [67] in a conditional image synthesis setting.

LCM [36] is a highly efficient one-stage guided distillation method that enables few-step or even one-step inference on pre-trained latent diffusion models. The integration of the FreeU into the LCM [36] has yielded significant advancements in image generation. The comparative analysis, shown in Fig. 14.14, reveals that the use of FreeU alongside LCM [36] not only enhances image quality but also improves the detail generations.

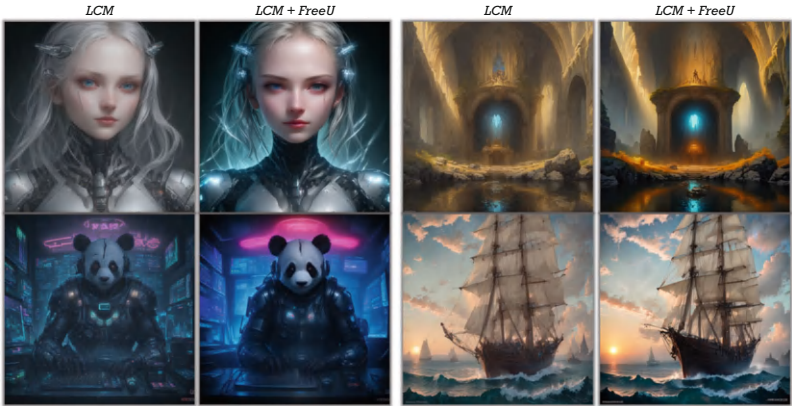


Fig. 14.14 Generated images from LCM [36] with and without FreeU enhancement (Image sourced from FreeU [52])



Fig. 14.15 Text-to-video generation results of ModelScope [37] with or without FreeU. Videos generated by ModelScope+FreeU show significantly improved appearance and motion compared to ModelScope (Image sourced from FreeU [52])

14.4.3 Video Generation

ModelScope [37], an avant-garde text-to-video diffusion model, stands at the forefront of video generation from textual descriptions. The infusion of FreeU augmentation into ModelScope [37] serves to further hone its video synthesis prowess, as substantiated by Fig. 14.15. For instance, in response to the prompt “An astronaut flying in space”, ModelScope [37], with the assistance of FreeU, can generate a clear and vivid portrayal of an astronaut. These results underscore the significant improvements achieved through the synergistic application of FreeU with ModelScope [37], resulting in high-quality generated content characterized by clear motion, rich detail, and semantic alignment.

A quantitative evaluation of FreeU for the text-to-video task is conducted, following a similar approach to the text-to-image evaluation. The results, as presented in

Table 14.3 Quantitative evaluation of MUSIQ-AVA and LAION-Aes for text-to-video generation

Method	MUSIQ-AVA $\uparrow$	LAION-Aes $\uparrow$
ModelScope [37]	4.115	4.469
ModelScope + FreeU	4.338	4.602



Fig. 14.16 Generated videos from Animatediff [14] with and without FreeU enhancement (Image sourced from FreeU [52])

Table 14.3, consistently indicate that FreeU enhances the original generation ability of ModelScope [37] across both metrics.

Animatediff [14] represents a framework designed to convert static text-to-image models into generators of animated videos. In Fig. 14.16, a comparison is made between the videos generated by Animatediff [14], with and without the incorporation of FreeU. The results demonstrate that FreeU significantly enhances the quality of each frame and ensures a higher level of consistency in appearance throughout the generated videos. For instance, when provided with a prompt “best quality, masterpiece, 1girl, looking at viewer, blurry background, upper body, contemporary, dress,” the version augmented with FreeU generates frames with a more consistent appearance of the dress across all frames.

Rerender [63] is a diffusion model tailored for zero-shot text-guided video-to-video translations. Figure 14.17 depicts the results: clear improvements in the detail and



Fig. 14.17 Generated videos from Rerender [63] with and without FreeU enhancement (Image sourced from FreeU [52])

realism of synthesized videos. For instance, when provided with the prompt “A dog wearing sunglasses” and an input video, Rerender [63] initially produces a dog video with artifacts related to the “sunglasses”. However, the incorporation of FreeU successfully eliminates such artifacts, resulting in a refined output.

14.4.4 Effects of Components in FreeU

This section evaluates the effects of the proposed FreeU strategy, *i.e.*, introducing backbone feature scaling factors and skip feature scaling factors to intricately balance the feature contributions from the backbone and skip connections. Figure 14.18 presents the results of our evaluations. In the case of $SD+FreeU(b)$, where backbone scaling factors are integrated during inference, we observe a noticeable improvement in the generation of vivid details compared to SD [46] alone. For instance, $SD+FreeU(b)$ generates a more realistic “rabbit” with normal arms and ears, as opposed to SD [46]. However, it is imperative to note that while the inclusion of feature scaling factors yields significant improvements, it can occasionally lead to an undesirable oversmoothing of textures. To mitigate this issue, skip feature

Fig. 14.18 Ablation study of backbone scaling factor b and skip scaling factor s (Image sourced from FreeU [52])

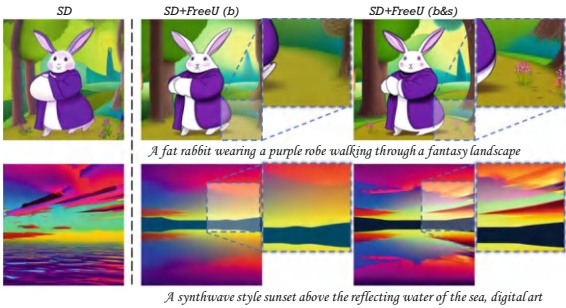
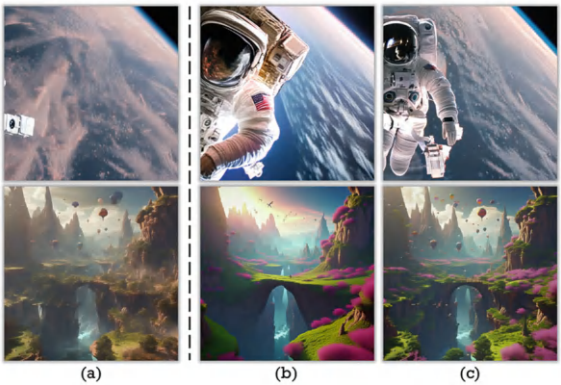


Fig. 14.19 Comparing image generation with different backbone factors: (a) SD, (b) FreeU with a constant factor, and (c) FreeU with a structure-related scaling factor map (Image sourced from FreeU [52])



scaling factors are introduced to reduce low-frequency information and alleviate the oversmoothing problem. As demonstrated in Fig. 14.18, the combination of both backbone and skip feature scaling factors in $SD+FreeU(b \ \& \ s)$ leads to the generation of more realistic images. This highlights the efficacy of FreeU strategy in balancing features and mitigating issues related to texture smoothing, ultimately resulting in more realistic image generation.

14.4.5 Effects of Backbone Structure-Related Factor

This section evaluates the effects of the proposed backbone scaling strategy, structure-related scaling, on the delicate balance between noise reduction and texture preservation. Illustrated in Fig. 14.19, when compared to the results generated by SD [46], we observe a substantial enhancement in the image quality generated by FreeU when utilizing a constant scaling factor. However, it is pertinent to highlight that the utilization of a constant factor can have undesirable consequences, manifesting as pronounced oversmoothing of textures and undesirable color oversaturation. Conversely, FreeU with the structure-related scaling factor map employs an adaptive scaling approach, leveraging structural information to



Fig. 14.20 The ablation study of backbone scaling factor b . As the backbone factor b increases, there is a noticeable enhancement in the quality of generated images. It is noteworthy, though, that excessively large values of the backbone factor may introduce oversmoothing issues (Image sourced from FreeU [52])

guide the assignment of the backbone factor map. The observations indicate that FreeU with the structure-related factor map effectively mitigates these issues and achieves significant improvements in generating vivid and intricate details.

14.4.6 Factor Selection

Backbone Factor This section conducts an evaluation to assess the effects of the backbone factor b . The results, presented in Fig. 14.20, reveal that as the backbone factor b increases, there is a noticeable enhancement in the quality of generated images. It's important to note that excessively large values of the backbone factor, such as when $b = 1.8$, can lead to oversmoothing issues. This is because increasing the backbone factor b enhances the denoising capability of U-Net, and an overly strong denoising capability compromises the preservation of high-frequency image details.

Skip Factor To mitigate the issue of oversmoothed textures resulting from enhanced denoising, this section introduces the skip factor denoted as s . This factor is employed to selectively reduce low-frequency components within the skip features. In our evaluation, as shown in Fig. 14.21, we observe that as the skip factor s decreases, the generated images exhibit more detailed backgrounds, and the oversmoothing issues are mitigated. These findings demonstrate that diminishing low-frequency components within the skip features can effectively ameliorate the

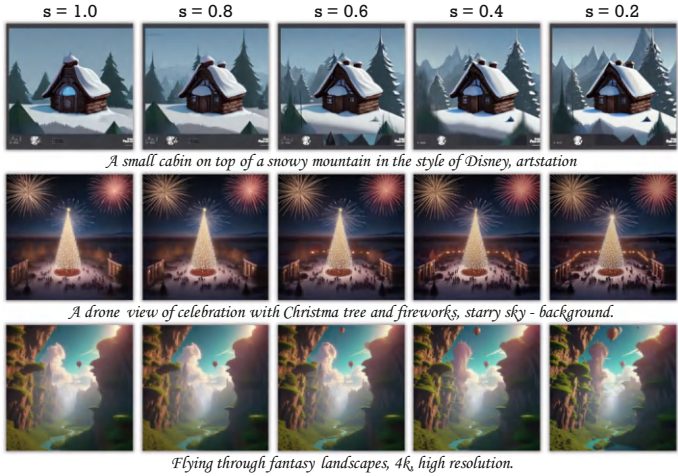


Fig. 14.21 The ablation study of skip scaling factor s . As the skip factor s decreases, the generated images exhibit more detailed backgrounds and the oversmoothing issues are mitigated (Image sourced from FreeU [52])

oversmoothing problem caused by the backbone factor. Therefore, this highlights the effectiveness of the comprehensive FreeU strategy in achieving a balance between features and alleviating issues related to texture smoothing, ultimately leading to the generation of more realistic images.

14.4.7 Channel Selection

In this section, an evaluation is conducted to investigate the impact of channel selection in the backbone scaling operation. Figure 14.22 presents the results. Figure 14.22a shows images generated using the standard SD approach, which serves as our baseline. Figure 14.22b, c, d, and e display images generated with the backbone scaling operation. We can observe that employing the backbone scaling operation contributes significantly to the improvement in image quality. However, as shown in Fig. 14.22b, applying scaling to all channels leads to oversmoothing issues, as the enhanced U-Net compromises high-frequency image details during denoising. In contrast, Fig. 14.22c, d, and e demonstrate that selecting only half of the channels for the backbone scaling operation using different methods helps mitigate the oversmoothing problem. This approach not only preserves fine-grained image details but also enhances overall image quality. Importantly, these results highlight that the specific channel selection method employed has a relatively minor impact on the generated results, as all of them contribute to the enhancement of detail generation.



Fig. 14.22 The ablation study of channel selection for backbone scaling operation. (a) The generated images of SD. (b) Scaling applied to all channels. (c) Scaling applied to the first half of the channels. (d) Scaling applied to the second half of the channels. (e) Scaling applied to a uniformly selected half of the channels (Image sourced from FreeU [52])

14.5 Conclusion

In this chapter, we commence our investigation by analyzing the process of image generation from noise. Subsequently, we delve into a detailed analysis of how the U-Net architecture implements the denoising process. Our investigation reveals that the backbone primarily contributes to denoising, while the skip connections predominantly introduce high-frequency features into the decoder, potentially leading to a neglect of essential backbone semantics. To address this, we introduce the elegantly simple yet highly effective approach, termed **FreeU**, which enhances U-Net's denoising capability by leveraging the strengths of both components of the U-Net architecture. Extensive experiments prove that FreeU can be seamlessly integrated into various diffusion foundation models and their downstream tasks and substantially improve diffusion model sample quality without additional training or fine-tuning.

Further Reading

Interested readers may refer to the following material:

- Rombach et al. [46], Podell et al. [43] for image generation with latent diffusion models
- Yang et al. [62] for a comprehensive survey on diffusion methods and their applications

References

1. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022)
2. Basri, R., Galun, M., Geifman, A., Jacobs, D., Kasten, Y., Kritchman, S.: Frequency bias in neural networks for input of non-uniform density. In: *International Conference on Machine Learning*, pp. 685–694. PMLR (2020)
3. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023)
4. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018)
5. Chen, Y., Li, G., Jin, C., Liu, S., Li, T.: Ssd-gan: Measuring the realness in the spatial and spectral domains. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1105–1112 (2021)
6. Choi, J., Kim, S., Jeong, Y., Gwon, Y., Yoon, S.: ILVR: Conditioning method for denoising diffusion probabilistic models. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021)
7. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. In: *Advances in Neural Information Processing Systems* (2021)
8. Esser, P., Rombach, R., Blattmann, A., Ommer, B.: ImageBART: Bidirectional context with multinomial diffusion for autoregressive image synthesis. In: *Advances in Neural Information Processing Systems* (2021)
9. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021)
10. Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., Holz, T.: Leveraging frequency analysis for deep fake image recognition. In: *International Conference on Machine Learning*, pp. 3247–3258. PMLR (2020)
11. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. In: *International Conference on Learning Representations* (2023)
12. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems* (2014)
13. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022)
14. Guo, Y., Yang, C., Rao, A., Wang, Y., Qiao, Y., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725* (2023)
15. He, Y., Yang, T., Zhang, Y., Shan, Y., Chen, Q.: Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221* (2022)
16. He, Y., Yang, S., Chen, H., Cun, X., Xia, M., Zhang, Y., Wang, X., He, R., Chen, Q., Shan, Y.: Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. *arXiv preprint arXiv:2310.07702* (2023)
17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: *Advances in Neural Information Processing Systems*, 30 (2017)
18. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: *Advances in Neural Information Processing Systems* (2020)

19. Hong, W., Ding, M., Zheng, W., Liu, X., Tang, J.: CogVideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868* (2022)
20. Hosu, V., Lin, H., Sziranyi, T., Saupe, D.: Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Trans. Image Process.* **29**, 4041–4056 (2020)
21. Huang, Z., Chan, K.C., Jiang, Y., Liu, Z.: Collaborative diffusion for multi-modal face generation and editing. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023)
22. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. *arXiv preprint arXiv:2311.17982* (2023)
23. Huang, Z., Wu, T., Jiang, Y., Chan, K.C., Liu, Z.: ReVersion: Diffusion-based relation inversion from images. *arXiv preprint arXiv:2303.13495* (2023)
24. Jiang, Y., Wu, T., Yang, S., Si, C., Lin, D., Qiao, Y., Loy, C.C., Liu, Z.: Videobooth: Diffusion-based video generation with image prompts. *arXiv preprint arXiv:2312.00777* (2023)
25. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: *International Conference on Learning Representations* (2018)
26. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019)
27. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020)
28. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: *Advances in Neural Information Processing Systems* (2021)
29. Kavar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276* (2022)
30. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: Multi-scale image quality transformer. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5148–5157 (2021)
31. Khayatkhoei, M., Elgammal, A.: Spatial frequency bias in convolutional generative adversarial networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 7152–7159 (2022)
32. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013)
33. Kumari, N., Zhang, B., Zhang, R., Shechtman, E., Zhu, J.Y.: Multi-concept customization of text-to-image diffusion. *arXiv preprint arXiv:2212.04488* (2022)
34. LAION-AI: aesthetic-predictor. <https://github.com/LAION-AI/aesthetic-predictor> (2022)
35. 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: *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer (2014)
36. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378* (2023)
37. Luo, Z., Chen, D., Zhang, Y., Huang, Y., Wang, L., Shen, Y., Zhao, D., Zhou, J., Tan, T.: VideoFusion: Decomposed diffusion models for high-quality video generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023)
38. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: *International Conference on Learning Representations* (2022)
39. Mirza, M., Osindero, S.: Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014)

40. Murray, N., Marchesotti, L., Perronnin, F.: Ava: A large-scale database for aesthetic visual analysis. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2408–2415. IEEE (2012)
41. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
42. Park, N., Kim, S.: How do vision transformers work? In: International Conference on Learning Representations (2021)
43. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
45. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125 (2022)
46. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
47. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)
48. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 Conference Proceedings (2022)
49. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., et al.: Photorealistic text-to-image diffusion models with deep language understanding. arXiv preprint arXiv:2205.11487 (2022)
50. Schwarz, K., Liao, Y., Geiger, A.: On the frequency bias of generative models. *Adv. Neural Inf. Process. Syst.* **34**, 18126–18136 (2021)
51. Si, C., Yu, W., Zhou, P., Zhou, Y., Wang, X., Yan, S.: Inception transformer. *Adv. Neural Inf. Process. Syst.* **35**, 23495–23509 (2022)
52. Si, C., Huang, Z., Jiang, Y., Liu, Z.: Freeu: Free lunch in diffusion u-net. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4733–4743 (2024)
53. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022)
54. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: *Advances in Neural Information Processing Systems* (2017)
55. Wang, H., Wu, X., Huang, Z., Xing, E.P.: High-frequency component helps explain the generalization of convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8684–8694 (2020)
56. Wang, P., Li, Y., Vasconcelos, N.: Rethinking and improving the robustness of image style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 124–133 (2021)
57. Wang, T., Zhang, T., Zhang, B., Ouyang, H., Chen, D., Chen, Q., Wen, F.: Pretraining is all you need for image-to-image translation. arXiv preprint arXiv:2205.12952 (2022)
58. Wang, Y., Chen, X., Ma, X., Zhou, S., Huang, Z., Wang, Y., Yang, C., He, Y., Yu, J., Yang, P., et al.: Lavie: High-quality video generation with cascaded latent diffusion models. arXiv preprint arXiv:2309.15103 (2023)
59. Wu, J.Z., Ge, Y., Wang, X., Lei, S.W., Gu, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. arXiv preprint arXiv:2212.11565 (2022)
60. Xu, Z.Q.J., Zhang, Y., Luo, T., Xiao, Y., Ma, Z.: Frequency principle: Fourier analysis sheds light on deep neural networks. arXiv preprint arXiv:1901.06523 (2019)

61. Xu, Z.Q.J., Zhang, Y., Xiao, Y.: Training behavior of deep neural network in frequency domain. In: International Conference on Neural Information Processing, pp. 264–274 (2019)
62. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M.H.: Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surveys* **56**(4), 1–39 (2023)
63. Yang, S., Zhou, Y., Liu, Z., Loy, C.C.: Rerender a video: Zero-shot text-guided video-to-video translation. *arXiv preprint arXiv:2306.07954* (2023)
64. Yang, X., Zhou, D., Feng, J., Wang, X.: Diffusion probabilistic model made slim. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22552–22562 (2023)
65. Yin, D., Gontijo Lopes, R., Shlens, J., Cubuk, E.D., Gilmer, J.: A fourier perspective on model robustness in computer vision. In: Advances in Neural Information Processing Systems, 32 (2019)
66. Zhang, D.J., Wu, J.Z., Liu, J.W., Zhao, R., Ran, L., Gu, Y., Gao, D., Shou, M.Z.: Show-1: Marrying pixel and latent diffusion models for text-to-video. *Int. J Comput. Vis.* 1–15 (2024).
67. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3836–3847 (2023)

Chapter 15

Text-Conditioned Zero-Shot 3D Avatar Creation and Animation



Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu

Abstract Creating 3D avatars has become essential in the digital landscape, yet the process remains extremely time-consuming and requires significant expertise. To make this technology accessible to a broader audience, this chapter discusses a novel framework called AvatarCLIP, which enables generation and animation of 3D avatars using text in a zero-shot manner. Unlike traditional software, which requires advanced technical knowledge, AvatarCLIP enables users with no expertise to design 3D avatars with specific shapes and textures and animate them using only descriptions in natural language. The core innovation lies in using the CLIP vision-language model to guide the generation of 3D human models, including their geometry, textures, and animations. Specifically, we begin by generating the 3D human shape through a VAE-based network, which is conditioned on textual prompts. Once the 3D shapes are produced, we apply a volume rendering approach to refine the geometry and textures. For animation, we introduce a novel method that combines motion priors learned in a motion VAE with CLIP-based guidance, enabling reference-driven motion synthesis. Comprehensive experiments demonstrate the robustness and versatility of AvatarCLIP, which can produce previously unseen avatars and animations with impressive zero-shot performance. Our code is publicly available at <https://github.com/hongfz16/AvatarCLIP>.

Keywords 3D avatars · Text-driven avatar generation · Vision-language model · Multimodal content generation

F. Hong · M. Zhang · L. Pan · Z. Liu (✉)

S-Lab, Nanyang Technological University, Singapore, Singapore

e-mail: fangzhou001@ntu.edu.sg; mingyuan001@ntu.edu.sg; liang.pan@ntu.edu.sg; ziwei.liu@ntu.edu.sg

Z. Cai

S-Lab, Nanyang Technological University and SenseTime Research, Singapore, Singapore

e-mail: caizhongang@sensetime.com

L. Yang

SenseTime Research, Hong Kong, China

e-mail: yanglei@sensetime.com

15.1 Introduction

Digital avatars have become a cornerstone in industries like film, gaming, and fashion. The creation of these avatars involves multiple complex steps: designing their shapes, applying textures, rigging skeletons, and animating them based on motion capture data. Traditionally, this process demands expertise from various specialists, the use of advanced software, significant labor, and costly equipment—resources typically available only to large enterprises. However, recent breakthroughs in the field, including large-scale pre-trained models [1] and sophisticated human representations [2], are gradually democratizing these capabilities, making them accessible to smaller studios and even individual users. In this work, we build on these advancements and introduce AvatarCLIP, a framework that can generate and animate 3D avatars entirely from natural language descriptions, as illustrated in Fig. 15.1.

Several prior efforts have sought to advance avatar generation and motion synthesis, moving toward the goal of simplifying digital avatar creation. For avatar generation, early approaches have explored 2D-GAN-based human generation [3, 4] and 3D neural human modeling [5, 6]. However, these methods often face trade-offs in terms of quality and diversity or provide limited control over the generation process. Additionally, animating these generated avatars remains a significant challenge. In the domain of motion synthesis, notable progress has been made, with models capable of generating motion sequences based on action categories [7, 8] or motion trajectories [9, 10]. However, these approaches typically rely on paired data for supervised training, which restricts the variety of generated motions and limits flexibility in the generation process. As a result, there remains a considerable gap between current methods and the vision of enabling easy and accessible digital avatar creation for a broader audience.

To simplify the complex processes involved in 3D avatar creation and animation, natural language can serve as an intuitive and accessible control signal for users. However, a significant challenge remains: there is currently no high-quality avatar-text dataset available to facilitate supervised text-based 3D avatar generation. In the realm of avatar animation, a few efforts [11, 12] have explored text-driven motion



Fig. 15.1 AvatarCLIP is an innovative pipeline for zero-shot 3D avatar generation and animation based on text input. By utilizing natural language descriptions of the avatar’s shape, appearance, and motion, AvatarCLIP efficiently produces 3D avatars with rich textures, detailed geometry, and realistic animations

generation using motion-text datasets. Yet, these fully supervised approaches are constrained by the lack of sufficient paired motion-text data, limiting their ability to generalize effectively.

Recent breakthroughs in vision-language models offer new possibilities for zero-shot, text-driven generation. CLIP [1], a vision-language model trained on large-scale image-text pairs, has demonstrated remarkable success in zero-shot text-guided image generation [13]. Building on this line of research, we leverage CLIP’s capabilities to enable zero-shot text-driven creation and animation of 3D avatars. However, CLIP cannot directly supervise the generation of 3D avatars or motion sequences, presenting significant challenges in both the creation of static avatars and the synthesis of realistic motions.

The generation of static 3D avatars presents challenges in three key areas: texture creation, geometry modeling, and ensuring that the avatar can be animated. Drawing inspiration from recent advancements in neural rendering [14], we apply CLIP supervision to rendered images, guiding the creation of an implicit 3D avatar that facilitates texture generation. To streamline the optimization process, we initialize the implicit function with a template human mesh and incorporate constraints based on this template to control the overall shape of the avatar during optimization. In addition to texture generation, high-quality geometry is essential, especially for later animation. To extract meshes from the generated implicit representations, we need a method to generate accurate geometry. A key observation is that, when examining 3D models on a computer, users often disable texture shading to obtain texture-less renderings that highlight the underlying geometry. Building on this idea, we cast random light on the surface of the implicit 3D avatar to generate texture-less renderings, which are then guided by CLIP supervision. Finally, to ensure the generated 3D avatars can be animated, we leverage recent developments in parametric human models [2]. Specifically, we align the generated avatar with a SMPL mesh and register it, enabling the avatar to be driven by the SMPL skeleton for subsequent animation.

Since CLIP is trained on static images and does not account for sequential motions, generating realistic motion sequences based solely on CLIP supervision presents a significant challenge. To address this, we decompose the process into two distinct stages: (1) generating candidate poses with CLIP guidance and (2) synthesizing smooth, coherent motion sequences using these poses as references. In the first stage, we create a codebook of diverse poses through clustering, from which CLIP selects poses that align with the given motion descriptions. These selected poses provide valuable information for the next stage. In the second stage, we employ a motion VAE to learn motion priors, which then aids in the reference-guided synthesis of smooth and valid motion sequences.

By carefully designing each stage of the pipeline, AvatarCLIP can generate high-quality 3D avatars and animate them with realistic motion sequences, all guided by natural language descriptions, as shown in Fig. 15.1. We evaluate the framework through comprehensive quantitative user studies, which assess both avatar generation and animation, demonstrating AvatarCLIP’s superiority over

existing methods. Additionally, qualitative experiments are conducted to validate the effectiveness of each individual component within our system.

15.2 Related Work

15.2.1 3D Avatar Modeling

Human modeling has been a key area of research for decades due to its broad industrial applications. With the advent of large-scale human body datasets [15], parametric models like SMPL [2] and SMPL-X [16] have been developed to represent the human body. Given its strong interpretability and seamless integration with modern graphics pipelines, we adopt SMPL as the template for our avatars. However, these models focus solely on the human body and do not account for clothing or accessories. To address this limitation, various approaches [17–20] have been proposed for modeling clothed humans.

Modeling clothing and accessories adds significant complexity to human representation, prompting the development of nonparametric approaches [21–29] that offer greater flexibility for creating realistic human models. Additionally, inspired by advancements in volume rendering, several nonparametric methods based on neural radiance fields (NeRF) [30, 31] have emerged, enabling more detailed and dynamic representations of humans [26, 32–36]. Recently, the combination of volume rendering with signed distance functions (SDF) in NeuS [37] has proven effective for achieving high-quality geometry and color reconstruction. This approach makes NeuS an ideal choice for representing the avatars in our framework.

15.2.2 3D Avatar Generation

Recent advancements in deep learning have led to remarkable progress in 2D image generation. In particular, 2D face generation [38–41] has become a well-established technology, owing to the relatively simple facial structure and the availability of large, high-quality face datasets [42]. Additionally, recent works [3, 4, 43, 44] have achieved impressive results in generating and manipulating 2D human bodies. However, animating 2D human figures remains a significant challenge due to the lack of 3D spatial understanding, which complicates the process of generating coherent and realistic movements [45–48].

3D human generation has only recently gained significant attention in research [49–51]. One notable approach is StylePeople [5], which combines the powerful SMPL-X model with StyleGAN to enable data-driven, animatable 3D avatar generation. Building on advancements in NeRF-GAN [52], a recent study [6]

introduces 3D awareness into traditional 2D generation pipelines, advancing the integration of 3D capabilities in generative models.

15.2.3 Human Motion Generation

Motion synthesis, a crucial component of animation, has been the focus of extensive research. Several large-scale datasets [53–59] provide 3D human motion data, typically represented as sequences of keypoints or SMPL parameters. The rapid development of these motion datasets has accelerated research in motion synthesis. Early approaches applied classical machine learning methods to unconditional motion generation [60–62]. More recent works, such as DVGANs [63], Text2Action [64], and Language2Pose [65], generate motions conditioned on textual descriptions using fully annotated data. Other methods, like Action2Motion [8] and Actor [7], focus on conditioning motion generation on predefined action classes. However, these approaches require large, annotated datasets, which limit their generalizability. In contrast, AvatarCLIP can generate motion directly from general natural language descriptions without relying on paired data. Additionally, some works [66, 67] explore music-conditioned motion synthesis, while others [68, 69] use physics-based approaches to generate more realistic motions by incorporating physical constraints.

15.2.4 Text-to-3D Generation

Text-to-image generation has been a long-standing area of research [70], with the ability to generalize to unseen categories demonstrated by Reed et al. [71]. The breakthrough in zero-shot text-to-image synthesis was further advanced by CLIP and DALL-E [13], which scaled up training data to unprecedented levels, unlocking impressive capabilities. Leveraging the zero-shot power of CLIP, several remarkable applications [72–74] have emerged in the field. In recent months, the integration of CLIP with 3D representations, such as NeRF or meshes, has enabled zero-shot text-driven generation [75–78] and manipulation of 3D objects [79], pushing the boundaries of what is possible with generative models.

15.3 Our Approach

Our goal is to achieve zero-shot text-driven 3D avatar generation and animation, which can be formally defined as follows. The input consists of three natural language descriptions: $\text{text} = t_{\text{shape}}, t_{\text{app}}, t_{\text{motion}}$. These texts correspond to the desired body shape, appearance, and motion of the avatar. The output has two parts:

(**a**) an animatable 3D avatar represented as a mesh $M = V, F, C$, where V are the vertices, F represent the faces, and C denote the vertex colors; (**b**) a sequence of poses $\Theta = \theta_{i=1}^L$ that define the desired motion, where L is the length of the sequence.

15.3.1 Preliminaries

CLIP [1] is a vision-language pre-trained model trained on large-scale image-text pairs. It consists of two components: an image encoder E_I and a text encoder E_T . These encoders are trained to map paired images and texts into a shared latent space, where the latent representations of matching pairs are brought closer together, while those of nonmatching pairs are pushed apart. This alignment of image and text embeddings enables zero-shot text-driven generation, where the latent code of the generated content is encouraged to align with the latent code of the corresponding text description. The CLIP-guided loss function is formally defined as

$$\mathcal{L}_{\text{clip}}(I, T) = 1 - \text{norm}(E_I(I)) \cdot \text{norm}(E_T(T)), \quad (15.1)$$

where $(\cdot)$ denotes the cosine similarity. By minimizing this loss, the generated image I is optimized to match the description provided in the text prompt T .

SMPL [2] is a widely used parametric model for human body representation, constructed from large-scale human surface scan datasets [15]. The model is expressed as $M_{\text{SMPL}}(\beta, \theta; \Phi)$, where $\beta \in \mathbb{R}^{10}$ controls the parameters of the body shape and $\theta \in \mathbb{R}^{24 \times 3}$ encodes the rotational angles for the joints. For our framework, we utilize SMPL as the foundational mesh to initialize the implicit 3D avatar. Additionally, SMPL is used to automate the skeleton binding process and facilitate animation of the generated avatar.

NeuS [37] introduces an innovative volume rendering technique that integrates the strengths of signed distance fields (SDF) and neural radiance fields (NeRF) [30]. This combination enables high-fidelity surface reconstruction and photorealistic rendering from novel viewpoints. To compute the color of a pixel corresponding to a particular view, the method accumulates color information along a ray cast from a viewpoint o in the direction v . The color is computed as

$$C(o, v) = \int_0^\infty w(t)c(p(t), v)dt, \quad (15.2)$$

where $p(t) = o + vt$ represents a point in the ray and $c(p(t), v)$ is the color at $p(t)$, which is determined by a multilayer perceptron (MLP). The function $w(t)$ is a weighting factor for the point $p(t)$, designed to be occlusion-aware and unbiased. This ensures that when optimizing on multi-view images, the model learns an

accurate signed distance field (SDF) representation. Specifically, $w(t)$ is defined as

$$w(t) = \frac{\phi_s(f(p(t)))}{\int_0^\infty \phi_s(f(p(u)))du}, \quad (15.3)$$

where ϕ_s is the logistic density function and f represents the SDF network. In our work, we use NeuS as the underlying implicit representation for generating 3D avatars.

15.3.2 Pipeline Overview

As illustrated in Figs. 15.2 and 15.3, our AvatarCLIP framework is divided into two main stages: static avatar generation and motion generation. In the first stage, the shape description t_{shape} is used to generate an initial body shape β_t . Along with a predefined standing pose θ_{stand} , this results in a template mesh $M_t = V_t, F_{\text{SMPL}}$, where V_t represents the vertices computed as $M(\beta_t, \theta_{\text{stand}}; \Phi)$ and F_{SMPL} are the

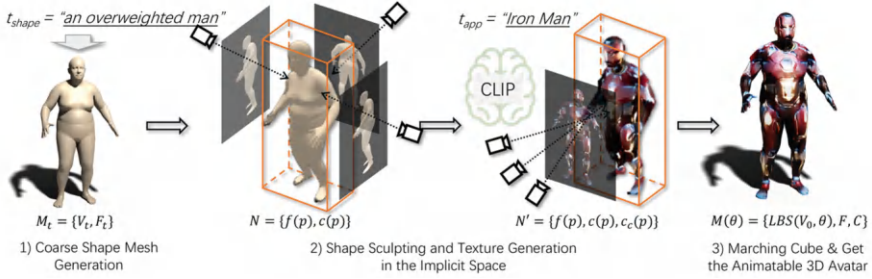


Fig. 15.2 Overview of the Static Avatar Generation Pipeline in AvatarCLIP. Suppose the user wants to generate “a tall and fat Iron Man that is running.” The pipeline first creates an animatable avatar using two input text descriptions: $t_{\text{shape}} = \text{“a tall and fat man”}$ and $t_{\text{app}} = \text{“Iron Man.”}$ The avatar is generated with these characteristics, providing both the desired body shape and the iconic appearance of Iron Man



Fig. 15.3 Overview of the Motion Generation Pipeline in AvatarCLIP. To animate the generated avatar, a motion sequence is created based on the input description $t_{\text{motion}} = \text{“running.”}$ This motion sequence is designed to match the specified action and drive the avatar’s animation accordingly

mesh faces from the SMPL model. This template mesh M_t is then rendered into multi-view images, which are used to train a NeuS model N . The output of N serves as the initial implicit representation of our 3D avatar. Guided by the appearance description t_{app} , the model N is further refined using CLIP to preserve the shape while sculpting the geometry and generating textures. Once optimized, the final 3D avatar mesh $M = V, F, C$, consisting of vertices V , faces F , and vertex colors C , is extracted from N' using the marching cubes algorithm [80] and aligned with the initial template mesh M_t to prepare it for animation. In the second stage, the motion description t_{motion} is used to generate a set of candidate poses from a pre-built codebook. These poses then act as references in the optimization of a pre-trained motion VAE, which ultimately generates the desired motion sequence for animating the avatar.

15.3.3 Static Avatar Generation

15.3.3.1 Coarse Shape Generation

To begin the avatar generation process, we create an initial template mesh, denoted as M_t , based on the input shape description t_{shape} . For this step, we leverage the SMPL model to provide a diverse range of human body shapes. As illustrated in Fig. 15.4, we first build a codebook by clustering different body shapes. Subsequently, CLIP is employed to extract features from the renderings of these shapes in the codebook, as well as from associated text descriptions, to select the body shape that best matches the input. While the procedure is conceptually simple, it requires careful design to fully harness the capabilities of these advanced tools.

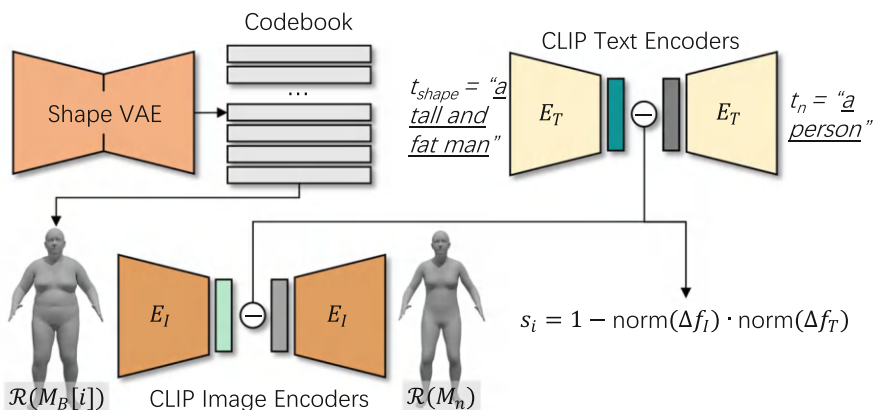


Fig. 15.4 Coarse shape generation. A shape VAE constructs a codebook for CLIP-guided querying to match the input text t_{shape} . A neutral shape M_n and text t_n serve as the anchor, with the relative latent space direction used for the query

For constructing the codebook, it is crucial to sample in a way that ensures a broad coverage of possible body shapes. The SMPL shape space, represented by $\beta \in \mathbb{R}^{10}$, is learned via PCA decomposition, which leads to a nonuniform distribution. To address this, we train a shape Variational Autoencoder (VAE) to obtain a more uniform latent space. The resulting codebook, $B_{\text{shape}} \in \mathbb{R}^{K \times d_s}$, is created by applying K-Means clustering on the latent space of the shape VAE. This codebook can then be decoded into a set of body meshes, $M_B = \{V_B, F_{\text{SMPL}}\}$, where K denotes the codebook size and d_s represents the dimensionality of the latent code.

When querying the codebook using CLIP, it is crucial to design appropriate query scores. We observe that certain attributes, such as body height, are challenging to determine solely from the renderings of a body shape without a reference. To address this, we use a reference and letting CLIP score the body shapes in a relative manner. This approach is inspired by the use of CLIP in 2D image editing tasks [74]. Specifically, we define a neutral state body shape M_n and its corresponding neutral state text description t_n as the reference. The scoring for each codebook entry s_i is given by

$$s_i = 1 - \text{norm}(\Delta f_I) \cdot \text{norm}(\Delta f_T), \quad (15.4)$$

where $\Delta f_I = E_I(\mathcal{R}(M_B[i])) - E_I(\mathcal{R}(M_n))$ and $\Delta f_T = E_T(t_{\text{shape}}) - E_T(t_n)$. Here, $\mathcal{R}(\cdot)$ represents a mesh renderer, while E_I and E_T denote the CLIP image and text encoders, respectively. The term Δf_T captures the relative direction defined by the text, which is intended to align with the visual differences Δf_I in the CLIP latent space. Finally, the entry in the codebook with the highest score is selected as the coarse shape: $M_t = M_B[\text{argmax}_i(s_i)]$.

15.3.3.2 Shape Sculpting and Texture Generation

The generated template mesh M_t serves as a coarse representation of the shape of the naked body. To create high-quality 3D avatars, further refinement of both shape and texture is required to align with the detailed description of the appearance t_{app} . As mentioned earlier, we adopt an implicit representation, specifically NeuS, as the foundational 3D model in this stage due to its strengths in capturing both geometry and color details. To accelerate the optimization process and, more crucially, to maintain control over the avatar’s overall shape, ensuring that it remains suitable for animation, we design this step as a two-stage optimization procedure.

The first stage involves creating an implicit representation of the template mesh M_t by optimizing a randomly initialized NeuS model, denoted as N , using multi-view renderings $\{I_{M_t}^i\}$ of M_t . Specifically, as illustrated in Fig. 15.5, the NeuS model $N = \{f(p), c(p)\}$ consists of two sub-networks. The signed distance function (SDF) network $f(p)$ takes a point p as input and outputs the signed distance to the nearest surface, while the color network $c(p)$ outputs the color at point p . Both $f(p)$ and $c(p)$ are implemented using multi-layer perceptrons (MLPs). Notably,

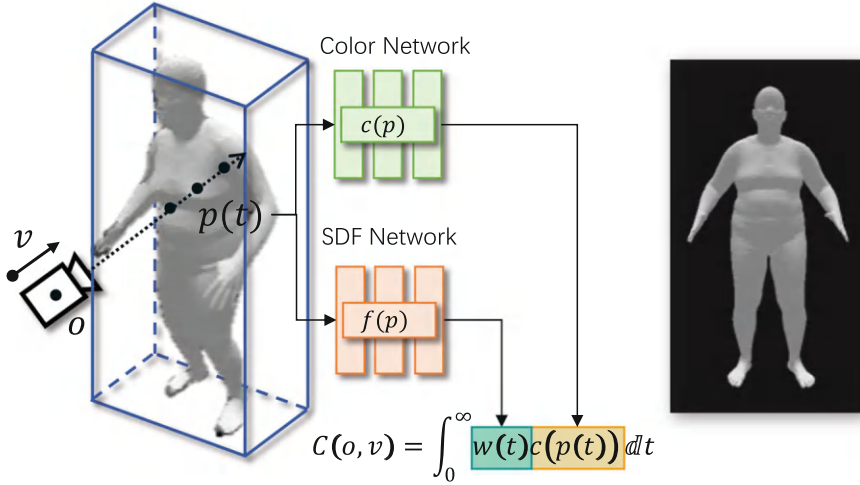


Fig. 15.5 Initialization of the implicit 3D avatar. To initialize the implicit 3D avatar, multi-view renderings of the template mesh M_t are employed to optimize a randomly initialized NeuS network N . This optimized network serves as the starting point for the avatar’s implicit 3D representation

in contrast to the original NeuS implementation, we omit the color network’s dependence on the viewing direction. This dependency was initially included to capture shading effects that change with the viewing angle. However, for our application, which aims to generate 3D avatars with consistent textures (ideally as albedo maps), such shading effects are unnecessary.

Similar to the original design, N is optimized using a three-part loss function:

$$\mathcal{L}_1 = \mathcal{L}_{\text{color}} + \lambda_1 \mathcal{L}_{\text{reg}} + \lambda_2 \mathcal{L}_{\text{mask}}, \quad (15.5)$$

where $\mathcal{L}_{\text{color}}$ is the reconstruction loss between the rendered images and the ground truth multi-view renderings $\{I_{M_t}^i\}$. The term $\mathcal{L}_{\text{reg}}$ is the Eikonal regularization term [81] applied to the SDF $f(p)$, while $\mathcal{L}_{\text{mask}}$ is a mask loss that ensures the network focuses on reconstructing only the foreground object, i.e., the template mesh M_t . After this stage, the resulting NeuS model N serves as the initialization for the second stage of optimization.

The second stage refines the initial NeuS model N from the first stage by incorporating CLIP to add stylization while preserving the coarse shape of M_t . This stage presents two potential solutions for achieving this balance. The first approach involves fixing the weights of the signed distance function (SDF) network $f(p)$, which ensures that the final generated shape remains identical to M_t . In this case, the color network $c(p)$ is optimized to “colorize” the fixed shape. However, as mentioned earlier, M_t is derived from the SMPL model, which only provides a naked human body shape. Simply adding textures to the template shape—while straightforward and computationally efficient—is not ideal. This approach overlooks

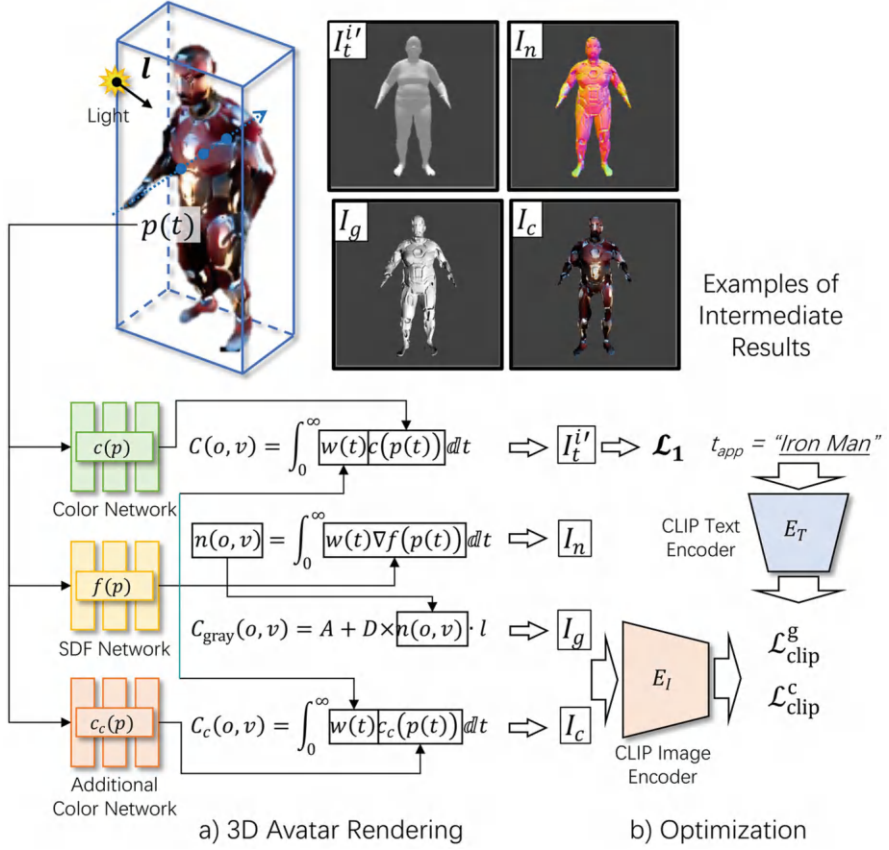


Fig. 15.6 Detailed method of shape sculpting and texture generation. For texture generation, an additional color network $c_c(p)$ is integrated into the initialized implicit 3D avatar, as depicted in (a). The optimization process incorporates three types of constraints, shown in (b), which include the reconstruction loss $\mathcal{L}_1$, CLIP-guided geometry sculpting loss $\mathcal{L}_{clip}^g$, and CLIP-guided texture generation loss $\mathcal{L}_{clip}^c$. Examples of intermediate outcomes are presented above

the importance of geometry, which is just as crucial as texture in creating a realistic avatar. Therefore, to allow for finer shape sculpting, we fine-tune the weights of $f(p)$.

Simultaneously, to preserve the overall shape of the template mesh M_t , the color network $c(p)$ should retain its original function of reconstructing the template shape, enabling the use of the reconstruction loss during optimization. This leads to the second solution, where we maintain the original two subnetworks, $f(p)$ and $c(p)$, but introduce an additional color network $c_c(p)$. As shown in Fig. 15.6, both color networks share the same SDF network $f(p)$. In this setup, $f(p)$ and $c(p)$ are responsible for reconstructing the template mesh M_t , while $f(p)$ and $c_c(p)$ together handle the stylization, ultimately forming the target implicit 3D avatar.

Formally, the updated NeuS model $N' = \{f(p), c(p), c_c(p)\}$ now consists of three subnetworks. The SDF network $f(p)$ and the original color network $c(p)$ are initialized from the pre-trained model N , while $c_c(p)$ is a randomly initialized color network. All three sub-networks are trainable. Similar to the previous setup, we omit the viewing direction dependency in the additional color network to maintain consistency in texture across views.

The second-stage optimization is guided by a three-part loss function:

$$\mathcal{L}_2 = \mathcal{L}_1 + \lambda_3 \mathcal{L}_{\text{clip}}^c + \lambda_4 \mathcal{L}_{\text{clip}}^g, \quad (15.6)$$

where $\mathcal{L}_1$ is the reconstruction loss for NeuS $\{f(p), c(p)\}$ as defined in Eq. 15.5. The additional terms $\mathcal{L}_{\text{clip}}^c$ and $\mathcal{L}_{\text{clip}}^g$ are CLIP-guided loss functions that direct the generation of texture and geometry to align with the appearance description t_{app} . These losses are computed using two distinct types of renderings, which are described in the following sections.

The first type of rendering involves generating a colored image I_c from the NeuS model $\{f(p), c_c(p)\}$, which is calculated by applying Eq. 15.2 at each pixel of the image. In contrast, the second type involves creating a textureless rendering I_g of the same NeuS model using ambient and diffuse shading techniques. To compute this, for each ray $\{o, v\}$, we first determine the normal direction $n(o, v)$ at the point where the ray intersects the surface. This is done by accumulating the gradient of the signed distance function (SDF) $f(p)$ along the ray, which can be expressed as

$$n(o, v) = \int_0^\infty w(t) \nabla f(p(t)) dt, \quad (15.7)$$

where $w(t)$ is the weighting function, as defined in Eq. 15.3. This method allows us to calculate the surface normals for the textureless rendering, which are essential for shading without relying on color information. For rendering the textureless model, it is necessary to sample a random light direction. To prevent situations where the light and camera are positioned on opposite sides of the model—thus preventing the geometry details from being visible—the light direction is sampled within a small range around the camera’s direction. Specifically, we define the camera direction in spherical coordinates using the polar and azimuthal angles $\{\theta_c, \phi_c\}$. The light direction l is then sampled as $\{\theta_c + X_1, \phi_c + X_2\}$, where $X_1, X_2 \sim \mathcal{U}(-\pi/4, \pi/4)$, ensuring that the light is within a small angular range around the camera. Since no color is required for the textureless rendering, we compute the gray-level intensity of the ray $\{o, v\}$ using the following equation:

$$C_{\text{gray}}(o, v) = A + D \times n(o, v) \cdot l, \quad (15.8)$$

where $A \sim \mathcal{U}(0, 0.2)$ is a random value drawn from a uniform distribution and $D = 1 - A$ represents the diffuse component. This formula calculates the gray level of the surface based on its normal $n(o, v)$ and the sampled light direction l . By applying Eq. 15.8 to each pixel, we obtain the textureless rendering I_g . In practice,

we observe that adding random shading to textured renderings I_c improves the uniformity of the generated textures, as we will discuss further in the ablation study. This random shading process is conceptually similar to the textureless rendering and is mathematically defined as

$$C_{shade}(o, v) = A + D \times n(o, v) \cdot I * C(o, v), \quad (15.9)$$

where $C(o, v)$ represents the color at point (o, v) . This shading adds variation to the texture, enhancing the quality of the final texture generation. The two CLIP-guided loss functions used in the optimization are then formally defined as $\mathcal{L}_{clip}^c = \mathcal{L}_{clip}(I_c, t_{app})$ for the colored renderings and $\mathcal{L}_{clip}^g = \mathcal{L}_{clip}(I_g, t_{app})$ for the textureless renderings. These losses guide the network to adjust both geometry and texture to better match the target appearance description t_{app} .

To render an image of size $H \times W$, a total of $H \times W \times Q$ queries need to be performed, where Q represents the number of query points per ray. Due to the high memory demand of volume rendering, the resolution $H \times W$ is significantly constrained. In our experiments, conducted on a 32 GB GPU, the maximum achievable resolution, H_{max} and W_{max} , is approximately 110, which is much lower than the resolution limit of 224 imposed by CLIP. To overcome this limitation and increase the rendering resolution, we use a dilated silhouette-based rendering strategy, based on the observation that rays which do not intersect any surface contribute nothing to the final image, yet still consume valuable memory.

Our approach involves identifying rays that are likely to hit a surface by calculating the silhouette of the template mesh M_t with the given camera parameters. The silhouette is then dilated to ensure that sufficient space for shape sculpting is preserved. Only rays within the dilated silhouette are calculated and contribute to the final rendering, reducing the overall memory usage. Let the ratio between the area of the dilated silhouette and the total image area be denoted as r_s . Using this strategy, we can dynamically increase the maximum resolution to $H'_{max} \times W'_{max} = \frac{H_{max} \times W_{max}}{r_s}$. Empirically, this method allows us to push the resolution up to around 150^2 , significantly improving the quality of the rendered images while managing memory constraints effectively.

To further enhance the robustness of the optimization process, we discuss three augmentation strategies: **(a)** random background augmentation, **(b)** random camera parameter sampling, and **(c)** semantic-aware prompt augmentation. Each of these strategies is designed to improve the diversity and generalization of the model during training, and they are detailed below.

Inspired by Dream Fields [76], the random background augmentation strategy is designed to help CLIP focus more on the foreground object and prevent the generation of floating or disconnected volumes. As illustrated in Fig. 15.7, we randomly modify the backgrounds of the renderings with the following options: (1) a pure black background, (2) a pure white background, (3) random Gaussian noise, and (4) a Gaussian-blurred chessboard pattern with randomly varying block sizes.



Fig. 15.7 Examples of four types of random background augmentations

This variation in backgrounds encourages the model to prioritize the foreground object and improves the quality and stability of the generated avatars.

To prevent the network from finding shortcut solutions that only work for a few fixed camera positions, we randomly sample the camera extrinsic parameters during each optimization iteration. This is done using a manually-defined importance sampling approach, ensuring more coherent and smoother results. Specifically, we use the “look-at” camera mode, which consists of a look-at point, a camera position, and an up direction. The up direction is fixed to always align with the up direction of the template body to maintain consistency.

The look-at position is sampled from a Gaussian distribution, $X, Y, Z \sim \mathcal{N}(0, 0.1)$ and then clipped between -0.3 and 0.3 to ensure that the avatar remains within the frame. For the camera position, we employ a spherical coordinate system, where the radius R is sampled from a uniform distribution $\mathcal{U}(1, 2)$. The polar angle θ_c is also sampled uniformly from $\mathcal{U}(0, 2\pi)$, while the azimuthal angle ϕ_c is sampled from a Gaussian distribution $\mathcal{N}(0, \pi/3)$ to keep the camera predominantly focused on the front side of the avatar during most of the iterations. This approach helps to avoid overfitting to specific viewpoints and encourages the model to generate more versatile and stable results.

To address the potential issue of generating textures at incorrect or missing body parts, we incorporate human priors into the optimization process by augmenting the prompts in a semantic-aware manner. Without such priors, there is a risk that the network may produce textures inappropriately or neglect important areas of the body, as will be demonstrated in the ablation study. To mitigate this, we explicitly guide the generation of key body parts by augmenting the appearance description t_{app} with additional, semantically specific prompts.

As shown in Fig. 15.8, if t_{app} is “Steve Jobs,” we augment this prompt with two additional ones: $t_{\text{face}} = \text{“the face of Steve Jobs”}$ and $t_{\text{back}} = \text{“the back of Steve Jobs.”}$ Every four iterations, we adjust the camera’s look-at point to the center of the face, ensuring that renderings of the face are generated, which are then supervised by the t_{face} prompt. This “face augmentation” directly supervises the generation of

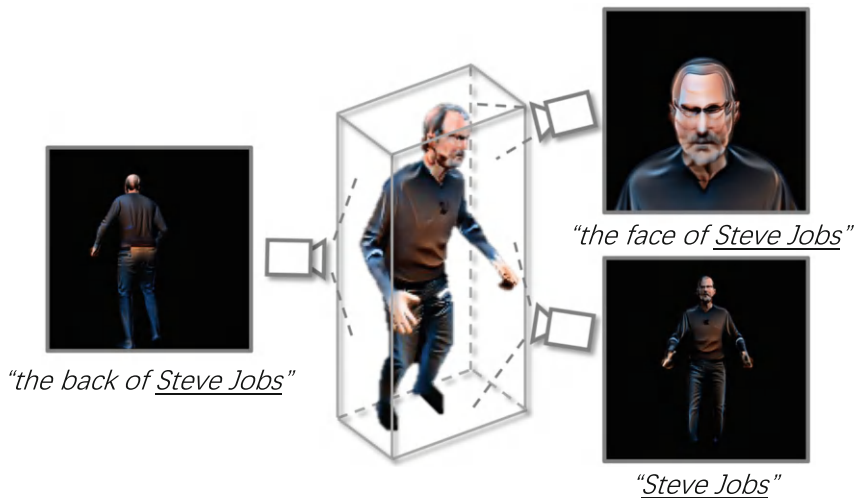


Fig. 15.8 An example of prompt augmentations: one designed for the detailed refinement of the face and the other focused on enhancing the back side of the avatar

the face, which is critical for the avatar’s quality, as humans are particularly sensitive to facial features.

Similarly, the second augmented prompt, t_{back} , is used to guide the generation of the avatar’s back. When the camera is oriented toward the back of the avatar, the t_{back} prompt is applied to explicitly supervise this region. This semantic-aware prompt augmentation helps ensure that essential parts of the avatar, like the face and back, are properly generated, improving the overall quality and accuracy of the 3D avatar.

15.3.3.3 Make the Static Avatar Animatable

Once the implicit 3D avatar N' is generated, we apply the marching cubes algorithm to extract its mesh representation $M = \{V, F, C\}$, where V represents the vertices, F the faces, and C the colors. Before making the avatar animatable, we ensure that its mesh is well-aligned with the initial template mesh, thanks to the careful optimization design that preserves the overall shape. The alignment and animation process proceeds as follows: First, for each vertex in V , we find its nearest neighbor in the vertices of the template mesh M_t . The blend weights for each vertex in V are then copied from the corresponding nearest vertex in M_t . This step ensures that the geometry of the generated mesh is aligned with the template and ready for deformation. Next, an inverse linear blend skinning (LBS) algorithm is applied to transform the avatar’s standing pose θ_{stand} back to the neutral zero pose θ_0 . This is done by applying the inverse of the skinning transformation to the vertices V , resulting in V_{θ_0} , which represents the mesh in the zero pose. Finally, V_{θ_0} can

be animated by applying any desired pose θ using the standard LBS algorithm. This allows us to drive the avatar into any pose while maintaining the appropriate deformations and blend weights. As a result, the animated avatar for any given pose can be formally defined as

$$M(\theta) = (\text{LBS}(V_{\theta_0}, \theta), F, C) \quad (15.10)$$

This ensures that the avatar remains both anatomically correct and smoothly deformable across various poses.

15.3.4 Motion Generation

Empirically, CLIP is not well-suited to directly assess the similarity between motion sequences and natural language descriptions. Additionally, it lacks the ability to evaluate the smoothness or coherence of motion sequences. These limitations make it challenging to generate realistic motion solely based on CLIP supervision. As a result, we need to incorporate additional modules to provide motion priors. However, CLIP excels at evaluating the similarity between a rendered human pose and a natural language description. This suggests that we can use CLIP to guide the generation of individual poses, and similar poses can serve as references for generating motion sequences. Based on these insights, we discuss a two-stage motion generation process: (1) candidate pose generation guided by CLIP and (2) motion sequence generation using motion priors, where the candidate poses are used as references. The process is detailed as follows.

15.3.4.1 Candidate Pose Generation

To create poses that align with the specified description t_{motion} , a straightforward approach is to directly optimize the parameter θ within the SMPL model or to adjust the latent code of a pre-trained pose VAE (e.g., VPoser [16]). However, this method often fails to generate realistic poses due to challenges encountered during the optimization process, as demonstrated in our experimental results. Consequently, directly optimizing the poses is not an ideal strategy.

As illustrated in Fig. 15.9, to circumvent direct optimization, we begin by constructing a codebook using the AMASS dataset [56]. To reduce the dimensionality, we leverage VPoser to encode the poses into a latent space $z \in \mathbb{R}^{d_p}$. Next, K-means clustering is applied to identify K centroids, which together form our pose codebook $B_z \in \mathbb{R}^{K \times d_p}$. Each entry in B_z is subsequently decoded by VPoser, resulting in a set of corresponding poses B_θ .

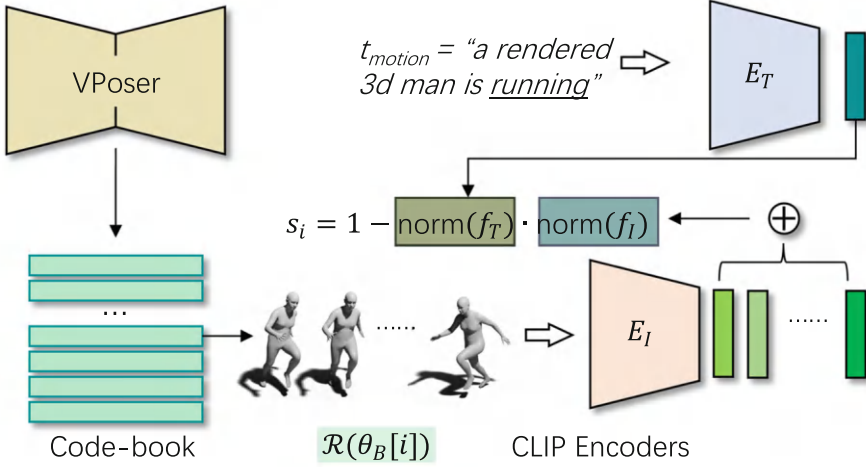


Fig. 15.9 Detailed pipeline of candidate pose generation. The process begins by utilizing the pre-trained VPoser to create a codebook. For a given text description t_{motion} , each pose feature f_I in the codebook is compared with the text feature f_T to compute their similarity. The Top-k most similar entries are then selected as the candidate poses

Given the motion description t_{motion} , we compute the similarity between t_{motion} and each pose $B_\theta[i]$ from the codebook B_θ , which is defined as follows:

$$s_i = 1 - \text{norm}(E_I(\mathcal{R}(\theta_B[i]))) \cdot \text{norm}(E_T(t_{\text{motion}})). \quad (15.11)$$

The top-k similarity scores s_i and their corresponding poses $B_\theta[i]$ are then selected to form the candidate pose set S . This set S will serve as a reference for generating a motion sequence in the subsequent stage.

15.3.4.2 Reference-Based Animation with Motion Prior

We introduce a two-step approach to generate a target motion sequence that aligns with the motion description t_{motion} . The first step involves training a motion VAE to learn human motion priors. In the second step, we optimize the latent code of the motion VAE, using the candidate poses S as reference points. The detailed procedure is explained below.

Motion VAE Pre-train We draw inspiration from Actor [7] in designing the motion VAE. Specifically, as illustrated in Fig. 15.10, the motion VAE consists of three main components: the motion encoder E_{motion} , a reparameterization module, and the motion decoder D_{motion} . A motion sequence is represented as $\Theta^+ \in \mathbb{R}^{L \times 24 \times 6}$, where L denotes the length of the sequence. Each joint is encoded as a continuous six-dimensional vector [82].

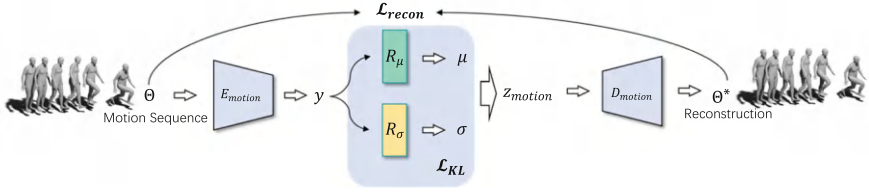


Fig. 15.10 Structure of the motion VAE. The motion VAE consists of three components: the encoder E_{motion} , the decoder D_{motion} , and a reparameterization module. During training, the motion VAE is optimized using the reconstruction loss $\mathcal{L}_{\text{recon}}$ along with the KL-divergence term $\mathcal{L}_{\text{KL}}$.

The motion encoder E_{motion} consists of several key components: a projection layer, a positional embedding layer, multiple transformer encoder layers, and an output layer. Formally, E_{motion} can be expressed as

$$y = E_{\text{motion}}(\Theta') = E_o(\text{TransEncoder}(\phi(E_p(\Theta')))). \quad (15.12)$$

Here, E_p and E_o represent fully connected layers, and ϕ denotes the positional embedding operation [83]. The TransEncoder refers to a series of transformer encoder layers [83].

The reparameterization module generates a Gaussian distribution where the mean $\mu = R_\mu(y)$ and the standard deviation $\sigma = R_\sigma(y)$. Here, R_μ and R_σ are fully connected layers that compute the mean and standard deviation of the distribution, respectively. Using the reparameterization trick [84], a random latent code z_{motion} is sampled from the distribution $\mathcal{N}(\mu, \sigma)$. This latent code is then decoded by D_{motion} , which can be formally expressed as

$$\Theta^* = D_{\text{motion}}(z_{\text{motion}}) = D_o(\text{TransDecoder}(D_p(z_{\text{motion}}), \phi(\mathbf{0}))). \quad (15.13)$$

In this equation, D_p and D_o are fully connected layers, $\mathbf{0}$ represents a zero vector, and TransDecoder consists of several transformer decoder layers.

To train the motion VAE, a loss function with two components is proposed, defined as

$$\mathcal{L}_{\text{mVAE}} = \lambda_5 \cdot \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{recon}}, \quad (15.14)$$

where λ_5 is a hyperparameter that balances the two terms. The first term, $\mathcal{L}_{\text{KL}}$, computes the KL-divergence to enforce the assumed distribution. The second term, $\mathcal{L}_{\text{recon}}$, is the mean squared error between the target motion sequence Θ^+ and the reconstructed motion sequence Θ^* .

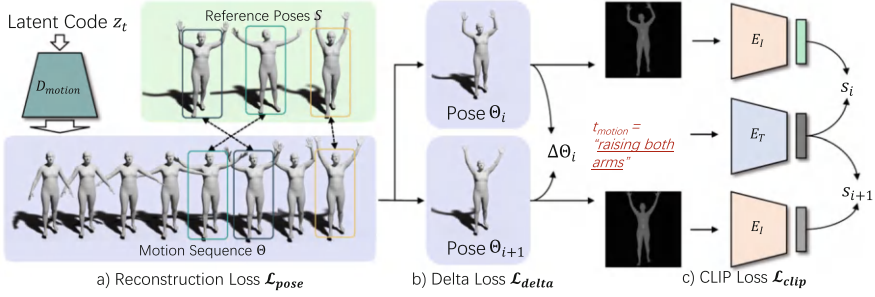


Fig. 15.11 Detailed pipeline of reference-based animation with motion prior. To optimize the latent code z_t , three constraint terms are introduced. For the motion sequence Θ generated by D_{motion} , the loss term $\mathcal{L}_{pose}$ ensures that the distance between each candidate pose and the closest pose in Θ is minimized. The loss term $\mathcal{L}_{delta}$ quantifies the differences between adjacent poses and offers control over the motion intensity. Lastly, $\mathcal{L}_{clip}^m$ computes the similarity between the text description t_{motion} and each pose in Θ

Optimization on the Motion VAE Using the pre-trained motion VAE, we aim to optimize its latent code z_t in order to synthesize a motion sequence $\Theta = D_{motion}(z_t)$. As shown in Fig. 15.11, three optimization constraints are defined as

$$\mathcal{L}_{motion} = \mathcal{L}_{pose} + \lambda_6 \mathcal{L}_{delta} + \lambda_7 \mathcal{L}_{clip}^m, \quad (15.15)$$

where λ_6 and λ_7 are hyperparameters used to balance the individual terms. The first term, $\mathcal{L}_{pose}$, corresponds to the reconstruction loss between the generated motion sequence Θ and the candidate poses S . The second term, $\mathcal{L}_{delta}$, measures the range of motion to prevent the sequence from becoming overly smoothed. Finally, $\mathcal{L}_{clip}^m$ ensures that each pose in the motion sequence aligns with the input motion description. The details of these three loss terms are explained below.

Given the reference poses $S = \{\theta_1, \theta_2, \dots, \theta_k\}$, the goal is to construct a motion sequence that closely matches these poses. To achieve this, we minimize the distance between each reference pose θ_i and its closest frame Θ_j , where $i \in \{1, 2, \dots, k\}$ and $j \in \{1, 2, \dots, L\}$. We assume that the reference poses become less similar to the motion description t_{motion} as i increases. Therefore, we introduce a weighting coefficient $\lambda_{pose}(i) = 1 - \frac{i-1}{k}$ to place more emphasis on reference poses with higher similarity. Formally, the reconstruction loss is defined as

$$\mathcal{L}_{pose} = \sum_{i=1}^k \lambda_{pose}(i) \min_j \{\|\theta_i - \Theta_j\|\}. \quad (15.16)$$

Empirically, relying solely on the reconstruction loss $\mathcal{L}_{pose}$ leads to generated motions that are overly smoothed. To encourage motions with a larger range of movement, we introduce a motion range term, $\mathcal{L}_{delta}$, which penalizes the

smoothness between adjacent poses. This term is defined as

$$\mathcal{L}_{\text{delta}} = - \sum_{i=1}^{L-1} \|\Theta_i - \Theta_{i+1}\|. \quad (15.17)$$

The negative sign ensures that larger changes between consecutive poses are encouraged. This term acts as a penalty against overly smoothed motions, and increasing λ_6 will result in more dynamic, intense motions.

The reconstruction term $\mathcal{L}_{\text{pose}}$ does not ensure the correct ordering of candidate poses, and the absence of supervision on pose sequencing can lead to unstable and inconsistent generation results. Moreover, since candidate poses may only represent a small fraction of the final motion sequence, this could result in unexpected or unrealistic motion segments. To address these issues, we introduce an additional CLIP-guided loss term, defined as

$$\mathcal{L}_{\text{clip}}^m = \sum_{i=1}^L \lambda_{\text{clip}}(i) \cdot s_i, \quad (15.18)$$

where s_i represents the similarity score between the pose Θ_i and the text description t_{motion} , calculated as

$$s_i = 1 - \text{norm}(E_I(\mathcal{R}(\Theta_i))) \cdot \text{norm}(E_T(t_{\text{motion}})). \quad (15.19)$$

Here, $\lambda_{\text{clip}}(i) = \frac{i}{L}$ is a monotonically increasing function, which ensures that later poses in the sequence incur a higher penalty in the optimization process. By including this CLIP-guided loss term, we promote a more consistent alignment of the entire motion sequence with the text description t_{motion} . Empirically, we observe that calculating this term for only a small subset of poses within Θ suffices for effective optimization, resulting in a significant reduction in computation time without noticeable degradation in performance.

15.4 Experiments

15.4.1 Experimental Setup

15.4.1.1 Implementation Details

The shape VAE (described in Sect. 15.3.3.1) employs a two-layer MLP architecture, with a latent space dimension of 16 for both the encoder and decoder. To construct the codebook, 40,960 latent codes are drawn from a Gaussian distribution and subsequently grouped into 2,048 centroids using K-means clustering. In the process of shape sculpting and texture generation (Sect. 15.3.3.2), we modify the original

NeuS model so that the SDF network is built using a six-layer MLP, while the color network is designed with a four-layer MLP. For each ray, we conduct 32 uniform samples and 32 importance samples. The optimization is carried out using the Adam algorithm [85] with a learning rate of 5×10^{-4} for 30,000 iterations.

For generating candidate poses (as described in Sect. 15.3.4.1), we leverage the pre-trained VPoser model [16] and apply K-means clustering to derive 4,096 centroids from the AMASS dataset [56]. From these, we select the top five poses to proceed to the next stage (Sect. 15.3.4.2). The motion VAE utilizes a latent space of 256 dimensions, with a motion sequence length of 60 frames. We train the motion VAE for 100 epochs on the AMASS dataset. For optimization, the Adam optimizer is used with a learning rate of 5×10^{-4} . When refining the latent code for t_{motion} , Adam is applied for 5,000 iterations, with a higher learning rate of 1×10^{-2} .

15.4.1.2 Baselines

Although this is the first approach to generate and animate 3D avatars in a zero-shot, text-driven fashion, we establish thoughtful baseline methods to evaluate each component of AvatarCLIP. For coarse shape generation, we discuss a baseline where shape parameters (such as the SMPL β or the latent code of the shape VAE) are optimized using CLIP-guided losses. In the case of shape sculpting and texture generation, we compare our method to Text2Mesh [75] and the modified Dream Field approach [76]. Our design incorporates an “additional color network,” which is adapted to the Dream Field framework to impose constraints on the overall shape of the avatar.

For candidate pose generation, we design three baseline methods to benchmark against our approach. To highlight the challenges of direct optimization, two of these baselines focus on directly optimizing the SMPL parameter θ and the latent code z in VPoser. Additionally, inspired by CLIP-Forge [77], we incorporate Real NVP [86] to establish a bidirectional mapping between the multivariate normal distribution and the latent space distribution of VPoser. This normalization flow network is conditioned on CLIP features, with rendered image features used during training and textual description features used during inference. Notably, this method does not require paired pose-text data.

For the motion generation phase, we design two additional baseline methods for comparison. Baseline (i) begins by sorting the candidate poses S based on their similarity scores s_i , followed by direct interpolation of the latent codes between adjacent poses to create the motion sequence. Baseline (ii) utilizes the motion VAE to inject motion priors into the generative pipeline but computes the reconstruction loss directly, without applying any re-weighting techniques.



Fig. 15.12 Overall results of AvatarCLIP. A series of renderings showcasing animated 3D avatars are presented in sequence. Below each sequence, the corresponding driving texts for shape, appearance, and motion are provided

15.4.2 Overall Results

The overall performance of the AvatarCLIP pipeline is presented in Fig. 15.12. The system generates avatars with a wide variety of body shapes and appearances, all of which exhibit high visual quality. These avatars are animated using motion sequences that align well with the input descriptions, ensuring both realism and consistency. In a zero-shot manner, AvatarCLIP is able to create fully animatable avatars and corresponding motions by leveraging the rich priors encoded in pre-trained models. What traditionally required expert knowledge of professional software tools can now be effortlessly controlled through natural language commands, thanks to the capabilities of AvatarCLIP.

15.4.3 Experiments on Avatar Generation

15.4.3.1 Ablation Study

To assess the impact of different design choices in the avatar generation module, we conduct a series of ablation studies. We evaluate the effects of four key components: (1) background augmentation, (2) supervision on textureless renderings, (3) random shading on textured renderings, and (4) semantic-aware prompt augmentation. The ablation configurations, as depicted in Fig. 15.13, involve sequentially adding these elements to a baseline method where only textured renderings are supervised by CLIP.

As shown in the first two columns of Fig. 15.13, incorporating background augmentation significantly enhances texture generation. Without this component,

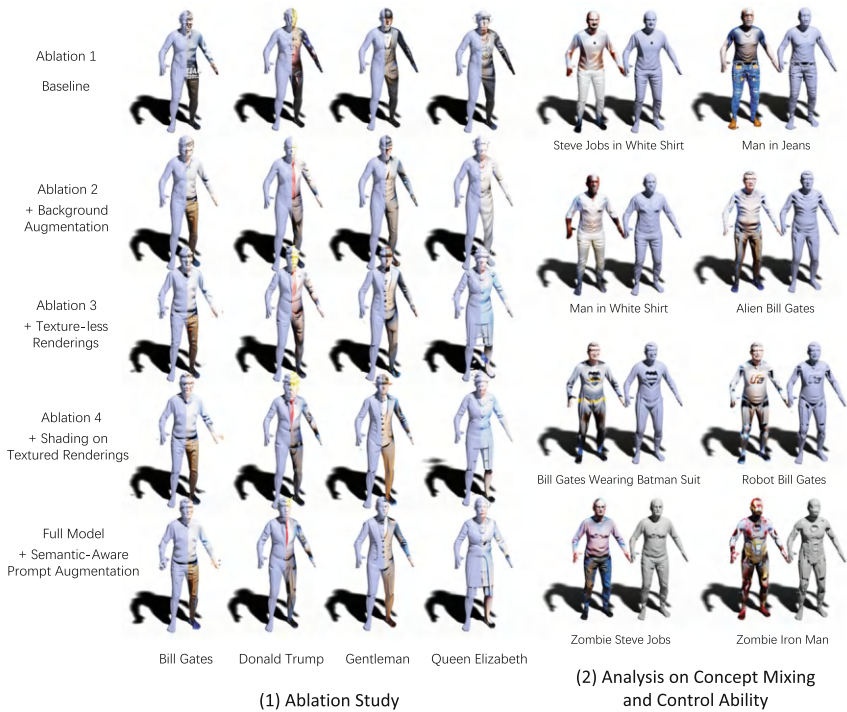


Fig. 15.13 Ablation study on static avatar generation. We conduct four ablation studies to evaluate the impact of various design decisions in avatar generation. The ablation settings progressively introduce: (1) background augmentation, (2) textureless renderings, (3) shading on textured renderings, and (4) semantic-aware prompt augmentation. Additionally, we present several results of concept-mixed avatar generation, showcasing the natural blending of different concepts. This further demonstrates the versatility and generalizability of our approach. The input texts corresponding to each rendering are provided below

the generated textures tend to appear excessively dark. Comparing the second and third columns, we see that introducing supervision on textureless renderings greatly improves the geometry, resulting in much cleaner shapes. The geometry in “Ablation 2” exhibits random bumps, creating a noisy surface, while “Ablation 3” produces smoother geometry with well-defined garment wrinkles. The addition of random shading to textured renderings, as seen in “Ablation 3” and “Ablation 4,” results in more consistent and uniform textures. For instance, in “Ablation 3,” the upper body of the “Donald Trump” avatar appears significantly brighter than the lower body, a discrepancy resolved in “Ablation 4.” Furthermore, without the inclusion of human body semantics, the earlier settings fail to generate accurate facial features. However, the final column, which employs semantic-aware prompt augmentation, yields the most accurate and realistic face generation.

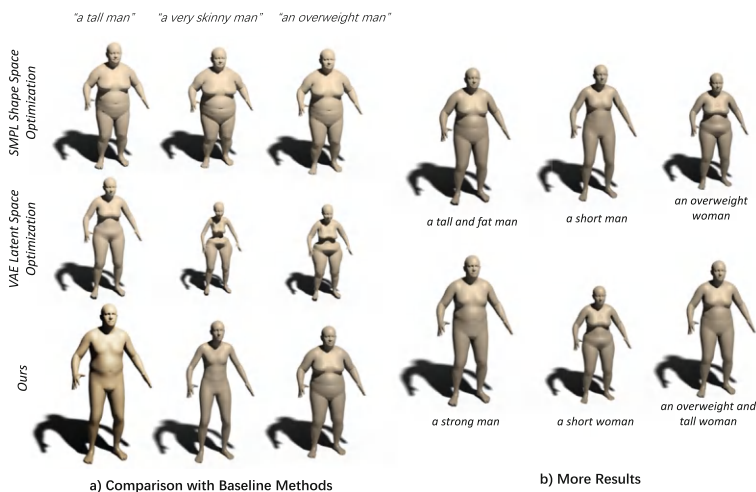


Fig. 15.14 Results of the coarse shape generation. (a) A qualitative comparison is made between our approach and two baseline methods: (i) direct optimization in the SMPL β space and (ii) direct optimization in the VAE latent space. Both baseline methods fail to produce plausible body shapes, while our method effectively generates body shapes that align with the provided descriptions. (b) Additional results of the coarse shape generation are presented for further comparison

15.4.3.2 Qualitative Results of Coarse Shape Generation

In this section, we discuss two straightforward baseline methods in which direct CLIP supervision is back-propagated to the shape parameters. As illustrated in Fig. 15.14a, both baseline methods fail to generate body shapes that align with the given textual descriptions. Notably, even when opposing descriptions are provided (e.g., “skinny” vs. “overweight”), both methods lead to the same optimization direction. In contrast, our approach consistently produces body shapes that accurately reflect the input text. Additional qualitative results showcasing the diversity of body shapes generated by our method can be found in Fig. 15.14b (Fig. 15.15).

15.4.3.3 Qualitative Results of Shape Sculpting and Texture Generation

Broad Range of Driving Texts Through extensive experiments, we demonstrate that our method can generate avatars based on a diverse set of appearance descriptions, encompassing three categories: (1) celebrities, (2) fictional characters, and (3) general descriptors of people, as shown in Fig. 15.16. In Fig. 15.16a, when given celebrity names as appearance descriptions, the method reliably generates avatars dressed in the celebrity’s most recognizable outfit. Thanks to the semantic-aware prompt augmentation design, the faces are also accurately rendered. For generating fictional characters, as shown in Fig. 15.16b, our method successfully creates avatars

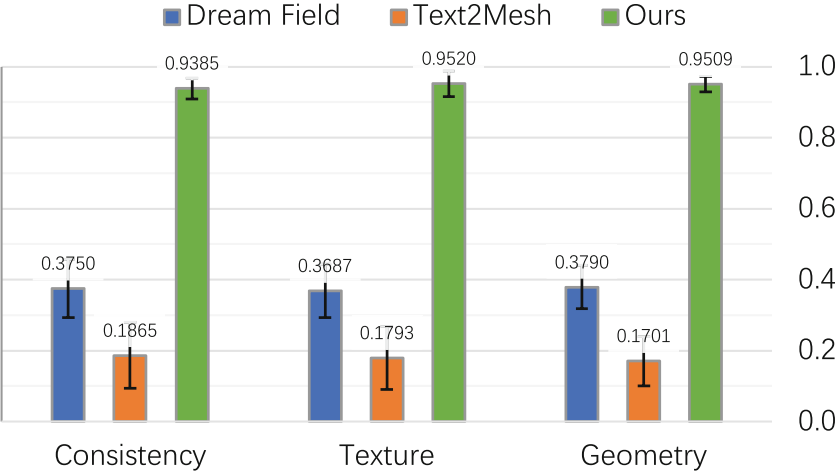


Fig. 15.15 User study on static avatar generation provides quantitative evidence of the superiority of AvatarCLIP over two baseline methods across three criteria: (1) consistency with the input text, (2) texture quality, and (3) geometry quality

that match the descriptions. Notably, for characters with complex accessories (e.g., Batman’s helmet or Elsa’s dress), the optimization process tends to “grow” new elements that integrate naturally with the human body template. In terms of general descriptions, our approach handles a wide range of input types, from common job titles (e.g., “professor,” “doctor”) to age-specific terms (e.g., “teenager,” “senior citizen”) and even imaginative professions (e.g., “witch,” “wizard”). An interesting observation is that not only can the avatars themselves be generated but also the iconic objects associated with them. For instance, the avatar of a “gardener” is depicted holding flowers and grass, reflecting the profession’s key attributes.

Zero-Shot Controlling Ability Beyond generating avatars based on broad appearance descriptions, our method also enables zero-shot control at a finer level of detail. As demonstrated in Fig. 15.13, we can adjust the faces of generated avatars—such as depicting Bill Gates in an Iron Man suit—by modifying the semantic-aware prompt augmentation. Additionally, we can fine-tune the avatar’s clothing using direct textual instructions, such as specifying “Steve Jobs in a white shirt” to shape the avatar’s outfit.

Geometry Quality The key contribution of textureless rendering supervision lies in its impact on geometry generation. We compare our method, AvatarCLIP, with the adapted version of Dream Field, which is based on NeRF. As demonstrated in Fig. 15.17, our approach consistently produces superior geometry quality compared to Dream Fields. It is capable of generating fine details, such as muscle definition, armor contours, and cloth wrinkles, with higher accuracy.



Fig. 15.16 Results of static avatar generation. A diverse set of driving texts is evaluated, including celebrity names, fictional character names, and general descriptions. The generated avatars are presented with and without textures, allowing for easier comparison of the textures and geometry

15.4.3.4 Quantitative Results

To quantitatively assess the performance of our avatar generation method, we conducted a user study with 22 volunteers, evaluating three key aspects: (1) consistency with input text, (2) texture quality, and (3) geometry quality. We randomly selected 8 input texts describing various appearances and used them to generate avatars through three methods: Dream Field, Text2Mesh, and our AvatarCLIP. For each generated sample, the volunteers rated the results on a scale from 1 to 5 for each of the three aspects. As shown in Fig. 15.15, our method consistently outperforms both baseline methods across all three criteria. Additionally, the standard deviations of our method’s scores are the lowest, further demonstrating its stable and reliable performance (Fig. 15.18).

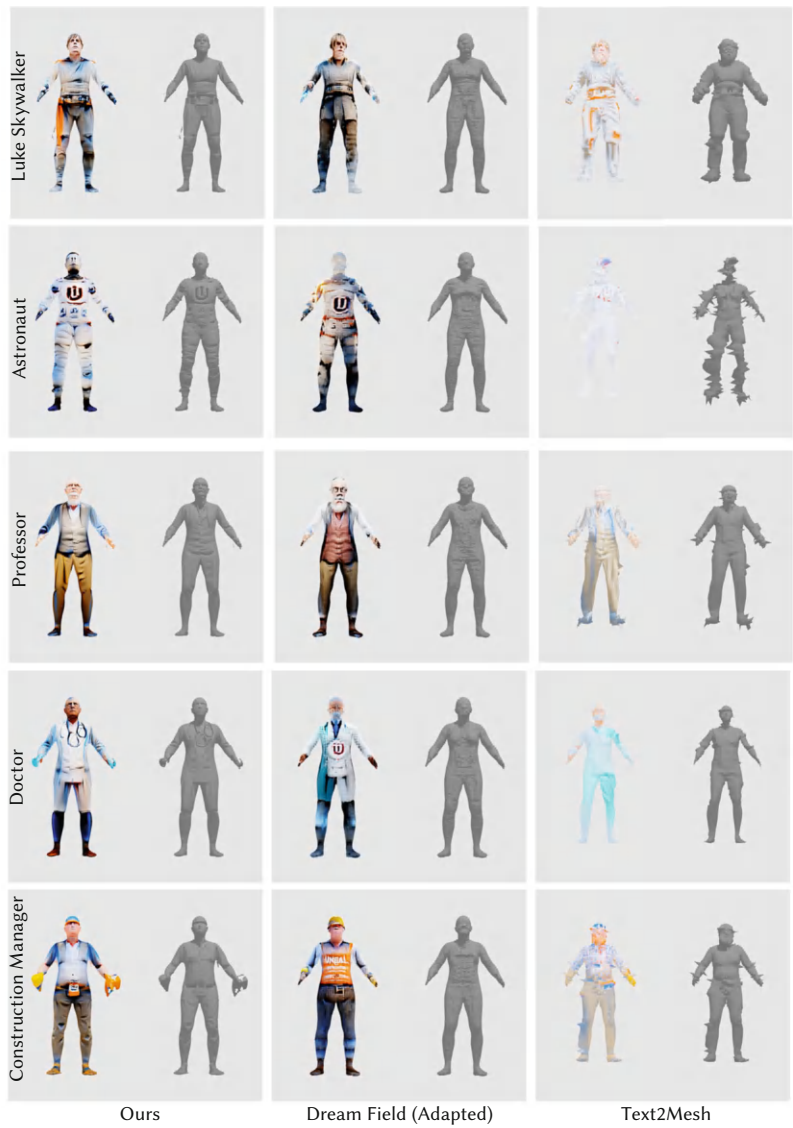


Fig. 15.17 Qualitative comparison with baseline methods. Side-by-side comparisons are presented between our method and two baseline approaches: Text2Mesh [75] (third row) and Dream Field [76] (second row). The results demonstrate that our method clearly shows superior quality in both geometry and texture

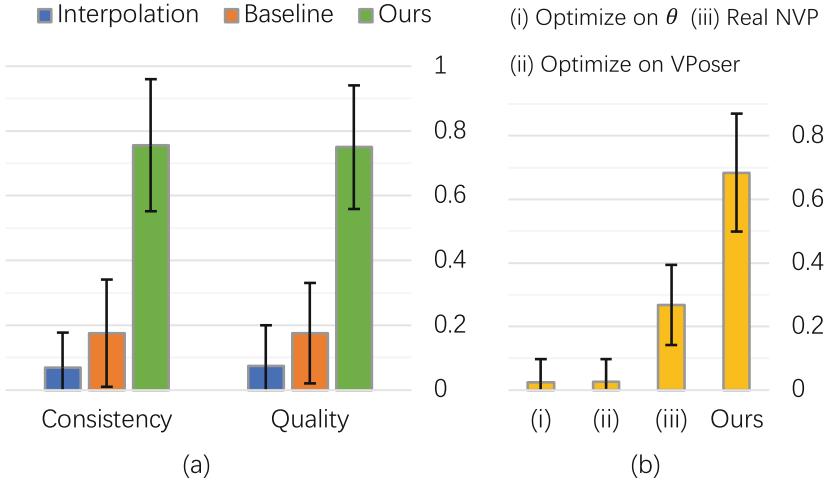


Fig. 15.18 Results of user study on motion generation. In candidate pose generation, our method outperforms both direct optimization and sampling from the multimodal Real NVP. When it comes to animation quality, our approach achieves significantly higher scores compared to both baseline methods. **(a)** References-based animation. **(b)** Candidate poses generation

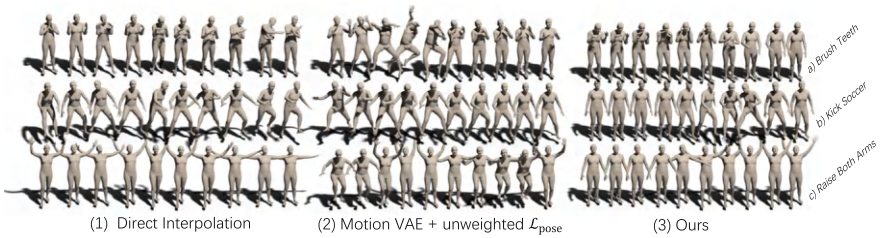


Fig. 15.19 Qualitative ablation of motion generation. In comparison to direct interpolation or using only the unweighted $\mathcal{L}_{\text{pose}}$, our proposed method generates stable and realistic motion sequences that remain consistent with the provided description

15.4.4 Experiments on Motion Generation

15.4.4.1 Qualitative Results of Motion Generation

Ablation Study To assess the effectiveness of the design choices in our reference-based animation module (i.e., the three discussed constraint terms and the use of the motion VAE as a motion prior), we compare our approach with two baseline methods, as illustrated in Fig. 15.19. For the task of “brush teeth,” method (i) generates disordered and unrelated pose sequences. Method (ii) also fails to produce reasonable motions, as it blindly focuses on reconstructing the unordered candidate poses. By incorporating a re-weighting mechanism, our method not only improves the reconstruction but also ensures the logical consistency of the generated motion.

For “kick soccer,” both (i) and (ii) fail to show significant changes in the leg motion during the kick. Here, the role of $\mathcal{L}_{\text{delta}}$ is crucial, as it helps control the intensity of the motion. In the case of “raise both arms,” the motion should transition from a neutral pose to one with raised arms. However, (i) produces a motion sequence that contradicts this expected progression, while (ii) introduces several unrelated actions. Using $\mathcal{L}_{\text{clip}}^m$, our method successfully generates motions with the correct sequence and greater consistency with the input descriptions.

15.4.4.2 Qualitative Results of Candidate Pose Generation

Comparison with Baseline Methods Figure 15.20a presents a comparison of various candidate pose generation methods. Both direct optimization approaches (i) and (ii) fail to produce reasonable poses, and they struggle even more to ensure consistency with the input description. These results indicate that directly optimizing human pose parameters is an impractical approach. In contrast, the conditioned Real NVP method (iii) produces more reasonable poses, though they still fall short of the quality and rationality achieved by our method, which relies on a codebook. While (iii) shows improvements over (i) and (ii), the poses it generates are generally of lower quality and less coherent than those generated by our solution.

Broad Range of Driving Texts To showcase the zero-shot capability of our pose generation method, we conduct experiments across four categories of motion

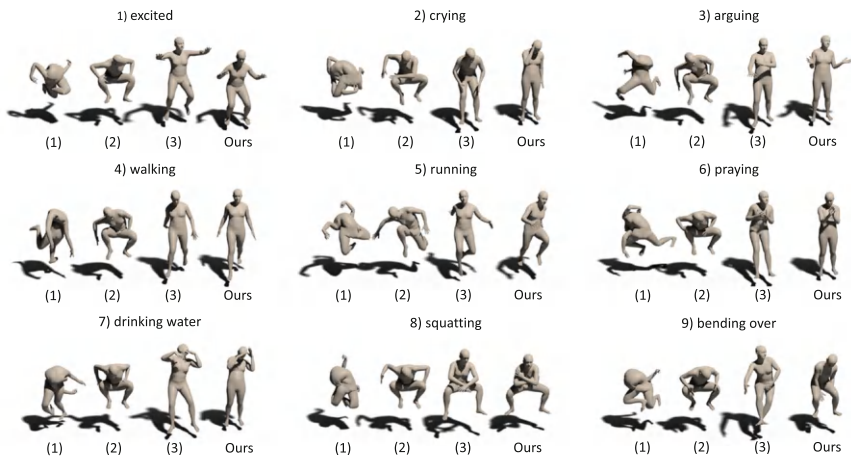


Fig. 15.20 Results of the candidate pose generation. (a) A comparison between our method and three baseline approaches is shown. Baseline (i) directly optimizes the SMPL parameter θ , while Baseline (ii) performs direct optimization over the latent space of the VPoser. Both methods struggle to generate plausible poses. Baseline (iii) uses a multi-modality Real NVP, which produces more reasonable poses but still falls short compared to our method by a significant margin. (b) Additional results of candidate pose generation are presented for further analysis

descriptions: (1) abstract emotional states (e.g., “tired” and “sad”), (2) common physical actions (e.g., “walking” and “squatting”), (3) body-part-specific motions (e.g., “raising both arms,” “washing hands”), and (4) interactions with objects (e.g., “shooting basketball”). The results of these experiments are presented in Fig. 15.20b.

15.4.4.3 Quantitative Results

We evaluate the performance of candidate pose generation and reference-based animation separately in a quantitative manner. For candidate pose generation, we enlist 58 volunteers to select the poses that best align with the given text descriptions from a set of 15 randomly chosen samples. The selection percentage for each method across all text inputs is then used as the score for evaluation. As shown in Fig. 15.18a, our method significantly outperforms the baseline approaches discussed in Sect. 15.4.4.2. For reference-based animation, we ask 20 volunteers to rate the results on a scale from 1 to 5, considering both consistency with the input text and overall quality, for 10 randomly selected samples. As seen in Fig. 15.18b, our method achieves substantial improvements over the two baseline methods introduced in Sect. 15.4.4.1, excelling in both consistency and quality.

15.5 Discussion

In this chapter, we introduce AvatarCLIP, a method that simplifies the otherwise complex process of creating 3D avatars, making it accessible to nonexpert users through a text-driven interface. This is achieved by leveraging the strong priors embedded in pre-trained models, such as the shape and motion VAEs, along with the large-scale vision-language model CLIP. We conduct extensive experiments to demonstrate the effectiveness of the thoughtful design choices incorporated into our approach.

Limitations and Potential Solutions For avatar generation, the limitations of weak supervision and CLIP’s low resolution result in imperfect outcomes, especially when zoomed in. This problem can be solved by switching the 2D foundation model to image diffusion models, e.g., Stable Diffusion [87]. In DreamFusion [88], authors propose Score Distillation Sampling (SDS) for text-to-3D generation with higher quality. Another potential solution is to use human image sets as the supervision signals. For example, EVA3D [89] uses internet human image collections for an unconditional high-resolution 3D human generation, with a technique named 3D-aware GAN [90].

Additionally, generating avatars with significant variation from the same prompt is challenging. Since the CLIP text feature remains consistent for the same prompt, the optimization directions are fixed, leading to similar results across different

runs. This is also a commonly observed issue in SDS, which can be improved by variational score distillation (VSD) [91]. In terms of motion synthesis, the constraints of the codebook design make it difficult to generate out-of-distribution poses, restricting the ability to create complex motions. Furthermore, the absence of a video version of CLIP hampers the generation of stylized motion sequences. Therefore, using paired motion-text datasets, e.g., KIT [92], to train a text-to-motion diffusion model [93–95] might be a better way in more practical motion synthesis.

Potential Negative Impacts The use of pre-trained models raises potential ethical concerns. For instance, when the input prompt t_{app} is set to “doctor,” the generated avatar is typically male, while setting t_{app} to “nurse” often results in a female avatar. This pattern highlights a gender bias that may be inherent in the large-scale internet data used to train CLIP, which could reflect societal biases if not properly curated. Addressing these ethical issues in large-scale pre-trained models is essential to ensure that zero-shot techniques can be applied safely. Additionally, the widespread accessibility of avatar and animation creation tools could lead to the production of fake videos featuring celebrities, which presents risks of misuse and negative social consequences.

Further Reading

Readers interested in 3D avatar creation or general 3D content creation may refer to the following materials for further reading.

- Hong et al. [89] for high-resolution photorealistic 3D human generation.
- Chen et al. [96], Hu et al. [97] for 3D human diffusion model.
- Mingyuan et al. [94] for text-to-motion diffusion model.
- Poole et al. [88] for general text-to-3D content creation from image diffusion priors.
- Hong et al. [98], Chen et al. [99] for native 3D diffusion model.

References

1. 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. Preprint arXiv:2103.00020 (2021)
2. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J., SMPL: a skinned multi-person linear model. *ACM Trans. Graph.* **34**(6), 1–16 (2015)
3. Sarkar, K., Liu, L., Golyanik, V., Theobalt, C.: HumanGAN: a generative model of humans images. Preprint. arXiv:2103.06902 (2021)
4. Lewis, K.M., Varadharajan, S., Kemelmacher-Shlizerman, I.: TryOnGAN: body-aware try-on via layered interpolation. *ACM Trans. Graph.* **40**(4), 1–10 (2021)

5. Grigorev, A., Isakov, K., Ianina, A., Bashirov, R., Zakharkin, I., Vakhitov, A., Lempitsky, V.: StylePeople: a generative model of fullbody human avatars. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5151–5160 (2021)
6. Zhang, J., Sanginetto, E., Tang, H., Siarohin, A., Zhong, Z., Sebe, N., Wang, W.: 3D-aware semantic-guided generative model for human synthesis. Preprint. arXiv:2112.01422 (2021)
7. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3D human motion synthesis with transformer VAE. Preprint. arXiv:2104.05670 (2021)
8. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: conditioned generation of 3D human motions. In: *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2021–2029 (2020)
9. Kania, K., Kowalski, M., Trzciński, T.: TrajeVAE—controllable human motion generation from trajectories. Preprint. arXiv:2104.00351 (2021)
10. Ling, H.Y., Zinno, F., Cheng, G., Van De Panne, M.: Character controllers using motion VAEs. *ACM Trans. Graph.* **39**(4), 40:1–40:12 (2020)
11. Ghosh, A., Cheema, N., Oguz, C., Theobalt, C., Slusallek, P.: Text-based motion synthesis with a hierarchical two-stream RNN. In: *ACM SIGGRAPH 2021 Posters*, pp. 1–2 (2021)
12. Tevet, G., Gordon, B., Hertz, A., Bermano, A.H., Cohen-Or, D.: Motionclip: exposing human motion generation to clip space. Preprint. arXiv:2203.08063 (2022)
13. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. Preprint. arXiv:2102.12092 (2021)
14. Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Treitsch, E., Wang, Y., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., et al.: Advances in neural rendering. Preprint. arXiv:2111.05849 (2021)
15. Pishchulin, L., Wuhler, S., Helten, T., Theobalt, C., Schiele, B.: Building statistical shape spaces for 3D human modeling. *Pattern Recogn.* **67**, 276–286 (2017)
16. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10975–10985 (2019)
17. Bhatnagar, B.L., Sminchisescu, C., Theobalt, C., Pons-Moll, G.: Combining implicit function learning and parametric models for 3D human reconstruction. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, August 23–28, 2020, Proceedings, Part II 16*, pp. 311–329. Springer, Berlin (2020)
18. Jiang, B., Zhang, J., Hong, Y., Luo, J., Liu, L., Bao, H., BCNet: learning body and cloth shape from a single image. In: *European Conference on Computer Vision*, pp. 18–35. Springer, Berlin (2020)
19. Bhatnagar, B.L., Tiwari, G., Theobalt, C., Pons-Moll, G.: Multi-garment net: learning to dress 3D people from images. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5420–5430 (2019)
20. Hong, F., Pan, L., Cai, Z., Liu, Z.: Garment4D: garment reconstruction from point cloud sequences. In: *Advances in Neural Information Processing Systems* (2021)
21. Corona, E., Pumarola, A., Aleny, G., Pons-Moll, G., Moreno-Noguer, F.: Smplicit: topology-aware generative model for clothed people. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11875–11885 (2021)
22. Palafox, P., Božić, A., Thies, J., Nießner, M., Dai, A.: NPMs: neural parametric models for 3D deformable shapes. Preprint. arXiv:2104.00702 (2021)
23. Saito, S., Yang, J., Ma, Q., Black, M.J.: SCANimate: weakly supervised learning of skinned clothed avatar networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2886–2897 (2021)
24. Burov, A., Nießner, M., Thies, J.: Dynamic surface function networks for clothed human bodies. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10754–10764 (2021)
25. Mihajlovic, M., Zhang, Y., Black, M.J., Tang, S.: LEAP: learning articulated occupancy of people. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10461–10471 (2021)

26. Habermann, M., Liu, L., Xu, W., Zollhoefer, M., Pons-Moll, G., Theobalt, C.: Real-time deep dynamic characters. *ACM Trans. Graph.* **40**(4), 1–16 (2021)
27. Huang, Z., Xu, Y., Lassner, C., Li, H., Tung, T.: Arch: animatable reconstruction of clothed humans. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3093–3102 (2020)
28. Weng, C.-Y., Curless, B., Kemelmacher-Shlizerman, I.: Photo wake-up: 3D character animation from a single photo. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5908–5917 (2019)
29. Alldieck, T., Magnor, M., Xu, W., Theobalt, C., Pons-Moll, G.: Video based reconstruction of 3D people models. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8387–8397 (2018)
30. 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: *European Conference on Computer Vision*, pp. 405–421. Springer, Berlin (2020)
31. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: semantically consistent few-shot view synthesis. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5885–5894 (2021)
32. Peng, S., Zhang, Y., Xu, Y., Wang, Q., Shuai, Q., Bao, H., Zhou, X.: Neural body: implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9054–9063 (2021)
33. Zhao, F., Yang, W., Zhang, J., Lin, P., Zhang, Y., Yu, J., Xu, L.: HumanNeRF: generalizable neural human radiance field from sparse inputs. Preprint. arXiv:2112.02789 (2021)
34. Liu, L., Habermann, M., Rudnev, V., Sarkar, K., Gu, J., Theobalt, C.: Neural actor: neural free-view synthesis of human actors with pose control. *ACM Trans. Graph.* **40**(6), 1–16 (2021)
35. Peng, S., Dong, J., Wang, Q., Zhang, S., Shuai, Q., Bao, H., Zhou, X.: Animatable neural radiance fields for human body modeling. arXiv e-prints, pp. arXiv–2105 (2021)
36. Xu, H., Alldieck, T., Sminchisescu, C.: H-NeRF: neural radiance fields for rendering and temporal reconstruction of humans in motion. *Adv. Neural Inf. Proces. Syst.* **34** (2021)
37. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: NeuS: learning neural implicit surfaces by volume rendering for multi-view reconstruction. Preprint. arXiv:2106.10689 (2021)
38. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410 (2019)
39. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119 (2020)
40. Jiang, Y., Huang, Z., Pan, X., Loy, C.C., Liu, Z.: Talk-to-edit: fine-grained facial editing via dialog. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13799–13808 (2021)
41. Fu, J., Li, S., Jiang, Y., Lin, K.-Y., Qian, C., Loy, C.-C., Wu, W., Liu, Z.: Stylegan-human: a data-centric odyssey of human generation. Preprint, vol. arXiv:2204.11823 (2022)
42. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: *Proceedings of International Conference on Computer Vision (ICCV)* (2015)
43. Han, X., Wu, Z., Wu, Z., Yu, R., Davis, L.S.: Viton: an image-based virtual try-on network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7543–7552 (2018)
44. Jiang, Y., Yang, S., Qiu, H., Wu, W., Loy, C.C., Liu, Z.: Text2human: text-driven controllable human image generation. *ACM Trans. Graph.* **41**(4), 1–11 (2022)
45. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: Animating arbitrary objects via deep motion transfer. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2377–2386 (2019)

46. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. *Adv. Neural Inf. Proces. Syst.* **32**, 7137–7147 (2019)
47. Yoon, J.S., Liu, L., Golyanik, V., Sarkar, K., Park, H.S., Theobalt, C.: Pose-guided human animation from a single image in the wild. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15039–15048 (2021)
48. Sarkar, K., Golyanik, V., Liu, L., Theobalt, C.: Style and pose control for image synthesis of humans from a single monocular view. Preprint. arXiv:2102.11263 (2021)
49. Chen, X., Jiang, T., Song, J., Yang, J., Black, M.J., Geiger, A., Hilliges, O.: gDNA: towards generative detailed neural avatars. arXiv (2022)
50. Noguchi, A., Sun, X., Lin, S., Harada, T.: Neural articulated radiance field. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5762–5772 (2021)
51. Noguchi, A., Sun, X., Lin, S., Harada, T.: Unsupervised learning of efficient geometry-aware neural articulated representations. Preprint. arXiv:2204.08839 (2022)
52. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-GAN: periodic implicit generative adversarial networks for 3D-aware image synthesis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5799–5809 (2021)
53. 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)
54. von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3D human pose in the wild using imus and a moving camera. In: *European Conference on Computer Vision (ECCV)* (2018)
55. Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3D human pose estimation in the wild using improved CNN supervision. In: *2017 International Conference on 3D Vision (3DV)*, pp. 506–516. IEEE, Piscataway (2017)
56. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: archive of motion capture as surface shapes. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5442–5451 (2019)
57. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: *CVPR* (2017)
58. Cai, Z., Zhang, M., Ren, J., Wei, C., Ren, D., Li, J., Lin, Z., Zhao, H., Yi, S., Yang, L., et al.: Playing for 3D human recovery. Preprint. arXiv:2110.07588 (2021)
59. Cai, Z., Ren, D., Zeng, A., Lin, Z., Yu, T., Wang, W., Fan, X., Gao, Y., Yu, Y., Pan, L., Hong, F., Zhang, M., Loy, C.C., Yang, L., Liu, Z.: HuMMan: multi-modal 4D human dataset for versatile sensing and modeling. Preprint. arXiv:2204.13686 (2022)
60. Rose, C., Cohen, M.F., Bodenheimer, B.: Verbs and adverbs: multidimensional motion interpolation. *IEEE Comput. Graph. Appl.* **18**, 32–40 (1998)
61. Ikemoto, L., Arikan, O., Forsyth, D.: Generalizing motion edits with gaussian processes. *ACM Trans. Graph.* **28**, 1–12 (2009)
62. Mukai, T., Kuriyama, S.: Geostatistical motion interpolation. *ACM Trans. Graph.* **24**, 1062–1070 (2005)
63. Xiao Lin, M.R.A.: Human motion modeling using DVGANs. Preprint. arXiv:1804.10652 (2014)
64. Ahn, H., Ha, T., Choi, Y., Yoo, H., Oh, S.: Text2action: generative adversarial synthesis from language to action. In: *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, May 21–25, 2018*, pp. 1–5. IEEE, Piscataway (2018)
65. Ahuja, C., Morency, L.-P.: Language2pose: natural language grounded pose forecasting. In: *2019 International Conference on 3D Vision (3DV)*, pp. 719–728 (2019)
66. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: AI choreographer: music conditioned 3D dance generation with aist++. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13401–13412 (2021)
67. Aggarwal, G., Parikh, D.: Dance2music: automatic dance-driven music generation. Preprint. arXiv:2107.06252 (2021)

68. Bergamin, K., Clavet, S., Holden, D., Forbes, J.R.: DReCon: data-driven responsive control of physics-based characters. *ACM Trans. Graph.* **38**(6), 1–11 (2019)
69. Won, J., Lee, J.: Learning body shape variation in physics-based characters. *ACM Trans. Graph.* **38**(6), 1–12 (2019)
70. Mansimov, E., Parisotto, E., Ba, J.L., Salakhutdinov, R.: Generating images from captions with attention. Preprint. arXiv:1511.02793 (2015)
71. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: *International Conference on Machine Learning*, pp. 1060–1069. PMLR (2016)
72. Frans, K., Soros, L., Witkowski, O.: Clipdraw: exploring text-to-drawing synthesis through language-image encoders. Preprint. arXiv:2106.14843 (2021)
73. Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., Bai, X.: A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. Preprint. arXiv:2112.14757 (2021)
74. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: text-driven manipulation of stylegan imagery. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2085–2094 (2021)
75. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: text-driven neural stylization for meshes. Preprint. arXiv:2112.03221 (2021)
76. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields. Preprint. arXiv:2112.01455 (2021)
77. Sanghi, A., Chu, H., Lambourne, J.G., Wang, Y., Cheng, C.-Y., Fumero, M.: Clip-forge: towards zero-shot text-to-shape generation. Preprint. arXiv:2110.02624 (2021)
78. Jetchev, N.: ClipMatrix: text-controlled creation of 3D textured meshes. Preprint. arXiv:2109.12922 (2021)
79. Wang, C., Chai, M., He, M., Chen, D., Liao, J.: Clip-nerf: text-and-image driven manipulation of neural radiance fields. Preprint. arXiv:2112.05139 (2021)
80. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. *ACM Siggraph Comput. Graph.* **21**(4), 163–169 (1987)
81. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. Preprint. arXiv:2002.10099 (2020)
82. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5745–5753 (2019)
83. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, pp. 5998–6008 (2017)
84. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. Preprint. arXiv:1312.6114 (2013)
85. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. Preprint. arXiv:1412.6980 (2014)
86. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real NVP. Preprint. arXiv:1605.08803 (2016)
87. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2021)
88. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: text-to-3d using 2D diffusion. arXiv (2022)
89. Hong, F., Chen, Z., Lan, Y., Pan, L., Liu, Z.: Eva3D: compositional 3D human generation from 2D image collections. Preprint. arXiv:2210.04888 (2022)
90. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3D generative adversarial networks. arXiv (2021)
91. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: ProlificDreamer: high-fidelity and diverse text-to-3D generation with variational score distillation. In: *Thirty-Seventh Conference on Neural Information Processing Systems* (2023)

92. Plappert, M., Mandery, C., Asfour, T.: The kit motion-language dataset. *Big Data* **4**(4), 236–252 (2016)
93. Mingyuan, Z., Daisheng, J., Chenyang, G., Fangzhou, H., Zhongang, C., Jingfang, H., Chongzhi, Z., Xinying, G., Yang Lei, L.Z., Ying, H.: Large motion model for unified multi-modal motion generation. Preprint. arXiv:2404.01284 (2024)
94. Mingyuan, Z., Zhongang, C., Liang, P., Fangzhou, H., Xinying, G., Lei, Y., Ziwei, L.: Motiondiffuse: text-driven human motion generation with diffusion model. Preprint. arXiv:2208.15001 (2022)
95. Mingyuan, Z., Xinying, G., Liang, P., Zhongang, C., Fangzhou, H., Huirong, L., Lei, Y., Ziwei, L.: Remodiffuse: retrieval-augmented motion diffusion model. Preprint. arXiv:2304.01116 (2023)
96. Chen, Z., Hong, F., Mei, H., Wang, G., Yang, L., Liu, Z.: Primdiffusion: volumetric primitives diffusion for 3D human generation. In: *Thirty-Seventh Conference on Neural Information Processing Systems* (2023)
97. Hu, T., Hong, F., Liu, Z.: StructLDM: structured latent diffusion for 3D human generation (2024)
98. Hong, F., Tang, J., Cao, Z., Shi, M., Wu, T., Chen, Z., Wang, T., Pan, L., Lin, D., Liu, Z.: 3DTopia: large text-to-3D generation model with hybrid diffusion priors. Preprint. arXiv:2403.02234 (2024)
99. Chen, Z., Tang, J., Dong, Y., Cao, Z., Hong, F., Lan, Y., Wang, T., Xie, H., Wu, T., Saito, S., Pan, L., Lin, D., Liu, Z.: 3DTopia-xl: high-quality 3D PBR asset generation via primitive diffusion. Preprint. arXiv:2409.12957 (2024)

Chapter 16

Text-Driven 3D Human Motion Generation



Mingyuan Zhang, Zhongang Cai, Liang Pan, Chenyang Gu, Fangzhou Hong, Xinying Guo, Jiawei Ren, Lei Yang, and Ziwei Liu

Abstract 3D human motion generation plays a crucial role in many applications. However, in previous studies, motion generation algorithms often focused on a limited number of motion categories or required complex methods to control the desired outputs. The goal of this chapter is to address the question of how to use natural language as the sole control signal to generate diverse and realistic motion sequences while enabling users to perform fine-grained control. This form of control greatly lowers the threshold for using this technology, allowing even layman users to benefit from the technology. Specifically, this chapter first uses MotionDiffuse as an example to introduce a classic text-driven motion generation pipeline design and studies how to fully integrate the text modality with the motion modality. Then, the chapter discusses how to leverage the concept of retrieval-augmented generation to further enhance the model's generation performance. Finally, this chapter introduces two effective solutions to solve the problem of fine-grained motion generation and demonstrates how such generation capability can benefit real-world applications.

Keywords Human motion · 3D · Text-driven motion generation · Vision-language model · CLIP

M. Zhang · L. Pan · C. Gu · F. Hong · X. Guo · J. Ren · Z. Liu (✉)
S-Lab, Nanyang Technological University, Singapore, Singapore
e-mail: mingyuan001@e.ntu.edu.sg; pan.liang@u.nus.edu; cgu002@e.ntu.edu.sg;
fangzhou001@e.ntu.edu.sg; xinying.guo@ntu.edu.sg; jiawei011@e.ntu.edu.sg;
ziwei.liu@ntu.edu.sg

Z. Cai
S-Lab, Nanyang Technological University and SenseTime Research, Singapore, Singapore
e-mail: caizhongang@sensetime.com

L. Yang
SenseTime Research, Hong Kong, China
e-mail: yanglei@sensetime.com

16.1 Introduction

3D graphics play an increasingly important role in our daily lives. In the realm of entertainment, 3D movies, 3D games, and the recently popular 3D virtual streamers have become favored pastimes among modern youth. In industrial scenarios, applications like industrial robots and virtual driving also rely on realistic 3D simulations to provide suitable training environments. Among these applications, 3D human data is a crucial component. Taking 3D games as an example, whether it's the player controlling their character or the movements of NPCs in the game world, high-quality 3D motion ensures that these characters move like real people, creating a more immersive experience for players. However, over the past few decades, people have still relied on complex motion capture systems or professional software like Blender and Maya for manual editing. This pipeline faces significant cost and efficiency challenges on the one hand, and on the other, it limits more users from generating their desired 3D motion data and incorporating it into their own animations or games. Therefore, it is essential to develop a motion generation algorithm that is easy to interact with, capable of producing high-quality results, and allows for fine-grained editing. Such a solution would lower the barrier to entry and encourage more people to participate in 3D creation.

In previous research, researchers have explored predefined motion categories, enabling motion generation algorithms to produce diverse motions within these categories [7, 10, 28]. Others have attempted to use music as input, allowing models to generate dance motions synchronized with the rhythm [14, 19, 20, 38, 49]. Additionally, there have been studies focusing on using natural language as input [1, 9, 22, 29]. Among these approaches, natural language is undoubtedly an excellent control signal, enabling users to easily describe the motion data they desire. Therefore, in this chapter, we focus on text-driven motion generation, which significantly removes the prerequisite for layman users.

Research on motion generation using text as input has only started gaining attention in the past years, and there are still many limitations in early studies. For example, TEMOS [29] cannot generate stylized motions and fails to produce diverse motion sequences for the same description. MotionCLIP [40], leveraging a pre-trained model, can interpret open-domain natural language and generate reasonably matching motion sequences for stylized descriptions. However, since its training relies on aligning features between short text inputs and motion sequences, it struggles to generate more complex motions. Additionally, both methods can only accept a short prompt and lack fine-grained editing capabilities, which limits users' creative flexibility and the quality of the generated outputs.

In this chapter, we first introduce how to leverage the powerful generative capabilities of the diffusion model, along with targeted model architecture design, to tackle with the abovementioned issues and elevate the quality of motion generation to a new level. Specifically, we use MotionDiffuse [46] as an example to explain how to overcome the challenges encountered when transferring diffusion model

techniques to motion generation and how to design an efficient motion diffusion model.

In addition, we introduce the retrieval-augmented motion generation technology. In the traditional 3D graphics field, retrieval-based techniques using motion libraries are a common application. In this chapter, we use ReMoDiffuse [45] to discuss how to combine retrieval techniques with generative models, proposing a new motion generation framework. Specifically, we explain how to perform retrieval based on both semantic and kinematic features, how to integrate the retrieved motion features into the motion generation process, and how to leverage the diffusion model’s classifier-free guidance to further enhance the retrieval-augmented generation effect.

Finally, we introduce fine-grained text-driven motion generation techniques. This technology allows users to define body part divisions and time segments. Users can provide different text descriptions for different body parts and time segments. To address this, we specifically introduce two implementation methods. One is the strategy used in MotionDiffuse during the inference stage, where it generates complete motion sequences for each text prompt in parallel. The different motion sequences naturally blend together during a series of denoising processes. The other is the FineMoGen [47] approach, which uses spatial-temporal independent modeling techniques. By incorporating body part modeling in the FFN structure and kinetic modeling in the attention mechanism, it supports precise control over different body parts and time segments.

16.2 Text-Driven Motion Diffusion Model

In this section, we use MotionDiffuse [46] as an example to introduce the task of text-driven motion generation and the commonly used motion diffusion model structures. In Sect. 16.2.1, we provide a formal definition of the text-driven motion generation task and discuss its relationship with action-conditioned motion generation. In Sect. 16.2.2, we explain the rationale for introducing diffusion models, as well as the details of the diffusion process and reverse process. Finally, in Sect. 16.2.3, we address the challenges encountered in network architecture design and present corresponding solutions.

16.2.1 Problem Definition

The motion sequence is represented as $\Theta \in \mathbb{R}^{F \times D}$, where F denotes the number of frames and D represents the dimensionality of each pose state. The specific representation of each pose state varies across different datasets, as shown in Table 16.1. The Y-axis is oriented perpendicular to the ground. In the HumanML3D and KIT-ML datasets, the positive direction of the Y-axis points upward, perpendicular to the ground. The Z-axis points forward, aligned with the initial front-facing direction of

Table 16.1 Datasets for text-driven motion generation

Dataset	Skeleton	Representation	#Dim
HumanML3D [11]	SMPL w/o hands, 22 joints	$(r^{va}, r^{vx}, r^{vz}, r^h, \mathbf{j}^p, \mathbf{j}^v, \mathbf{j}', \mathbf{f})$	263
KIT-ML [30]	MMM, 21 joints	$(r^{va}, r^{vx}, r^{vz}, r^h, \mathbf{j}^p, \mathbf{j}^v, \mathbf{j}', \mathbf{f})$	251
UESTC [15]	SMPL, 24 joints	$(\mathbf{t}, \mathbf{j}')$	147
HumanAct12 [10]	SMPL, 24 joints	$(\mathbf{j}^p)$	144
BABEL [31]	SMPL, 24 joints	$(\mathbf{t}, \mathbf{j}')$	147
HuMMan-MoGen [47]	SMPL, 24 joints	$(\mathbf{t}, \mathbf{j}')$	147

the human figure, and the X-axis points from the left shoulder to the right shoulder. Other datasets follow the same coordinate system as SMPL. Below, we provide detailed explanations for each symbol:

- $r^{va}, r^{vx}, r^{vz} \in \mathbb{R}$: Represent the angular velocity of the root joint along the Y-axis and the linear velocities along the X-axis and Z-axis, respectively.
- $r^h \in \mathbb{R}$: Denotes the height of the root joint.
- $\mathbf{j}^p, \mathbf{j}^v \in \mathbb{R}^{J \times 3}$: Represent the local position and linear velocity of each joint, where J is the total number of joints. The definition of $\mathbf{j}^p$ varies slightly across different datasets:
 - In HumanML3D and KIT-ML, all keypoints are first aligned to the positive Z-axis (the forward-facing direction). The local position $\mathbf{j}^p$ is then computed as the coordinate difference relative to the root joint after this alignment.
 - In HumanAct12, the local position $\mathbf{j}^p$ is directly calculated as the coordinate difference relative to the root joint, without any prior alignment.
- $\mathbf{j}' \in \mathbb{R}^{J \times 6}$: Represents the 6D rotation [48] of each joint.
- $\mathbf{f} \in \{0, 1\}^4$: Indicates whether the heel or toe of each foot (left or right) is in contact with the ground, with 1 signifying contact and 0 indicating no contact.
- $\mathbf{t} \in \mathbb{R}^3$: Represents the trajectory of the root joint.

In standard text-driven motion generation, the training data consists of paired text and motion samples. During inference, the goal is to generate a motion sequence that corresponds to each provided textual description. This task is often referred to as text-to-motion(T2M) translation, a term we use throughout this chapter. A closely related challenge is action-conditioned motion(A2M) generation, where the objective is to synthesize motion sequences based on a predefined set of action categories. Here, the training data comprises action labels paired with motion sequences. MotionDiffuse enhances this task by replacing category IDs with their semantic descriptions, allowing us to unify the action-conditioned framework with the T2M pipeline.

16.2.2 Motion Diffusion Model

Comparison of Motion Generative Model In the literature, generative adversarial networks (GANs), implicit functions, auto-encoders (AEs) and variational auto-encoders (VAEs) are widely adopted in the field of motion generation [7, 9, 29, 42]. However, these generative models have severe issues that limit the further development. GANs apply an adversarial training scheme and are typically difficult to train [16, 35]. Implicit functions are straightforward and effective for small datasets but often fail to generalize under complex conditions. T2M task requires the methods to understand complicated text condition, which makes it more difficult to apply this type of generative models. AEs and VAEs encode text and motion into a shared embedding space to enable deterministic text-to-motion mapping. However, high-level text features typically lack the granularity to guide detailed and diverse motion generation, which restricts the diversity of outputs. To overcome these challenges, MotionDiffuse builds a new text-driven motion generation pipeline on diffusion models, a novel class of generative models that have demonstrated success in various domains, such as image and 3D point cloud generation [8, 12, 25, 26].

Diffusion Model Diffusion models gradually denoise Gaussian noise into the target data, following a probabilistic framework:

$$p_{\theta}(\mathbf{x}_0) = \int p_{\theta}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}, \quad (16.1)$$

where $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ represents real data and $\mathbf{x}_1, \dots, \mathbf{x}_T$ are the generated noised data. The diffusion model consists of a diffusion process and a reverse process, as shown in Fig. 16.1.

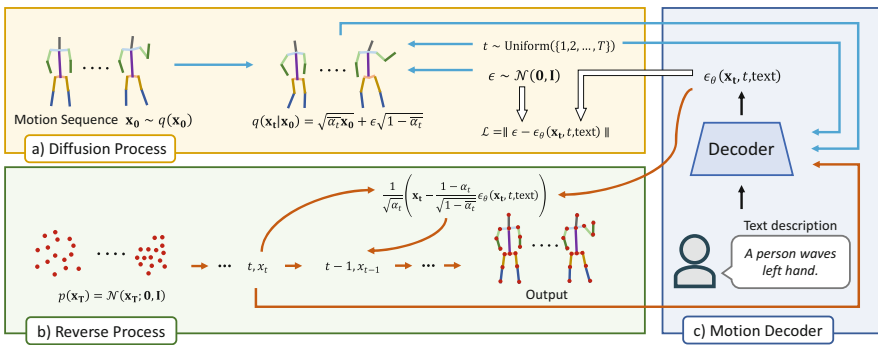


Fig. 16.1 Overall pipeline of motion diffusion model. The colors of the arrows indicate different stages: blue for training, red for inference, and black for both training and inference

During diffusion process, noise is gradually added to the data via a Markov chain:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad (16.2)$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}).$$

where β_t defines the variance schedule.

In the reverse process, the model learns to denoise iteratively, starting from $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad (16.3)$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)).$$

During training, the forward process can be reformulated as

$$q(\mathbf{x}_t | \mathbf{x}_0) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (16.4)$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. This allows direct sampling of $\mathbf{x}_t$ by introducing noise $\boldsymbol{\epsilon}$.

Training and Inference Instead of predicting $\mathbf{x}_{t-1}$, MotionDiffuse follows GLIDE [26] to predict $\boldsymbol{\epsilon}$ by minimizing the following loss:

$$\mathcal{L} = \mathbb{E}_{t \sim [1, T], \mathbf{x}_0 \sim q(\mathbf{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \text{text})\|^2 \right], \quad (16.5)$$

where $\boldsymbol{\epsilon}_\theta$ denotes the motion decoder, which can estimate the added noise $\boldsymbol{\epsilon}$ given the noised motion sequences $\mathbf{x}_t$, time step t , and text conditions.

To generate motion sequences, the reverse process starts from $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The mean $\mu_\theta(\mathbf{x}_t, t, \text{text})$ is estimated as

$$\mu_\theta(\mathbf{x}_t, t, \text{text}) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \text{text}) \right). \quad (16.6)$$

For simplicity, $\Sigma_\theta(\mathbf{x}_t, t, \text{text})$ is set to β_t , following [8]. By iteratively applying this process, a clean motion sequence conditioned on the given text can be generated.

16.2.3 Motion Decoder

As illustrated in Sect. 16.2.2, the core of motion diffusion model is to establish an effective and efficient motion decoder $\epsilon_\theta(\mathbf{x}_t, t, \text{text})$, which is used for denoising steps. Figure 16.2 shows the architecture design of MotionDiffuse. Previous works [8, 12, 25, 26] mainly utilize UNet-like structure as the denoising model. The main reason for using the transformer here is to address the challenge of generating motions with variable lengths. The model is primarily composed of the embedding layer, text encoder, and motion decoder. The embedding layer applies a simple linear projection to each frame's pose data and uses learnable position embeddings to introduce the temporal order of the sequence. Inspired by other text-driven generation tasks, in the text encoder section, MotionDiffuse aims to leverage pre-trained generation models on large datasets to extract features from raw input text. The benefit of this approach is that we can obtain more representative features and achieve better generalization during the testing phase. Following AvatarCLIP [13] and MotionCLIP [40], CLIP [33]'s pre-trained weights are used for initializing the text encoder. This part is fixed during training, followed by several learnable transformer encoder layers to better adapt the text features to our T2M task. Below, we mainly introduce the design of motion decoder.

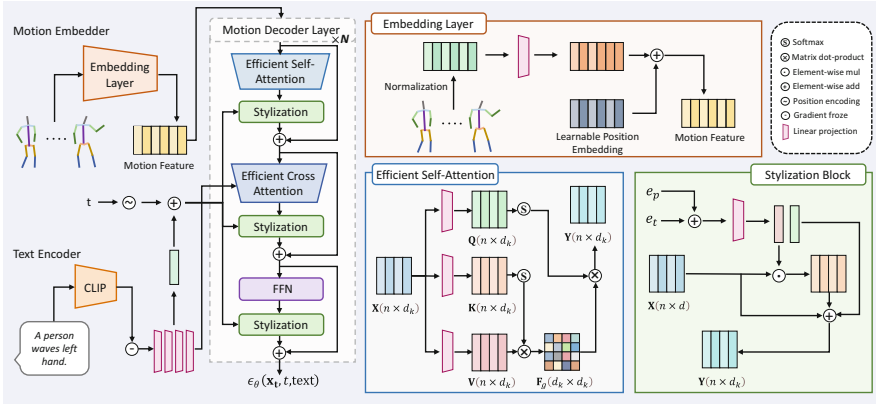


Fig. 16.2 Architecture of MotionDiffuse. The left side shows the overall structure of the MotionDiffuse model, while the right side displays the detailed design of three modules: the embedding layer, efficient self-Attention, and stylization block. The entire model is based on the transformer architecture, with additional designs tailored for the diffusion model and text-driven motion generation tasks. Key differences include the use of efficient self-attention, efficient cross-attention, and the stylization block. Compared to traditional self-attention and cross-attention mechanisms, the two attention mechanisms enhance computational efficiency and are more effective at capturing global information, enabling the model to better plan the pose for each frame. The stylization block injects text features and timestep information into the motion features, allowing the model to leverage the current denoising step and generate motion sequences that are more consistent with the textual description

Efficient Self-Attention and Cross-Attention The main benefit of self-attention is its ability to provide a global perspective of the input sequence, which is crucial for estimating the injected noise ϵ . However, the time complexity of calculating attention weights $\mathbf{A} \in \mathbb{R}^{n \times n}$ is $O(n^2d)$, which leads to quadratic time growth as the sequence length increases. In the T2M task, where the sequence length can be several hundred frames, this results in significant slowdowns. Therefore, efficient attention [37] is adopted to accelerate the whole module. Rather than computing pairwise attention weights, efficient attention generates a global feature map $\mathbf{F}_g \in \mathbb{R}^{d_k \times d_k}$, where d_k is the feature dimension after splitting the multi-heads:

$$\mathbf{F}_g = \text{softmax}(\mathbf{K}^\top) \otimes \mathbf{V}, \mathbf{Y} = \text{softmax}(\mathbf{Q}) \otimes \mathbf{F}_g. \quad (16.7)$$

The time complexity of these two steps is $O(d_k^2nk) = O(dd_kn)$, where n represents the number of sequence elements and k is the number of attention heads. Efficient cross-attention has similar formulation with efficient self-attention, which computes $\mathbf{K}$ and $\mathbf{V}$ from the text feature.

The traditional self-attention mechanism focuses on modeling the relationships between frames to ensure greater consistency in the generated results. In our motion diffusion model, this can be leveraged to better assist the model in estimating the injected noise. When using efficient attention, the introduction of the global template concept allows the model to more easily capture the overall motion content from the noised sequence. This also facilitates a holistic integration of the text condition, aiding in the optimization of individual frame features. As a result, efficient attention not only enhances efficiency but also significantly improves model accuracy in our T2M task.

Stylization Block The structure of stylization block is depicted in Fig. 16.2. Following a design similar to GLIDE [26], a text embedding $\mathbf{e}_p$ is obtained by linearly transforming the text features and a timestamp embedding $\mathbf{e}_t$ using positional embedding [43]. These two vectors are then summed to form a single embedding $\mathbf{e}$. Given the original output $\mathbf{Y}$ from other blocks, the stylization block processes the features as follows:

$$\mathbf{B} = \psi_b(\phi(\mathbf{e})), \quad \mathbf{W} = \psi_w(\phi(\mathbf{e})), \quad \mathbf{Y}' = \mathbf{Y} \cdot \mathbf{W} + \mathbf{B}, \quad (16.8)$$

where $(\cdot)$ denotes the Hadamard product and $\mathbf{Y}'$ is the output of the stylization block. ψ_b , ψ_w , and ϕ are different linear projections. In this way, both the diffusion timestep information and text embeddings can be injected into the motion features. The introduction of the timestep helps the model better understand which step of the reverse process it is currently in, reducing the difficulty of denoising. Meanwhile, incorporating the text embedding ensures that the generated motion sequence aligns more closely with the given text description.

16.3 Retrieval-Augmented Motion Generation

In this chapter, we use ReMoDiffuse [45] as an example to introduce how retrieval-augmentation techniques can enhance motion generation results. In Sect. 16.3.1, we discuss the motivation for using retrieval-augmentation techniques and the three key challenges faced. In Sects. 16.3.2, 16.3.3, and 16.3.4, we present corresponding solutions to these three challenges.

16.3.1 Motivation and Challenges

In traditional motion generation tasks, motion retrieval is a widely used technique, particularly in the gaming industry. For example, in sports games, after creating a high-quality motion library, required motion clips are retrieved during gameplay, and blending algorithms are applied to seamlessly connect these individual segments, resulting in a smooth and coherent motion. A natural idea is to introduce retrieval samples into the design of motion generative models and create corresponding algorithms that allow the model to gain additional guidance from the retrieved samples. While this idea is effective, it faces three significant challenges: 1) how to retrieve the most relevant motion sequence (this involves not only focusing on textual similarity but also assessing the low-level differences between the retrieved motions and the desired outcomes from certain perspectives), 2) how to utilize the retrieved samples to refine the motion feature sequence (this requires addressing the discrepancies between the retrieved motions and the desired outcomes, selectively incorporating information beneficial for the generation task), and 3) how to design an efficient inference strategy during the testing phase (this includes leveraging classifier-free guidance to maximize the influence of retrieved samples on the generated results).

To tackle with the abovementioned challenges, ReMoDiffuse incorporates three key techniques. In the retrieval stage, a **Hybrid Retrieval** technique that assess both semantic and kinematic similarities simultaneously to find out the most relevant samples. In the refinement stage, **semantics-modulated attention** is used to acquire additional guidance from the retrieved samples and generate semantic-consistent motion sequences. During inference, **condition mixture** technique allows diffusion model to generate high-fidelity and description-consistent motion sequences. The overall architecture design of ReMoDiffuse is shown in Fig. 16.3.

16.3.2 Hybrid Retrieval

To retrieve motion-text pairs that are similar to the given target, the primary consideration is semantic similarity, which is calculated by the pre-trained CLIP. Formally,

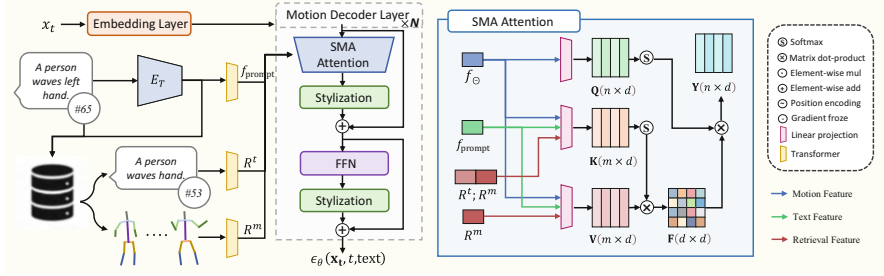


Fig. 16.3 Architecture of ReMoDiffuse. Given a text description and the expected motion length, the retrieval algorithm searches for the most relevant motion-text pairs in the database based on both aspects. For each pair, ReMoDiffuse uses two separate encoders to extract the text features R^t and motion features R^m . These features are then passed into the attention layer of the motion decoder. One distinction in the transformer architecture of ReMoDiffuse compared to MotionDiffuse is that while MotionDiffuse has two independent attention layers for computing self-attention and cross-attention, in ReMoDiffuse, the SMA module handles both tasks and additionally calculates the impact of retrieved samples on the motion sequence features

for each data point $(\text{text}_i, \Theta_i)$, ReMoDiffuse first computes $f_i^t = E_T(\text{text}_i)$ as the text-query feature, where E_T is the text encoder in the CLIP model. Thanks to the powerful generalization ability of CLIP, we can effectively compute semantic similarity for a wide range of general scenarios.

Another important aspect is kinematic similarity. However, since we cannot get the target motion sequence during testing, it is hard to acquire informative kinematic features. The only information we can leverage is the length of the motion, an unexpectedly effective kinematic feature. Specifically, we want to understand the relative scale between the expected motion length and the lengths of each entity in the database.

In summary, the similarity score s_i between the i -th data point, the given description prompt, and the expected motion length L is defined as follows:

$$s_i = \langle f_i^t, f_p^t \rangle \cdot e^{-\lambda \cdot \gamma}, \quad f_p^t = E_t(\text{prompt}), \quad \gamma = \frac{\|l_i - L\|}{\max l_i, L}, \quad (16.9)$$

where $\langle \cdot, \cdot \rangle$ represents the cosine similarity between two text features, indicating the semantic similarity. l_i is the length of the motion sequence Θ_i , which is used for assessing kinematic similarity. The similarity score s_i increases when the text-query closely matches the prompt feature and also when the expected motion length is close to the length of the entity. The parameter λ is a hyperparameter used to balance the contributions of these two similarities.

16.3.3 Semantics-Modulated Attention

In addition to text encoder, ReMoDiffuse also introduces a series encoders for the retrieved samples as shown in Fig. 16.3. R^t , R^m represent the extracted text features and motion features from the retrieved samples. As for the attention part, ReMoDiffuse proposed a new technique, called semantics-modulated attention(SMA). The noised motion sequence serves as the query vector $Q \in \mathbb{R}^{F \times D}$. For the key vector K and the value vector V , SMA draws from three data sources: (1) the motion feature sequence $f_\Theta \in \mathbb{R}^{F \times D}$; (2) the text condition f_{prompt} ; and (3) features R^m , R^t . For the value vector V , SMA concatenates f_Θ , f_{prompt} , R^m , and for the key vector K , ReMoDiffuse concatenates f_Θ , f_{prompt} , $[R^m; R^t]$, where $[\cdot; \cdot]$ denotes the concatenation of the two terms. This design allows SMA to incorporate low-level motion information from the retrieved samples while fully considering semantic similarities.

16.3.4 Condition Mixture

Classifier-free guidance is a significant technique in conditional diffusion model. During training, the text condition is dropped with probability 10%. During inference, the output of each denoising step is described as below:

$$\epsilon = w \cdot \epsilon_\theta(\mathbf{x}_t, t, \text{text}) - (w - 1) \cdot \epsilon_\theta(\mathbf{x}_t, t), \quad (16.10)$$

where w indicates the intensity of the guidance from given condition signals. In ReMoDiffuse, the motion decoder estimates the $\mathbf{x}_0$ instead of ϵ . The retrieved samples are regarded as an additional condition. Therefore, there are four types of estimations based on the combination of conditions: $S_\theta(\mathbf{x}_t, t, \text{retr}, \text{text})$, $S_\theta(\mathbf{x}_t, t, \text{retr})$, $S_\theta(\mathbf{x}_t, t, \text{text})$, $S_\theta(\mathbf{x}_t, t)$, where $S_\theta(\cdot)$ denotes the predicted $\mathbf{x}_0$ from the motion decoder. To combine all these four terms, four hyperparameters are required. In ReMoDiffuse, condition mixture is applied to generate a better combination of w_1, w_2, w_3, w_4 and get the final output $\hat{S}$ as

$$\begin{aligned} \hat{S}_\theta(\mathbf{x}_t, t, \text{retr}, \text{text}) = & w_1 \cdot S_\theta(\mathbf{x}_t, t, \text{retr}, \text{text}) + w_2 \cdot S_\theta(\mathbf{x}_t, t, \text{text}) + \\ & w_3 \cdot S_\theta(\mathbf{x}_t, t, \text{retr}) + w_4 \cdot S_\theta(\mathbf{x}_t, t). \end{aligned} \quad (16.11)$$

Empirically, the tendency of Frechet inception distance (FID) is highly similar to that of precision when the hyperparameters are nearly optimal. Therefore, minimizing FID can yield an optimal combination of w_1, w_2, w_3, w_4 .

16.4 Fine-Grained Text-Driven Generation

In this section, we introduce methods for fine-grained text-driven motion generation. First, in Sect. 16.4.1, we provide a rigorous definition of this task. Then, we present two specific implementation approaches. Section 16.4.2 introduces the fine-grained generation based on a customized inference strategy proposed by MotionDiffuse. Section 16.4.3 introduces the spatial-temporal mixture attention used in FineMoGen [47], which provides the model with spatial-temporal independence, enabling fine-grained generation.

16.4.1 Problem Definition

In contrast to standard text-driven motion generation tasks, fine-grained spatiotemporal motion generation requires the model to produce realistic action sequences that align with detailed text descriptions. The key distinction is that, in this case, multiple granular descriptions are provided, each associated with specific time ranges and body part scopes. Formally, in the fine-grained spatiotemporal motion generation task, the annotation format is defined as a description matrix $\text{Text}_{i,j}$, where $i \in [1, N_T]$ represents the stage index and $j \in [1, N_S]$ represents the body part index, with N_T being the number of stages and N_S being the number of body parts. Consequently, the generated motion sequences must satisfy all constraints: the entire action sequence Θ must be divided into N_T stages, with each stage containing N_S body parts. Each motion segment $\Theta_{i,j}$, corresponding to the j th body part in the i th stage, must align with its corresponding text description $\text{Text}_{i,j}$. Furthermore, we aim for the generated motions to appear natural and cohesive, ensuring smooth coordination between body parts and seamless transitions across time segments.

16.4.2 Test-Time Fine-Grained Generation

In addition to the standard diffusion inference strategy, MotionDiffuse explores how to achieve fine-grained control during the testing phase. Compared to VAEs, the multistep iterative process of diffusion models offers favorable properties. Specifically, we can generate different sequences for different prompts and then merge these sequences during the iterative process to produce a unified and natural motion sequence. This method has the advantage of imposing no requirements on the model’s training process or structural design, making it applicable to almost any motion diffusion model. Below, we detail the implementation of body part-independent control and time-varied control in MotionDiffuse.

Body Part-Isndependent Control In the denoising process, MotionDiffuse predicts the noise term $\epsilon_{\theta}(\mathbf{x}_t, t, \text{text}) \in \mathbb{R}^{F \times D}$, where F denotes the number of frames

and D represents the dimensionality of each pose. This noise term governs the denoising direction for the entire body.

Inspired by latent code interpolation, MotionDiffuse introduces a “noise interpolation” method to independently control different body parts. During each denoising step, it calculates the noise term $\epsilon_j^{\text{part}} = \epsilon_\theta(\mathbf{x}_t, t, \text{Text}_{*,j})$, $j \in [1, N_S]$ for each body part. These terms are combined through

$$\bar{\epsilon}^{\text{part}} = \left(\sum_{j=1}^{N_S} \epsilon_j^{\text{part}} \cdot M_j \right) + \lambda_1 \cdot \nabla \left(\sum_{1 \leq j_1, j_2 \leq N_S} \|\epsilon_{j_1}^{\text{part}} - \epsilon_{j_2}^{\text{part}}\| \right), \quad (16.12)$$

where $M_j \in 0, 1^D$ is a binary vector that specifies the body part of interest. The symbol $(\cdot)$ denotes the Hadamard product, with broadcasting omitted for simplicity in computation. ∇ denotes the gradient operator, and λ_1 serves as a hyperparameter to balance the two components. This correction aims to ensure consistency in overlapping regions.

Time-Variied Control Similar to body part-independent control, MotionDiffuse first calculates $\epsilon_i^{\text{time}} = \epsilon_\theta(\mathbf{x}_t, t, \text{Text}_{i,*})$ for each time segment and then interpolate these noise terms with a correction:

$$\bar{\epsilon}^{\text{time}} = \sum_{i=1}^{N_T} \bar{\epsilon}_i^{\text{time}} + \lambda_2 \cdot \nabla \left(\sum_{1 \leq i_1, i_2 \leq N_T} \|\bar{\epsilon}_{i_1}^{\text{time}} - \bar{\epsilon}_{i_2}^{\text{time}}\| \right), \quad (16.13)$$

where $\bar{\epsilon}_i^{\text{time}}$ is the padded version of ϵ_i^{time} and λ_2 is another hyperparameter.

16.4.3 Attention-Based Fine-Grained Generation

The fine-grained generation method proposed by MotionDiffuse during the testing phase can produce smooth motions in simple scenarios while maintaining semantic consistency across different parts with the given prompts. However, this zero-shot approach introduces additional challenges. Its correction mechanism only applies smoothing to the coordinate dimensions, which can result in abrupt changes in velocity and acceleration, leading to suboptimal generation outcomes in practical applications. To address this issue, FineMoGen introduces spatial-temporal independent modeling within the attention module. This approach enables the model to be trained on a single-prompt dataset while achieving effective fine-grained generation during testing, delivering significantly improved results.

Baseline Mixed Efficient Attention Imagine the SMA module in ReMoDiffuse without the retrieval branch, which is a combination of efficient self-attention and efficient cross-attention. Let the input motion feature sequence and text feature

sequence be denoted as $\mathbf{X}_m \in \mathbb{R}^{N_m \times L_m}$ and $\mathbf{X}_t \in \mathbb{R}^{N_t \times L_t}$, respectively. These two feature vectors are then projected linearly using weights $W_m^V \in \mathbb{R}^{L_m \times (H \cdot L_g)}$ and $W_t^V \in \mathbb{R}^{L_t \times (H \cdot L_g)}$. The results are combined to form the matrix $\mathbf{V} \in \mathbb{R}^{(N_m+N_t) \times H \times L_g}$. Here, N_m and N_t represent the lengths of the motion and text sequences, while L_m and L_t correspond to their respective dimensionalities. H is the number of heads in the multi-head attention mechanism, and L_g is the dimensionality of the global templates. $\mathbf{V}_{j,i}$ indicates the feature vector for i -th body part, generated from the j -th element. Similarly, a matrix $\mathbf{K} \in \mathbb{R}^{H \times N_g \times (N_m+N_t)}$ is generated, where N_g denotes the number of global templates per head. A softmax operation is then applied along the last dimension of $\mathbf{K}$, the result of which we denote as $\hat{\mathbf{K}}$. Here, $\hat{\mathbf{K}}_{i,k,j}$ indicate the significance of $\mathbf{V}_{j,i}$ for the k -th global template and i -th body part. The features in $\mathbf{V}$ are then aggregated into $H \times N_g$ groups, weighted by the normalized values from $\mathbf{K}$, results in the global templates $\mathbf{G} \in \mathbb{R}^{H \times N_g \times L_g}$. Formally, this process is mathematically expressed as follows:

$$\mathbf{V} = [W_m^V \mathbf{X}_m; W_t^V \mathbf{X}_t], \mathbf{K} = [W_m^K \mathbf{X}_m; W_t^K \mathbf{X}_t], \mathbf{G} = \text{Softmax}(\mathbf{K})\mathbf{V}, \quad (16.14)$$

where $[\cdot; \cdot]$ represents the concatenation of two tensors.

In a manner akin to traditional self-attention, $\mathbf{Q} \in \mathbb{R}^{N_m \times H \times N_g}$ is generated to query the global templates. The element $\mathbf{Q}_{k,i,j}$ represents the relevance of the i -th global template $\mathbf{G}_{i,j}$ to the k -th element. Softmax operation is applied along the last dimension and then compute the product between $\mathbf{Q}$ and $\mathbf{G}$ to obtain the refined features, which serve as the output of this attention module. This process can be expressed as

$$\mathbf{Q} = W_m^Q \mathbf{X}_m, \mathbf{Y} = \text{Softmax}(\mathbf{Q})\mathbf{G}, \quad (16.15)$$

where $\mathbf{Y} \in \mathbb{R}^{N_m \times H \times L_g}$ is the output matrix.

FineMoGen revisits the role of global templates and multi-head attention here, incorporating temporal independence and spatial independence, and introduces the SpAtio-Temporal Mixture Attention (SAMI). Specifically, SAMI contains two branches: temporal branch and spatial branch. The output of them are then combined as the final output: $\mathbf{Y}_{k,i} = \mathbf{Y}_{k,i}^T + \mathbf{Y}_{k,i}^S$, where $\mathbf{Y}_{k,i}^T$ and $\mathbf{Y}_{k,i}^S$ are the refined feature for the k -th element's i -th body part from temporal branch and spatial branch, respectively. We will discuss the detailed implementation of these two branches in the following paragraphs.

Temporal Branch To model interactions between different time intervals in a scalable way, SAMI explicitly incorporates the concept of time into the process. Formally, SAMI proposes the following approximation for refining temporal features:

$$\mathbf{Y}_{k,i}^T \approx \mu_i(x_k) = \sum_{j=1}^{N_g} \mathbf{G}'_{i,j}(x_k) \cdot \mathbf{G}_{i,j}^*(x_k). \quad (16.16)$$

In this context, x_k represents the time position of the k -th pose state in the motion sequence. $\mathbf{G}'_{i,j}(x)$ refers to the time-varying signal derived from the feature vector $\mathbf{G}_{i,j}$, while $\mathbf{G}^*_{i,j}(x_k)$ captures the relative importance of this template at the k -th position in the sequence. SAMI further defines $\mathbf{G}'_{i,j}(x)$ and $\mathbf{G}^*_{i,j}(x_k)$ as follows:

$$\mathbf{G}'_{i,j}(x) = \mathbf{G}^s_{i,j} + \mathbf{G}^v_{i,j} \cdot (x - \mathbf{G}^t_{i,j}), \mathbf{G}^*_{i,j}(x_k) = \frac{e^{-(x_k - \mathbf{G}^t_{i,j})^2 / \sigma^2}}{\sum_{l \in [1, N_g]} e^{-(x_k - \mathbf{G}^t_{i,l})^2 / \sigma^2}}, \quad (16.17)$$

In this framework, the j -th global template of the i -th group is treated as a set of signals radiating outward from the temporal center $\mathbf{G}^t_{i,j}$. SAMI conceptualizes a global template $\mathbf{G}_{i,j}$ as an anchor, with its initial state defined as $\mathbf{G}^s_{i,j}$ and its velocity described by $\mathbf{G}^v_{i,j}$. These three matrices are derived from the original $\mathbf{G}_{i,j}$ using three distinct linear projections.

Spatial Branch In traditional motion generative models [29, 41, 46], the applied FFN module combines data from different body parts, which complicates body part-aware modeling. To address this, SAMI manually segments the raw representation into N_S groups, with each group corresponding to a specific body part. Additionally, SAMI sets $N_S = N_g$ to map each attention head directly to one body part.

As a supplement, SAMI introduces the spatial feature refinement here to allow the communication between different body parts:

$$\mathbf{Y}^S_{k,i} = \sum_{j=1}^{N_S} \mathbf{B}_{k,j} \cdot \omega_{i,j}. \quad (16.18)$$

In this context, $\mathbf{B}_{k,j} \in \mathbb{R}^{L_g}$ represents the projected feature of the j -th body part for the k -th element. The parameter $\omega_{k,j} \in \mathbb{R}^{N_S \times N_S}$ is a learnable weight that captures the relative significance between body parts. This parameter ω is shared across all values of k and i to compute $\mathbf{Y}^S_{k,i}$. It can be interpreted as a shared knowledge that encodes the correlations between different body parts.

16.5 Experiments

In this section, we present the experimental results related to text-driven motion generation. In Sect. 16.5.1, we introduce the datasets used and the corresponding evaluation metrics. In Sect. 16.5.2, we discuss the experimental results for single text prompts, while in Sect. 16.5.3, we present the results for fine-grained text descriptions.

16.5.1 *Datasets and Evaluation Metrics*

Datasets Here, we introduce six datasets related to text-driven motion generation and two datasets related to action-conditioned motion generation.

- **KIT Motion Language Dataset** [30]: This dataset integrates human motion data with natural language annotations, comprising 3,911 motions and 6,353 textual descriptions collected from various institutions. It features a wide range of motion types, including gestures, locomotion, manipulation, and sports. Additionally, variations in speed, body parts involved, and directions provide support for targeted research.
- **HumanML3D** [11]: Built on reannotated versions of the HumanAct12 [10] and AMASS [24] datasets, this large-scale scripted dataset includes 14,616 motions paired with 44,970 descriptions. It spans a diverse set of human actions, from everyday activities to sports and artistry, making it the most comprehensive collection of scripted human motions.
- **Babel** [31]: This dataset extends the AMASS[24] motion collection with approximately 43 hours of data, annotated with 28k sequential labels and 63k frame-level labels. Sequential labels describe the overarching action in each motion sequence, while frame labels precisely mark all concurrent actions, enabling the study of overlapping activities with temporally aligned annotations.
- **HuMMan-MoGen** dataset [47]: A large-scale benchmark for evaluating fine-grained spatiotemporal motion generation. Built on the HuMMan [5] dataset, it focuses on 2,968 videos representing 160 distinct fitness actions, annotated with detailed descriptions to reflect fine-grained motion semantics across various body parts and time frames.
- **HumanAct12** dataset [10]: This dataset comprises 12 categories of human motions derived from the PHSPD dataset [50], which includes 1,191 videos. These videos are processed into trimmed motion clips, allowing for focused analysis of individual motion categories.
- **UESTC** dataset [15]: An essential benchmark for action-conditioned motion generation, featuring 25,000 motion sequences across 40 action categories. Additionally, ACTOR [28] extracts SMPL [23] sequences from UESTC using a pre-trained VIBE [17] model and provides an action recognition model for evaluation.

Evaluation Metrics In the following tasks, we use the following five metrics, with FID being the primary metric, representing the quality of motion generation.

- **Fréchet Inception Distance (FID)**: This metric quantifies the distance between the feature representations of real and generated motion sequences, offering a way to assess the quality of motion generation.
- **R-Precision**: This evaluates the alignment between text descriptions and the generated motion sequences. It measures the probability of the correct text appearing in the top-k rankings after sorting. For each generated motion, its ground-truth

description is mixed with 31 randomly selected incorrect descriptions from the test set, creating a pool of descriptions. The Euclidean distances between the motion features and each description in the pool are computed and ranked. The average accuracy is then measured at top-1, top-2, and top-3 positions. In action-conditioned motion generation task, the metric **Accuracy** is used instead. Since we have motion recognition model, we can directly calculate the consistency between the generated motion sequences and the given action categories.

- **Diversity**: This metric gauges the variety and richness of generated action sequences by comparing two randomly selected subsets of equal size, conditioned on different descriptions. It is calculated as follows [11]:

$$Diversity = \frac{1}{S_d} \sum_{i=1}^{S_d} \|\mathbf{v}_i - \mathbf{v}'_i\|,$$

where $\mathbf{v}_i$ and $\mathbf{v}'_i$, for $i = 1, 2, \dots, S_d$, represent the motion features from the two subsets.

- **Multimodality(MM)**: This measures the average variation in generated motion sequences based on a single text description. For each description, two subsets of equal size S_m are randomly selected, and multimodality is computed as follows [11]:

$$Multimodality = \frac{1}{S_m \times C} \sum_{c=1}^C \sum_{i=1}^{S_m} \|\mathbf{v}_{c,i} - \mathbf{v}'_{c,i}\|,$$

where $\mathbf{v}_{c,i}$ and $\mathbf{v}'_{c,i}$, for $i = 1, 2, \dots, S_m$, are the motion features of the two subsets.

- **Multimodal Distance (MM Dist)**: This metric calculates the average Euclidean distance between the motion feature representations and their corresponding text description features.

16.5.2 Evaluation on Single Prompt

Overall Quantitative Results Tables 16.2 and 16.3 compare the three methods introduced in this chapter with previous works. From these comparisons, it is evident that the diffusion model shows a significant improvement over other motion generative models. Table 16.4 shows the quantitative results on action-conditioned motion generation task. We were pleasantly surprised to find that, although MotionDiffuse was not specifically designed for this task, its flexibility in text conditioning, combined with the powerful capabilities of the motion diffusion model, led to significant improvements in both tasks on this dataset.

Table 16.2 Quantitative results on the HumanML3D test set. For a fair comparison, all methods use the real motion length from the ground truth as the extra given information. ‘ $\uparrow$ ’ (‘ $\downarrow$ ’) indicates that the values are better if the metric is larger (smaller). The evaluations are repeated 20 times. $x \pm y$ indicates that the average metric is x and the the 95% confidence interval is y . The best result and the second best result are in red cells and blue cells, respectively

Methods	R Precision $\uparrow$			FID $\downarrow$	MM Dist $\downarrow$	Diversity $\uparrow$	MM $\uparrow$
	Top 1	Top 2	Top 3				
Real motions	0.511 $\pm$.003	0.703 $\pm$.003	0.797 $\pm$.002	0.002 $\pm$.000	2.974 $\pm$.008	9.503 $\pm$.065	–
Language2Pose [1]	0.246 $\pm$.002	0.387 $\pm$.002	0.486 $\pm$.002	11.02 $\pm$.046	5.296 $\pm$.008	7.676 $\pm$.058	–
Text2Gesture [4]	0.165 $\pm$.001	0.267 $\pm$.002	0.345 $\pm$.002	7.664 $\pm$.030	6.030 $\pm$.008	6.409 $\pm$.071	–
MoCoGAN [42]	0.037 $\pm$.000	0.072 $\pm$.001	0.106 $\pm$.001	94.41 $\pm$.021	9.643 $\pm$.006	0.462 $\pm$.008	0.019 $\pm$.000
Dance2Music [18]	0.033 $\pm$.000	0.065 $\pm$.001	0.097 $\pm$.001	66.98 $\pm$.016	8.116 $\pm$.006	0.725 $\pm$.011	0.043 $\pm$.001
Guo et al. [11]	0.457 $\pm$.002	0.639 $\pm$.003	0.740 $\pm$.003	1.067 $\pm$.002	3.340 $\pm$.008	9.188 $\pm$.002	2.090 $\pm$.083
MotionDiffuse [46]	0.491 $\pm$.001	0.681 $\pm$.001	0.782 $\pm$.001	0.630 $\pm$.001	3.113 $\pm$.001	9.410 $\pm$.049	1.553 $\pm$.042
ReMoDiffuse [45]	0.510 $\pm$.005	0.698 $\pm$.006	0.795 $\pm$.004	0.103 $\pm$.004	2.974 $\pm$.016	9.018 $\pm$.075	1.795 $\pm$.043
FineMoGen [47]	0.504 $\pm$.002	0.690 $\pm$.002	0.784 $\pm$.002	0.151 $\pm$.008	2.998 $\pm$.008	9.263 $\pm$.094	2.696 $\pm$.079

Table 16.3 Quantitative results on the KIT-ML test set

Methods	R Precision $\uparrow$			FID $\downarrow$	MM Dist $\downarrow$	Diversity $\uparrow$	MM $\uparrow$
	Top 1	Top 2	Top 3				
Real motions	0.424 $\pm$.005	0.649 $\pm$.006	0.779 $\pm$.006	0.031 $\pm$.004	2.788 $\pm$.012	11.08 $\pm$.097	—
Language2Pose [1]	0.221 $\pm$.005	0.373 $\pm$.004	0.483 $\pm$.005	6.545 $\pm$.072	5.147 $\pm$.030	9.073 $\pm$.100	—
Text2Gesture [4]	0.156 $\pm$.004	0.255 $\pm$.004	0.338 $\pm$.005	12.12 $\pm$.183	6.964 $\pm$.029	9.334 $\pm$.079	—
MoCoGAN [42]	0.022 $\pm$.002	0.042 $\pm$.003	0.063 $\pm$.003	82.69 $\pm$.242	10.47 $\pm$.012	3.091 $\pm$.043	0.250 $\pm$.009
Dance2Music [18]	0.031 $\pm$.002	0.058 $\pm$.002	0.086 $\pm$.003	115.4 $\pm$.240	10.40 $\pm$.016	0.241 $\pm$.004	0.062 $\pm$.002
Guo et al. [11]	0.370 $\pm$.005	0.569 $\pm$.007	0.693 $\pm$.007	2.770 $\pm$.109	3.401 $\pm$.008	10.91 $\pm$.119	1.482 $\pm$.065
MotionDiffuse [46]	0.417 $\pm$.004	0.621 $\pm$.004	0.739 $\pm$.004	1.954 $\pm$.062	2.958 $\pm$.005	11.10 $\pm$.143	0.730 $\pm$.013
ReMoDiffuse [45]	0.427 $\pm$.014	0.641 $\pm$.004	0.765 $\pm$.055	0.155 $\pm$.006	2.814 $\pm$.012	10.80 $\pm$.105	1.239 $\pm$.028
FineMoGen [47]	0.432 $\pm$.006	0.649 $\pm$.005	0.772 $\pm$.006	0.178 $\pm$.007	2.869 $\pm$.014	10.85 $\pm$.115	1.877 $\pm$.093

Table 16.4 Quantitative results for action-conditioned motion generation

Methods	HumanAct12				UESTC			
	FID↓	Accuracy↑	Diversity→	MM→	FID↓	Accuracy↑	Diversity→	MM→
Real motions	0.020	0.997	6.850	2.450	2.79	0.988	33.34	14.16
Action2Motion [10]	0.338	0.917	6.879	2.511	—	—	—	—
ACTOR [28]	0.12	0.955	6.84	2.53	23.43	0.911	31.96	14.52
INR [7]	0.088	0.973	6.881	2.569	15.00	0.941	31.59	14.68
MotionDiffuse [46]	0.07	0.992	6.85	2.46	9.10	0.950	32.42	14.74

Table 16.5 Ablation of MotionDiffuse on KIT-ML test set

Pretrained CLIP	Efficient attention	Stylization block	R Precision $\uparrow$			Time(s) $\downarrow$
			Top 1	Top 2	Top 3	
–	–	–	0.109 $\pm$.003	0.198 $\pm$.003	0.254 $\pm$.003	0.002
No	No	Yes	0.288 $\pm$.004	0.440 $\pm$.004	0.539 $\pm$.004	0.317
No	Yes	Yes	0.136 $\pm$.003	0.233 $\pm$.003	0.309 $\pm$.003	0.129
Yes	No	Yes	0.357 $\pm$.004	0.555 $\pm$.004	0.679 $\pm$.005	0.317
Yes	Yes	No	0.152 $\pm$.003	0.247 $\pm$.003	0.329 $\pm$.003	0.117
Yes	Yes	Yes	0.417 $\pm$.004	0.621 $\pm$.004	0.739 $\pm$.004	0.129

Table 16.6 Ablation of ReMoDiffuse on KIT-ML test set

Retrieval		Attention		FID $\downarrow$
Text	Motion	Text	Motion	
No	No	No	No	0.245 $\pm$.008
Yes	No	No	Yes	0.314 $\pm$.012
Yes	Yes	No	Yes	0.192 $\pm$.008
Yes	No	Yes	Yes	0.307 $\pm$.010
Yes	Yes	Yes	Yes	0.155 $\pm$.006

Evaluation of Motion Decoder Design Table 16.5 showcases ablation studies on the motion decoder design within the MotionDiffuse framework. The results highlight two key findings. First, incorporating the current diffusion time step into the transformer design significantly enhances the model’s performance, aiding in more effective denoising. Second, replacing traditional self-attention with an efficient attention mechanism not only improves computational efficiency but also enhances the model’s ability to capture global attention, enabling better planning of the entire motion sequence.

Table 16.6 presents ablation studies on how retrieval mechanisms improve motion generation quality in ReMoDiffuse. The first row shows experiments without retrieval, which already demonstrate a significant improvement compared to MotionDiffuse’s FID of 1.954. This enhancement primarily stems from two factors: (1) ReMoDiffuse employs mixed efficient attention, integrating efficient self-attention and efficient cross-attention within a unified module; and (2) the use of classifier-free guidance in ReMoDiffuse. These strategies establish a robust baseline for the model. The second row examines the approach of using text to retrieve semantically similar motions and directly applying the retrieved motion features in the attention mechanism. Surprisingly, this method performs slightly worse than the baseline, indicating that retrieval augmentation in motion generation faces significant challenges. Comparing rows 3, 4, 5, and 6 reveals two critical aspects: incorporating kinematic similarity during retrieval and accounting for the semantic discrepancy between the retrieved motion and the target motion during attention are essential considerations for improving generation performance.

16.5.3 *Evaluation on Fine-Grained Prompt*

Overall Quantitative Results As shown in Table 16.7, both MotionDiffuse and FineMoGen exhibit a significant accuracy improvement compared to the previous VAE-based method, TEACH. Additionally, MotionDiffuse uses a zero-shot approach, and the zero-shot version of FineMoGen also performs well, which strongly demonstrates the effectiveness of these two methods.

16.6 Conclusion

This chapter discussed the task of text-driven motion generation. In the design of MotionDiffuse, it is evident that diffusion models hold significant potential as motion generative frameworks. On one hand, they demonstrate superior generative capabilities when dealing with complex conditions and diverse motion data distributions, showing notable improvements over previous architectures such as VAE and GAN. On the other hand, diffusion models allow for fine-grained control during the testing phase, enabling versatile control strategies that greatly benefit downstream applications. Building on this, we further explored retrieval-augmented generation techniques. The results reveal that if semantic similarity and kinematic similarity are thoroughly considered during the retrieval and enhancement stages, the quality of generation can be substantially improved. For fine-grained generation tasks, the findings indicate that compared to sampling strategies employed during the testing phase, modeling temporal and spatial independence at the attention stage from the perspective of model structure leads to better performance. This also underscores that the design of attention mechanisms for text-driven motion generation still holds significant room for innovation and optimization.

Table 16.7 Quantitative results of temporal composition on the BABEL test set and HuMMan-MoGen test set. “corr” indicates correction item. “zs” and “fs” denote “zero-shot” and “fully supervised,” respectively

Methods	BABEL				HuMMan-MoGen			
	R Prec↑	FID↓	Diversity →	MM↑	R Prec↑	FID↓	Diversity →	MM↑
Real motions	0.62	0.004	8.51	3.57	0.49	0.007	7.28	2.15
TEACH [3]	0.46	1.12	8.28	7.14	0.38	1.59	7.04	6.37
PriorMDM [36]	0.43	1.04	8.14	7.39	0.31	1.43	6.89	6.45
MotionDiffuse [46] (w/o corr)	0.42	1.58	8.21	7.62	–	–	–	–
MotionDiffuse [46] (w/ corr)	0.50	0.97	8.59	8.03	–	–	–	–
FineMoGen [47] zs	0.51	0.84	8.01	7.28	0.41	1.34	6.87	6.41
FineMoGen [47] fs	0.56	0.73	7.94	7.17	0.43	1.21	6.71	6.28

Further Reading

Interested readers may refer to the following material:

- Ren et al. [34], Yuan et al. [44] for introducing physical constraints into the motion generation process to achieve higher generation quality.
- Peng et al. [27], Liang et al. [21], Tanaka and Fujiwara [39] introduce frameworks for text-driven human interaction generation, including human-object interaction and human-human interaction.
- Athanasiou et al. [2] for text-driven motion editing.
- Cai et al. [6], Qing et al. [32] for long-term 3D animation generation.

References

1. Ahuja, C., Morency, L.P.: Language2Pose: natural language grounded pose forecasting. In: 2019 International Conference on 3D Vision (3DV), pp. 719–728. IEEE, Piscataway (2019)
2. Athanasiou, N., Ceske, A., Diomataris, M., Black, M.J., Varol, G.: Motionfix: text-driven 3D human motion editing. In: SIGGRAPH Asia 2024 Technical Communications (2024)
3. Athanasiou, N., Petrovich, M., Black, M.J., Varol, G.: Teach: temporal action composition for 3D humans. In: 2022 International Conference on 3D Vision (3DV), pp. 414–423. IEEE, Piscataway (2022)
4. Bhattacharya, U., Rewkowski, N., Banerjee, A., Guhan, P., Bera, A., Manocha, D.: Text2gestures: a transformer-based network for generating emotive body gestures for virtual agents. In: 2021 IEEE Virtual Reality and 3D User Interfaces (VR), pp. 1–10. IEEE, Piscataway (2021)
5. Cai, Z., Ren, D., Zeng, A., Lin, Z., Yu, T., Wang, W., Fan, X., Gao, Y., Yu, Y., Pan, L., et al.: HuMMan: multi-modal 4D human dataset for versatile sensing and modeling. In: European Conference on Computer Vision, pp. 557–577. Springer, Berlin (2022)
6. Cai, Z., Jiang, J., Qing, Z., Guo, X., Zhang, M., Lin, Z., Mei, H., Wei, C., Wang, R., Yin, W., et al.: Digital life project: autonomous 3D characters with social intelligence. Preprint. arXiv:2312.04547 (2023)
7. Cervantes, P., Sekikawa, Y., Sato, I., Shinoda, K.: Implicit neural representations for variable length human motion generation. In: European Conference on Computer Vision, pp. 356–372. Springer, Berlin (2022)
8. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. *Adv. Neural Inf. Proces. Syst.* **34**, 8780–8794 (2021)
9. Ghosh, A., Cheema, N., Oguz, C., Theobalt, C., Slusallek, P.: Synthesis of compositional animations from textual descriptions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1396–1406 (2021)
10. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2Motion: conditioned generation of 3D human motions. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 2021–2029 (2020)
11. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3D human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5152–5161 (2022)
12. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Adv. Neural Inf. Proces. Syst.* **33**, 6840–6851 (2020)
13. Hong, F., Zhang, M., Pan, L., Cai, Z., Yang, L., Liu, Z.: Avatarclip: zero-shot text-driven generation and animation of 3D avatars. *ACM Trans. Graph.* **41**(4), 1–19 (2022)

14. Huang, R., Hu, H., Wu, W., Sawada, K., Zhang, M., Jiang, D.: Dance revolution: long-term dance generation with music via curriculum learning. Preprint. arXiv:2006.06119 (2020)
15. Ji, Y., Xu, F., Yang, Y., Shen, F., Shen, H.T., Zheng, W.S.: A large-scale RGB-D database for arbitrary-view human action recognition. In: Proceedings of the 26th ACM international Conference on Multimedia, pp. 1510–1518 (2018)
16. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. *Adv. Neural Inf. Proces. Syst.* **33**, 12104–12114 (2020)
17. Kocabas, M., Athanasiou, N., Black, M.J.: Vibe: video inference for human body pose and shape estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5253–5263 (2020)
18. Lee, H.Y., Yang, X., Liu, M.Y., Wang, T.C., Lu, Y.D., Yang, M.H., Kautz, J.: Dancing to music. *Adv. Neural Inf. Proces. Syst.* **32**, 3586–3596 (2019)
19. Li, J., Yin, Y., Chu, H., Zhou, Y., Wang, T., Fidler, S., Li, H.: Learning to generate diverse dance motions with transformer. Preprint. arXiv:2008.08171 (2020)
20. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: AI choreographer: music conditioned 3D dance generation with aist++. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13401–13412 (2021)
21. Liang, H., Zhang, W., Li, W., Yu, J., Xu, L.: Intergen: diffusion-based multi-human motion generation under complex interactions. Preprint. arXiv:2304.05684 (2023)
22. Lin, A.S., Wu, L., Corona, R., Tai, K., Huang, Q., Mooney, R.J.: Generating animated videos of human activities from natural language descriptions. In: NeurIPS Workshop (2018)
23. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: a skinned multi-person linear model. *ACM Trans. Graph.* **34**(6), 1–16 (2015)
24. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: archive of motion capture as surface shapes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5442–5451 (2019)
25. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning, pp. 8162–8171. PMLR (2021)
26. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: towards photorealistic image generation and editing with text-guided diffusion models. In: International Conference on Machine Learning, pp. 16784–16804. PMLR (2022)
27. Peng, X., Xie, Y., Wu, Z., Jampani, V., Sun, D., Jiang, H.: HOI-Diff: text-driven synthesis of 3d human-object interactions using diffusion models. Preprint. arXiv:2312.06553 (2023)
28. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3D human motion synthesis with transformer VAE. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10985–10995 (2021)
29. Petrovich, M., Black, M.J., Varol, G.: TEMOS: generating diverse human motions from textual descriptions. In: European Conference on Computer Vision, pp. 480–497. Springer, Berlin (2022)
30. Plappert, M., Mandery, C., Asfour, T.: The kit motion-language dataset. *Big Data* **4**(4), 236–252 (2016)
31. Punakkal, A.R., Chandrasekaran, A., Athanasiou, N., Quiros-Ramirez, A., Black, M.J.: Babel: bodies, action and behavior with english labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 722–731 (2021)
32. Qing, Z., Cai, Z., Yang, Z., Yang, L.: Story-to-motion: synthesizing infinite and controllable character animation from long text. In: SIGGRAPH Asia 2023 Technical Communications, pp. 1–4 (2023)
33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
34. Ren, J., Zhang, M., Yu, C., Ma, X., Pan, L., Liu, Z.: Insactor: instruction-driven physics-based characters. *Adv. Neural Inf. Proces. Syst.* **36**, 59911–59923 (2024)

35. Saxena, D., Cao, J.: Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Comput. Surv.* **54**(3), 1–42 (2021)
36. Shafir, Y., Tevet, G., Kapon, R., Bermano, A.H.: Human motion diffusion as a generative prior. Preprint. arXiv:2303.01418 (2023)
37. Shen, Z., Zhang, M., Zhao, H., Yi, S., Li, H.: Efficient attention: attention with linear complexities. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3531–3539 (2021)
38. Siyao, L., Yu, W., Gu, T., Lin, C., Wang, Q., Qian, C., Loy, C.C., Liu, Z.: Bailando: 3D dance generation by actor-critic gpt with choreographic memory. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11050–11059 (2022)
39. Tanaka, M., Fujiwara, K.: Role-aware interaction generation from textual description. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15999–16009 (2023)
40. Tevet, G., Gordon, B., Hertz, A., Bermano, A.H., Cohen-Or, D.: Motionclip: exposing human motion generation to clip space. In: *European Conference on Computer Vision*, pp. 358–374. Springer, Berlin (2022)
41. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-or, D., Bermano, A.H.: Human motion diffusion model. In: *The Eleventh International Conference on Learning Representations* (2022)
42. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: Mocogan: decomposing motion and content for video generation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1526–1535 (2018)
43. 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. Proces. Syst.* **30**, 5998–6008 (2017)
44. Yuan, Y., Song, J., Iqbal, U., Vahdat, A., Kautz, J.: Physdiff: physics-guided human motion diffusion model. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16010–16021 (2023)
45. Zhang, M., Guo, X., Pan, L., Cai, Z., Hong, F., Li, H., Yang, L., Liu, Z.: Remodiffuse: retrieval-augmented motion diffusion model. In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, October 1–6, 2023*, pp. 373 (2023)
46. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: text-driven human motion generation with diffusion model. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 4115–4128 (2024)
47. Zhang, M., Li, H., Cai, Z., Ren, J., Yang, L., Liu, Z.: Finemogen: fine-grained spatio-temporal motion generation and editing. *Adv. Neural Inf. Proces. Syst.* **36**, 13981–13992 (2024)
48. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5745–5753 (2019)
49. Zhuang, W., Wang, C., Chai, J., Wang, Y., Shao, M., Xia, S.: Music2Dance: dancenet for music-driven dance generation. *ACM Trans. Multimedia Comput. Commun. Appl.* **18**(2), 1–21 (2022)
50. Zou, S., Zuo, X., Qian, Y., Wang, S., Xu, C., Gong, M., Cheng, L.: 3D human shape reconstruction from a polarization image. In: *European Conference on Computer Vision*, pp. 351–368. Springer, Berlin (2020)

Chapter 17

Text-Driven Scene Generation



Zhaoxi Chen, Guangcong Wang, and Ziwei Liu

Abstract High-quality scene representations, including HDR panoramas, are widely used to achieve photorealistic lighting and immersive reflections in 3D graphics. However, capturing such high-fidelity scene data is often challenging, prompting the need for a versatile and user-friendly generative model that allows intuitive control. This chapter discusses a zero-shot text-driven framework for generating high-resolution 3D scenes directly from free-form textual descriptions, with no paired training data required. The approach synthesizes scenes through two key stages: (1) text-driven generation of low-resolution and low-dynamic-range representations and (2) super-resolution reconstruction to enhance both spatial resolution and dynamic range. To enable zero-shot scene generation, the approach utilizes a dual-codebook architecture to encode diverse environmental features as discrete representations. Moreover, a global sampler is designed based on a vision-language model to interpret input text and sample holistic scene semantics from the global codebook. A local sampler is then used to refine these semantics to synthesize detailed low-resolution scene representations in a patch-by-patch manner. The results not only justify the capability of generating visually compelling scenes with high fidelity but also demonstrate the practical utility of the framework in challenging scenarios such as realistic rendering and immersive virtual environments.

Keywords 3D scene generation · Text to 3D · Vision-language model · HDR panoramas

Z. Chen · Z. Liu (✉)

Nanyang Technological University, Singapore, Singapore

e-mail: zhaoxi001@ntu.edu.sg; ziwei.liu@ntu.edu.sg

G. Wang

Great Bay University, Dongguan, China

17.1 Introduction

The rapid development of the metaverse and virtual reality has significantly increased the demand for photorealistic 3D scene rendering [5, 30]. Among various techniques, high-quality scene representations, such as HDR panoramas, have become essential for achieving realistic lighting and immersive environmental effects in computer graphics (CG). A desirable scene representation for graphics pipelines should feature high resolution, rich dynamic range, and intricate details. However, creating such representations is challenging, as the acquisition process is both resource-intensive and limited by physical constraints. This motivates the development of an automated generation pipeline.

Unlike standard images, HDR panoramas provide 360° coverage of a scene’s radiance in high dynamic range, making them more diverse and detailed. To enhance accessibility and usability, an intuitive method for generating these complex scene representations is crucial, especially for nonexpert users. For instance, due to the high volume of information encoded in a single scene representation, manual design and refinement are impractical. Instead, leveraging free-form text descriptions, such as “a serene lake with a wooden dock surrounded by lush green trees on a sunny day,” offers a more user-friendly approach to control the generation process, as illustrated in Fig. 17.1.

Despite their potential, text-driven scene generation methods face several challenges:

- (1) Resolution: Existing generative models [9, 17, 19, 20, 24] struggle to synthesize ultra-high-resolution (4K+) scene-level content with sufficient detail and diversity.



Fig. 17.1 Zero-shot text-driven scene generation. Our method can generate the HDR scene in 4K+ resolution with a form of panoramic images using free-form text solely. Such generated scenes can be virtually toured as shown on the left

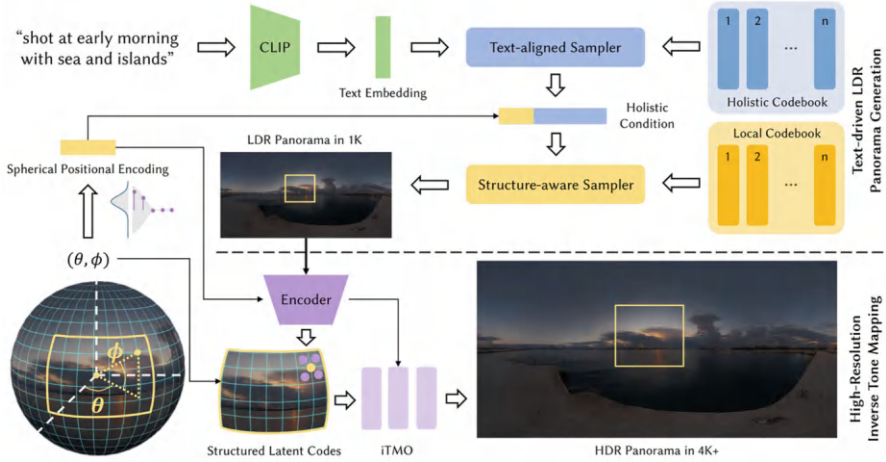


Fig. 17.2 Framework overview. The HDR panorama generation process is divided into two phases. Phase I involves converting the input text into an LDR panorama using a dual-codebook discrete representation. Initially, the input text is embedded through the pre-trained CLIP model [33]. Next, a text-conditioned global sampler extracts holistic semantics from the global codebook, guided by the input text. Following this, a structure-aware local sampler generates local patches and assembles them accordingly. Phase II enhances the LDR output from Phase I by employing structured latent codes as continuous representations. In this phase, we introduce a novel super-resolution inverse tone mapping operator (SR-iTMO), which simultaneously boosts both the spatial resolution and dynamic range of the panorama

- (2) **Coherence:** Unlike object-level images, scene representations often encompass multiple objects and structural layouts, making it difficult to maintain semantic coherence and structural consistency.
- (3) **Text alignment:** Collecting paired data for training models to generate scene-level content from free-form text descriptions is expensive and labor-intensive, leaving zero-shot generation largely unexplored.
- (4) **Dynamic range:** Generating scenes with a high dynamic range poses additional challenges, as converting low-dynamic-range (LDR) outputs into HDR representations can be unstable.

To address these challenges, we discuss a novel framework for zero-shot text-driven scene generation, as illustrated in Fig. 17.2. Due to the inherent complexity of scenes, including their detailed textures, spatial structures, and dynamic ranges, our framework decomposes the generation process into two stages. **(1) Stage I:** Given a textual description, the framework generates a low-resolution, low-dynamic-range (LDR) scene. **(2) Stage II:** The output from Stage I is then enhanced in both resolution and dynamic range using a super-resolution reconstruction module. **Stage I** ensures structural coherence and semantic alignment with the input text without requiring paired data, while **Stage II** produces high-resolution and high-dynamic-range scenes suitable for realistic rendering and immersive applications.

In Stage I, we address the high diversity of environmental features by introducing a dual-codebook architecture, commonly used in vector-quantized models [9, 29]. This architecture captures both global and local scene information in a hierarchical manner. The global codebook encodes overall scene semantics and structure, while the local codebook focuses on fine-grained details. To generate a scene representation, we utilize the pre-trained CLIP model [33] to extract text embeddings from user-provided descriptions. These embeddings guide a global sampler to retrieve holistic features from the global codebook, followed by a local sampler that refines these features into an LDR scene representation with spherical positional encoding.

In Stage II, we treat the generated LDR scene as a continuous field on a spherical domain. This representation complements the discrete codebooks of Stage I by enabling spatial interpolation. Each panorama is encoded into structured latent codes distributed across the sphere, allowing any position (θ, ϕ) to be processed through a multilayer perceptron (MLP)-based super-resolution module. This module enhances both the resolution and dynamic range, producing high-fidelity HDR scene representations.

17.2 Related Work

17.2.1 High-Resolution Generative Models

The field of high-resolution image generation has advanced rapidly, with a focus on models capable of synthesizing images with greater fidelity and detail [3, 4, 10, 12, 15, 17, 19, 20, 36, 37, 43, 47]. The StyleGAN series (including StyleGAN2 and StyleGAN3) [17, 19] has made significant strides in image resolution and detail, allowing the generation of highly realistic images within a confined field of view. However, these models face challenges when it comes to generating complex, diverse scenes. Furthermore, training on entire image frames can be computationally demanding, limiting scalability. Another issue is that these models struggle to maintain both the overall structure and the intricate details of scenes, due to their restricted receptive fields. Patch-based generative approaches [9, 23, 24, 43] address these challenges by operating on smaller patches, allowing for more efficient generation of high-resolution scenes. However, models like Taming Transformer [9] are restricted to generating lower-resolution outputs (e.g., megapixel images) that require specific conditions like depth or segmentation maps. InfinityGAN [24], while claiming to generate infinite-resolution images, falls short in maintaining global scene coherence, leading to fidelity issues when viewed in greater detail.

17.2.2 *Text-Driven Synthesis*

Text-to-image generation has garnered significant attention, with many approaches [7, 28, 35, 38] leveraging supervised learning on paired text-image datasets. More recently, the CLIP model [33] has opened new possibilities for text-to-image generation and manipulation without the need for paired data [2, 14, 16, 21, 26, 31, 41, 48, 53]. Despite these advancements, directly conditioning generative models on CLIP embeddings can struggle with bridging the semantic gap between the textual description and the generated images, especially for complex scenes. The challenge remains in aligning the nuanced semantics of a scene described by text with the generated visual content, making this a difficult problem in practical applications.

17.2.3 *Inverse Tone Mapping*

For photorealistic rendering, capturing the high dynamic range (HDR) of real-world environments is essential. Inverse tone mapping [1] offers a method to convert low dynamic range (LDR) images back into their HDR equivalents. Several convolutional approaches [6, 22, 25, 27, 34, 46, 49, 51, 52] have been proposed to generate HDR from LDR images, but these methods often struggle with handling images at various scales [50]. Our approach addresses this by representing HDR panoramas as continuous fields mapped onto a spherical surface, enabling efficient inverse tone mapping from any resolution using multilayer perceptron (MLP) networks.

17.3 Panoramic Scene Representation

We first introduce our scene representation for this chapter. Panoramic images, or 360° images, are an efficient scene representation in VR and film applications. Different from common scenery images with limited field of views, panoramas can be regarded as 360° spherical fields. To integrate this prior into scene generation, we introduce a spherical positional encoding in this section, or SPE for short, to represent 360° scenes in the entire pipeline.

To effectively capture the inherent geometric structure of 3D scenes, we designed a representation that encodes spatial positional information. This approach ensures the model can learn position-sensitive features critical for generating coherent and realistic scene layouts. For a scene represented as a 2D projection $I \in \mathbb{R}^{H \times W \times 3}$, each pixel is associated with a point on a unit sphere S . Let $I(i, j)$ be the pixel

value at (i, j) in I where $i \in \{1, \dots, H\}$ and $j \in \{1, \dots, W\}$; we represent its spherical representation as $S(\theta, \phi, r)$:

$$\begin{aligned} S(\theta, \phi, r) &= I(i, j), \\ \theta &= (2i/H - 1)\pi, \\ \phi &= (2j/W - 1)\pi/2, \\ r &= 1. \end{aligned} \tag{17.1}$$

This transformation essentially reverses the equirectangular projection used to map a 3D scene onto a 2D plane. Here, the sphere’s center aligns with the camera’s origin in the 3D space, establishing a spatial reference for scene representation.

To further enhance the model’s ability to capture spatial relationships, especially high-frequency details, we leverage Fourier positional encoding $\gamma(\cdot)$ to map (θ, ϕ) to a higher dimensional space:

$$\begin{aligned} \gamma(\theta) &= [\sin(2^0\pi\theta), \cos(2^0\pi\theta), \dots, \sin(2^{L-1}\pi\theta), \cos(2^{L-1}\pi\theta)], \\ \gamma(\phi) &= [\sin(2^0\pi\phi), \cos(2^0\pi\phi), \dots, \sin(2^{L-1}\pi\phi), \cos(2^{L-1}\pi\phi)], \end{aligned} \tag{17.2}$$

where L is set to 4 in our experiments.

This encoding introduces a powerful inductive bias, enabling the model to preserve fine-grained details and spatial consistency in the scene generation process. Specifically, it ensures high-frequency details are retained during the initial stages and incorporates spatial priors in the subsequent steps. Further usage of this representation is detailed in the relevant sections.

17.4 Scene Generation from Texts

Our objective is to generate detailed and high-resolution 3D scenes based on textual descriptions. The task presents several challenges: ensuring scene-level coherence with high-resolution details, achieving zero-shot text-driven generation, and managing dynamic range transitions effectively. Current state-of-the-art methods, such as StyleGAN3 [20] and InfinityGAN [24], encounter difficulties in producing realistic scene-level representations at even moderate resolutions (e.g., 1K). To address these limitations, we introduce a novel framework designed for controllable, text-driven scene generation, as illustrated in Fig. 17.2.

Our method mainly consists of two stages. In **Stage I** (Sect. 17.4.1), we propose a hierarchical framework to generate LDR panorama driven by the text T , which is based on a dual-codebook discrete representation. Driven by the CLIP [33] model, a text-conditioned global sampler learns to sample holistic semantics from the global codebook conditioned on the input text. Furthermore, a structure-aware local sampler learns to synthesize LDR panoramas patch by patch, guided by the holistic

condition. In **Stage II** (Sect. 17.4.2), we propose a new operator which upscales the result from Stage I both in spatial resolution and dynamic range to HDR 4K+ version using a continuous representation. Our method generates ultra-high-resolution and high-dynamic panoramas, which can be directly used in rendering engines or VR applications.

17.4.1 Stage I: Text-Driven LDR Scene Generation

In Stage I, we introduce a hierarchical framework designed to generate low-resolution scenes (1024×512) with low dynamic range, entirely driven by natural language descriptions. This process is divided into three main components:

- **Dual-Codebook Representation:** We leverage two distinct codebooks to capture both the overall scene appearance and the finer local details, ensuring a comprehensive representation of the scene’s features.
- **Text-Conditioned Global Sampler:** The global codebook is used to sample the scene’s broad semantic content, conditioned on the textual input, which defines the overall scene structure and context.
- **Structure-Aware Local Sampler:** Using the holistic scene representation, we synthesize the low-resolution scene by generating local patches, each guided by the global semantic condition to maintain spatial consistency and structural coherence.

An overview of Stage I is illustrated in Fig. 17.2.

17.4.1.1 Dual-Codebook Representation

Given that the textures of scenes are diverse and complicated, a single VQVAE [29] often fails to synthesize coherent appearance and sufficient local details simultaneously. Therefore, we derive a hierarchical discrete representation of LDR panoramas as dual codebooks. Specifically, to represent holistic semantics and coarse but global features, we first build a *global codebook* $\mathcal{Z}_g$ via training a VQVAE [29] to reconstruct a downsampled panorama $I_g \in \mathbb{R}^{128 \times 256 \times 3}$. Let E_g and G_g be the corresponding encoder and decoder, respectively. The encoder E_g turns I_g into a continuous feature $\hat{z}_g = E_g(I_g) \in \mathbb{R}^{8 \times 16 \times c_z}$, and then, the coarse feature z_g is derived from $\mathcal{Z}_g$ via quantization. Furthermore, to represent sufficient details and diverse textures across different scenes, we build a *local codebook* $\mathcal{Z}_l$ via training another VQVAE to reconstruct high-resolution patches $I_l \in \mathbb{R}^{256 \times 256 \times 3}$. Let E_l and G_l be the corresponding encoder and decoder; we can obtain a fine-grained feature $z_l \in \mathbb{R}^{16 \times 16 \times c_z}$ by E_l and $\mathcal{Z}_l$. During training, the patch I_l is randomly cropped from the full LDR panorama $I \in \mathbb{R}^{512 \times 1024 \times 3}$, and c_z is set to 256.

17.4.1.2 Text-Conditioned Global Sampler

While zero-shot text-driven scene synthesis can be performed by directly optimizing a generative model with guidance from CLIP [26, 31], this approach struggles with generating large-scale scenes due to the model’s lack of understanding of the overall scene semantics. It is essential to condition the generation process on global semantics derived from the input text to address this limitation. Motivated by this insight, we introduce a text-conditioned global sampler that samples feature representations from the global codebook, enabling alignment between the text and scene in a language-agnostic training setup.

Our key insight is to align the sampler with the input text using unsupervised learning techniques, specifically K-nearest neighbors and contrastive learning. The goal is to generate a text condition C_{txt} that approximates the true semantic meaning of the target scene I , without relying on paired text-scene training data. Let E_{txt} and E_{img} represent the text and image encoders from CLIP [33], respectively. Drawing inspiration from [53], we begin by perturbing the image feature $E_{\text{img}}(I)$ to generate a pseudo-text feature, thereby bridging the gap between the image and text modalities:

$$\hat{C}_{\text{txt}} = (1 - \alpha)E_{\text{img}}(I) + \alpha\epsilon/\|E_{\text{img}}(I)\|_2/\|\epsilon\|_2, \quad (17.3)$$

where α is a fixed hyper-parameter and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is the Gaussian noise. Furthermore, we fetch the top-K nearest image embeddings $C_{\text{knn}} = \{E_{\text{img}}(I_k)\}_{k=1}^K$ of $\hat{C}_{\text{txt}}$ as the additional condition. Consequently, the text condition C_{txt} is written as

$$C_{\text{txt}} = \{C_{\text{knn}}|\hat{C}_{\text{txt}}\}. \quad (17.4)$$

Specifically, we adapt a transformer-based model [45] as a text-conditioned sampler that samples holistic features from the global codebook $\mathcal{Z}_g$ based on the input text T . The sampler is trained in an autoregressive manner, conditioned on the text-derived feature. The discrete representation corresponds to a sequence, which consists of indices from the global codebook. Sampling the global features is framed as an autoregressive task, where the next index in the sequence is predicted given the previous ones and the text condition. For a given sequence, the transformer-based sampler learns to predict the distribution of the next index, i.e., $p(s_i|s_{<i}, C_{\text{txt}})$. The likelihood of the full sequence is then computed as $p(s) = \prod_i p(s_i|s_{<i}, C_{\text{txt}})$. We optimize the likelihood function to train the sampler, ensuring that it is aligned with the input text:

$$\mathcal{L}_{\text{txt}} = \mathbb{E}_{I \sim p(I)}[-\log p(s)] + \mathcal{L}_{\text{con}}, \quad (17.5)$$

where $\mathcal{L}_{\text{con}}$ is the contrastive regularization

$$\mathcal{L}_{\text{con}} = -\tau \sum_{i=1}^n \log \frac{\exp((E_{\text{img}}(I_i) \cdot \hat{C}_{\text{txt}})/\tau)}{\sum_{j=1}^n \exp((E_{\text{img}}(I_j) \cdot \hat{C}_{\text{txt}})/\tau)}, \quad (17.6)$$

where we maximize the cosine similarity between the real image feature and the pseudo-text feature. $\tau \in (0, 1]$ is a hyper-parameter.

Trained using the loss function in Eq. 17.5, the sampler becomes aligned with real text features, allowing for zero-shot transferability during inference. This is achieved by substituting $\hat{C}_{\text{txt}}$ with $E_{\text{txt}}(T)$, the text feature encoding. Intuitively, since the global codebook represents the full panorama at low resolution (256×128), the text-conditioned global sampler captures a broad overview of the scene and generates the holistic condition z_g . This serves as the guide for the subsequent local patch synthesis step.

17.4.1.3 Structure-Aware Local Sampler

With the holistic condition in hand, our focus shifts to synthesizing local patches that capture fine-grained details. Utilizing the local codebook, we are able to generate high-fidelity individual patches. Additionally, the synthesizer must account for the inherent structure of the panorama to correctly assemble these patches in alignment with the holistic condition. This is where the spherical positional encoding (Sect. 17.3) plays a crucial role, ensuring the proper spatial arrangement of the patches.

Specifically, we use another transformer as a structure-aware sampler, which samples features from the local codebook. Given the holistic condition z_g , our goal is to synthesize the local patch with a spherical representation derived from Eq. 17.1. To incorporate spherical positional encoding (SPE), it is directly prepended to the beginning of the sequence after the input tokens are converted to embeddings. The spherical coordinates form a vector of $2 \times 256 = 512$ dimension. The Fourier-based SPE, as defined in Eq. 17.2, is computed and stacked, which is then concatenated with the embeddings in the transformer.

The structure-aware sampler learns to predict the next possible indices, i.e., $p(s_i | s_{<i}, z_g, \gamma(\theta_l), \gamma(\phi_l))$, given the previous indices, the holistic condition, and the spherical encodings. We maximize the log-likelihood over the entire sequence to obtain the final representation:

$$\mathcal{L}_{\text{local}} = \mathbb{E}_{I_l \sim p(I_l)} [-\log p(s)], \quad (17.7)$$

where $p(s) = \prod_i p(s_i | s_{<i}, z_g, \gamma(\theta_l), \gamma(\phi_l))$.

During inference, we utilize this sampler in a sliding-window manner, as illustrated in Fig. 17.2. For example, to generate a full panorama with the holistic condition z_g , we first generate its SPE as $\theta \in \mathbb{R}^{512 \times 1024}$ and $\phi \in \mathbb{R}^{512 \times 1024}$.

Then, we patch-wisely slide the attention window of the sampler to obtain the full representation $z_l \in \mathbb{R}^{32 \times 64 \times c_z}$. Finally, the full panorama in 1K resolution can be generated using the decoder of the local codebook, i.e., $I = G_l(z_l)$.

17.4.2 Stage II: LDR Scenes to High-Resolution HDR Scenes

In Stage II, the model enhances the output from Stage I by increasing both the spatial resolution and dynamic range, producing a high-quality HDR panorama in 4K+ resolution. To achieve this, we construct a continuous representation that enables querying HDR values at any spherical position on the sphere S :

$$S(\theta, \phi, 1) = f_c(z_c, \theta, \phi), \quad (17.8)$$

where z_c is the feature derived from the LDR patch and $S(\theta, \phi, 1)$ is introduced in Sect. 17.3. An overview of Stage II is shown in Fig. 17.2.

17.4.2.1 Structured Latent Codes

The feature z_c in Eq. 17.8 is derived from a set of latent codes $\mathcal{Z}_c$, which represent features distributed continuously across the sphere. Given an LDR panorama, we first randomly crop a patch, where the corresponding field of view on the sphere is $S'(\theta', \phi', 1)$. The structured latent codes are then obtained as a pixel-aligned feature map $\mathcal{Z}_c = E(I') \in \mathbb{R}^{128 \times 128 \times c'_z}$, where c'_z represents the dimension of each latent code. To compute the feature vector z_c at any given position (θ, ϕ) , we perform an area-based interpolation:

$$z_c = \sum_{i \in \{00, 01, 10, 11\}} \frac{A_i}{A} z_i, \quad z_i \in \mathcal{Z}_c, \quad (17.9)$$

where z_i is the nearest latent code at (θ, ϕ) and A_i is the area of the rectangle as illustrated in Fig. 17.2.

17.4.2.2 Super-Resolution Inverse Tone Mapping Operator

Using the continuous feature z_c , we can employ a composite function $f_c = f_{sr} \circ f_{itmo}$, referred to as the super-resolution inverse tone mapping operator (SR-iTMO), to map value points from LDR patches to their corresponding high-resolution HDR values. The SR-iTMO is composed of two small MLPs, f_{sr} and f_{itmo} , which sequentially increase the spatial resolution and dynamic range of the patches, as shown in Fig. 17.2. The MLP f_{sr} , consisting of four layers, is responsible for

upsampling the resolution of the LDR patch:

$$I_{\text{hr}}^{\text{ldr}}, c_{\text{hr}} = f_{\text{sr}}(z_c), \quad (17.10)$$

where c_{hr} is the intermediate feature of the second layer of f_{hr} . Then, f_{itmo} upscales the dynamic range of the high-resolution patch:

$$I_{\text{hr}}^{\text{hdr}} = f_{\text{itmo}}(c_{\text{hr}}, \theta, \phi), \quad (17.11)$$

where θ and ϕ are the spherical coordinates of the queried point.

17.4.2.3 Training

We adopt an end-to-end training approach for both the encoder E and the SR-iTMO model f_c . Thanks to our continuous representation, the networks can be trained directly at any resolution without needing to downscale the input images. Our dataset consists of HDR scene data with resolutions ranging from 4K to 8K, and we do not rescale them to a fixed resolution for training purposes. Initially, we focus the model on enhancing spatial resolution by minimizing the following loss function:

$$\mathcal{L}_{\text{sr}} = \frac{1}{n} \sum ||I_{\text{hr}}^{\text{ldr}} - \hat{I}_{\text{hr}}^{\text{ldr}}||_1, \quad (17.12)$$

where n denotes the number of pixel samples and $I_{\text{hr}}^{\text{ldr}}$ refers to the model's predicted output from Eq. 17.10.

Furthermore, the model is trained to expand the dynamic range of the generated scene by minimizing a scale-invariant objective, as follows:

$$\mathcal{L}_{\text{itmo}} = \frac{1}{n} \sum D^2(I_{\text{hr}}^{\text{hdr}}, \hat{I}_{\text{hr}}^{\text{hdr}}) - \left[\frac{1}{n} \sum D(I_{\text{hr}}^{\text{hdr}}, \hat{I}_{\text{hr}}^{\text{hdr}}) \right]^2, \quad (17.13)$$

where $D(I_1, I_2) = \log(I_1) - \log(I_2)$ represents the logarithmic distance between two images.

Therefore, the overall training objective for Stage II is the sum of the two components: $\mathcal{L}_{\text{full}} = \mathcal{L}_{\text{sr}} + \mathcal{L}_{\text{itmo}}$. This ensures that the model simultaneously learns to enhance both the resolution and dynamic range of the generated scene.

17.5 Evaluations

For a fair comparison, we demonstrate significant improvements over existing methods in panorama generation, zero-shot text-driven synthesis, and inverse tone mapping.

17.5.1 Panoramic Scene Generation

To assess the quality of the generated panoramas, we employ two common evaluation metrics: Fréchet inception distance (**FID**) [11] and inception score (**IS**) [40]. These metrics measure the quality of the generated images by comparing them to real images in terms of both visual quality and diversity.

In addition to these quantitative metrics, we conduct a user study to further evaluate the perceptual quality of the panoramas. Each panorama is presented individually to 25 users, who are asked to rate:

- **Perceptual Quality (PQ)**: How visually appealing the panorama is.
- **Structural Integrity (SI)**: How well the panorama maintains coherent structures (e.g., no distortion or unnatural seams).

Both scores are on a 5-point scale, where 5 indicates the best possible score. We compare our method against four state-of-the-art generators, as outlined below:

- **StyleGAN2** [19] is a style-based generative model which synthesizes images from a latent distribution. We directly train it on the LDR panoramas with 1K resolution. Note that we employ the adaptive augmentation technique [18] to stabilize training.
- **StyleGAN3** [20] is an alias-free version of StyleGAN2 whose internal representation is equivariant to translation and rotation. We consider two configurations (StyleGAN3-T and StyleGAN3-R) in our experiments.
- **InfinityGAN** [24] is a patch-based method that aims at arbitrary-sized image generation. We use the default setting to train their model on our LDR patches.
- **Taming Transformer** [9] is a VQVAE-based generative model, succeeding in synthesizing high-resolution images with dense conditions, e.g., depth maps and segmentation maps. As there is no direct condition for HDR panoramas, we train a simple model that conditions on pixel coordinates.

We begin by presenting the quantitative results for LDR panorama generation, comparing our method with representative approaches, as shown in Table 17.1. Notably, three variants of StyleGAN are trained directly on the full panorama with 1K resolution, while the remaining two baselines are patch based during training and only synthesize the full panorama during evaluation. In summary, the proposed method achieves the lowest FID and the highest IS among all methods, indicating superior image quality. Additionally, the user study results demonstrate that our

Table 17.1 Quantitative results on panoramic scene generation

Methods	FID ↓	IS ↑	PQ ↑	SI ↑
StyleGAN2	14.62	5.07 ± 0.17	2.43	2.20
StyleGAN3-T	16.61	5.06 ± 0.18	2.57	2.42
StyleGAN3-R	15.13	5.01 ± 0.19	2.19	1.87
InfinityGAN	104.62	4.96 ± 0.16	2.03	1.80
Taming transformer	112.38	3.04 ± 0.08	2.73	2.55
w/o global	308.38	1.82 ± 0.02	1.41	1.36
w/o SP (Eq. 17.1)	56.37	4.22 ± 0.12	3.45	3.18
w/o SPE (Eq. 17.2)	31.19	6.32 ± 0.27	3.23	3.09
Ours	10.72	6.65 ± 0.22	4.26	4.25

method outperforms others in terms of perceptual quality while also maintaining the structural integrity of panoramic scenes.

For qualitative comparison, we show examples of our method alongside baselines in Fig. 17.3. Although the three variants of StyleGAN, trained on full-framed panoramas, achieve relatively low FID scores, they fail to capture the intrinsic structural properties of the panorama, resulting in low fidelity. For instance, the church interior generated by StyleGAN2 appears realistic initially, but the ceiling and bench are distorted and broken. InfinityGAN, on the other hand, generates repeating patterns near the horizontal regions and textureless pixels near the poles. The coordinate-conditioned Taming Transformer struggles to synthesize the full panorama when trained on patches without dense conditions, likely due to the lack of inductive bias in a plain transformer model. In contrast, our method produces high-quality LDR panoramas, thanks to its careful design for global semantic alignment and structure-aware local patch synthesis.

17.5.2 Text-Driven Synthesis

We further evaluate the performance of our framework on zero-shot text-driven panorama synthesis, demonstrating that simple extensions of previous methods are insufficient for this challenging task. As in Sect. 17.5.1, we use FID and IS to assess the quality of the generated panoramas. Additionally, we conduct another user study where participants are asked to rate:

- **Perceptual Quality (PQ)** of the generated panoramas.
- **Textural Consistency (TC)** between the input text and the generated output.

All scores are on a 5-point scale, where 5 represents the best performance. We compare two text-driven synthesis methods as follows:

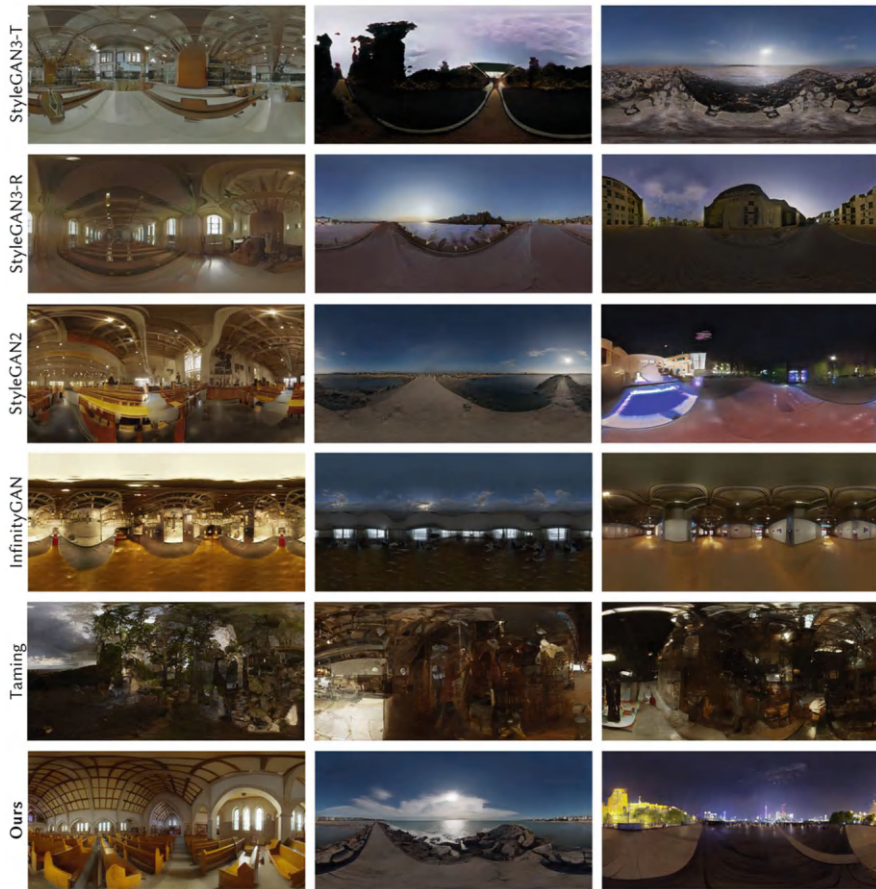


Fig. 17.3 Comparison of LDR panorama generation methods. StyleGAN-based models [19, 20], trained on entire panoramas at 1K resolution, produce results that appear convincing at first glance but reveal noticeable local artifacts upon closer examination. Moreover, these methods are unable to preserve the spherical integrity of panoramic images. InfinityGAN [24], on the other hand, introduces repetitive patterns near the horizon and lacks texture detail near the poles. The coordinate-conditioned Taming Transformer [9] struggles to represent the diversity and intricate structures characteristic of panoramas. Unlike these approaches, our model excels in generating panoramas with rich details and diverse, coherent scenes

- **StyleCLIP** [31] is a method that directly optimizes the latent space of StyleGAN using the guidance of CLIP. It iterates multiple times to maximize the cosine similarity of the generated image and the input text.
- **FuseDream** [26] is an updated version of StyleCLIP, which applies augmentations on images to improve the robustness of aligning free-form texts to the generated image.

Table 17.2 Quantitative results on zero-shot text-driven synthesis

Methods	FID ↓	IS ↑	PQ ↑	TC ↑
StyleCLIP	57.86	4.28 ± 0.18	1.98	1.33
FuseDream	38.16	5.89 ± 0.28	2.41	1.57
w/o KNN	52.85	4.77 ± 0.23	2.14	1.23
w/o $\mathcal{L}_{\text{con}}$	34.70	6.45 ± 0.26	3.81	3.50
Ours	32.01	6.46 ± 0.37	4.38	4.56

We further evaluate the capability of all methods in generating high-fidelity, text-aligned content in a zero-shot manner. The quantitative results, compared with two CLIP+GAN-based methods, are presented in Table 17.2. Our careful design of the text-conditioned global sampler leads to a significant improvement in generation quality.

In addition, we show visual comparisons in Fig. 17.4. A simple combination of generative models and CLIP fails to capture the accurate semantics of the entire scene when given free-form text, as shown by the limitations of StyleCLIP [31]. While the robust optimization process in FuseDream [26] can somewhat enhance the consistency between the content and the input text, it still falls short in fully translating textual information into a coherent panoramic scene. In contrast, our model generates diverse textures that precisely match the input text, even for indoor scenes with high complexity. The hierarchical design of our model allows the synthesizer to align with the scene description on a global level while also generating coherent local details, such as the consistent structure of benches and ceilings in a cathedral interior.

17.5.3 Dynamic Range and Resolution Upscaling

We use mean absolute error (MAE) and root mean square error (RMSE) as metrics to evaluate the performance of our upscaling in Stage II. Additionally, to assess the quality of HDRIs in lighting 3D assets, we render four shader balls with different materials using these HDRIs and compare the resulting visual quality. We compare our method with three existing approaches as follows:

- **LANet** [51] is a luminance-attentive network for HDR reconstruction from a single LDR image. It utilizes convolutional networks with a dedicated branch that outputs segmentation masks for areas with high luminance.
- **ExpandNet** [27] is a fully convolutional model with three branches. It decomposes the input image into local, global, and dilated features, which are fused at the network’s output layer.
- **HDR-CNN** [8] is a convolutional autoencoder for HDR reconstruction. Its encoder transforms an LDR input into a latent feature representation, which the decoder then reconstructs into an HDR image in the logarithmic domain.



Fig. 17.4 Visual comparisons of zero-shot text-driven synthesis. (Above) **baseline comparisons.** The proposed method surpasses optimization-based approaches by generating high-fidelity textures that align seamlessly with the input text, even for highly complex indoor scenes. It achieves a remarkable level of diversity and detail in the textures. (Below) **ablation studies.** Incorporating KNN (K-nearest neighbors) greatly enhances the robustness of the zero-shot synthesis process. Additionally, contrastive regularization enables the model to better align fine-grained visual details with specific textual descriptions

The quantitative comparisons on the testing set are presented in Table 17.3. For a fair comparison, we set the upscaling factor $\beta = 1$ in our method. We demonstrate the potential of implementing iTMO using MLPs, eliminating the need for heavy fully convolutional architectures. Our iTMO achieves the lowest error on the testing set, outperforming the other methods.

Table 17.3 Quantitative results on inverse tone mapping

Methods	MAE ↓	RMSE ↓
LANet	0.3218	51.73
ExpandNet	0.1894	26.68
HDR-CNN	0.1713	26.40
w/ single MLP	0.1798	31.36
Ours	0.1442	26.38

17.5.4 Ablation Studies

17.5.4.1 Effectiveness of the Global Codebook

We demonstrate the significant improvement achieved by our hierarchical framework, particularly through the use of the global codebook $\mathcal{Z}_g$. The term “w/o global” refers to a variant of our model where the global codebook is removed. As shown in Table 17.1 and Fig. 17.5, the quality of generated panoramas decreases drastically without the global codebook. Since the local codebook and structure-aware sampler are trained on image patches, the model cannot effectively synthesize full panoramas without knowledge of the global appearance. Additionally, without the global codebook, our method loses the ability to align the generated content with the input text.

17.5.4.2 SPE for Structural Coherence

To validate the importance of our proposed spherical parametric embedding (SPE) for preserving structural coherence in panoramic images, we implement two baselines. The first, “w/o SP,” removes the spherical representation in Eqs. 17.1 and 17.2 and instead uses horizontal outpainting to synthesize full panoramas. The second, “w/o SPE,” removes only the Fourier feature of SPE in Eq. 17.2. As shown in Table 17.1, our spherical representation plays a crucial role in improving both the overall quality and structural integrity of the generated panoramas. A visual comparison in Fig. 17.5 highlights this further. Without SPE, the model tends to generate repeated textures, while the encoding in Eq. 17.2 significantly enhances the preservation of structural coherence. Specifically, our full model generates continuous elements, such as ceilings and lanes, which are critical for maintaining the fidelity of panoramic scenes.

17.5.4.3 The Learning Objectives Matter for Text-Driven Synthesis

To enhance the alignment between free-form text and the generated scene through language-free training, we leverage unsupervised learning techniques, namely, K-nearest neighbors (KNN, as defined in Eq. 17.4) and contrastive learning (Eq. 17.6).

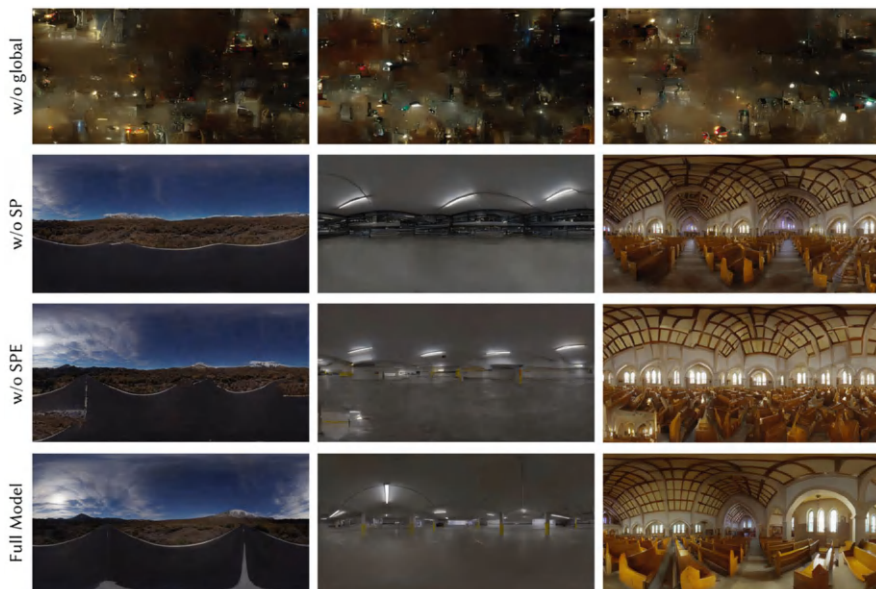


Fig. 17.5 Qualitative ablation studies on LDR panorama generation. The global codebook and its associated sampling step play a vital role in ensuring global coherence in the generated panorama. The use of spherical positional encoding (Eqs. 17.1 and 17.2) is essential for maintaining structural consistency and preventing the repetition of textures. Without this encoding, our model tends to produce repetitive textures that spread horizontally

To assess their contributions, we also train our model without these methods to evaluate their impact on zero-shot transferability. As illustrated in Table 17.2, KNN plays a crucial role in improving both the quality of generation and its consistency with the input text, while contrastive regularization offers a modest enhancement. Qualitative examples in Fig. 17.4 demonstrate that KNN effectively reduces artifacts during text-to-scene alignment. Additionally, contrastive regularization refines our model’s ability to pinpoint and accurately reflect fine-grained details between text and imagery.

17.6 Downstream Applications

17.6.1 Immersive Virtual Reality

Our generation pipeline also unlocks an exciting application: enabling users to create immersive 3D scenes using free-form text descriptions. Users can explore these scenes in any direction for a full VR experience. Two examples are illustrated in the bottom row of Fig. 17.6.

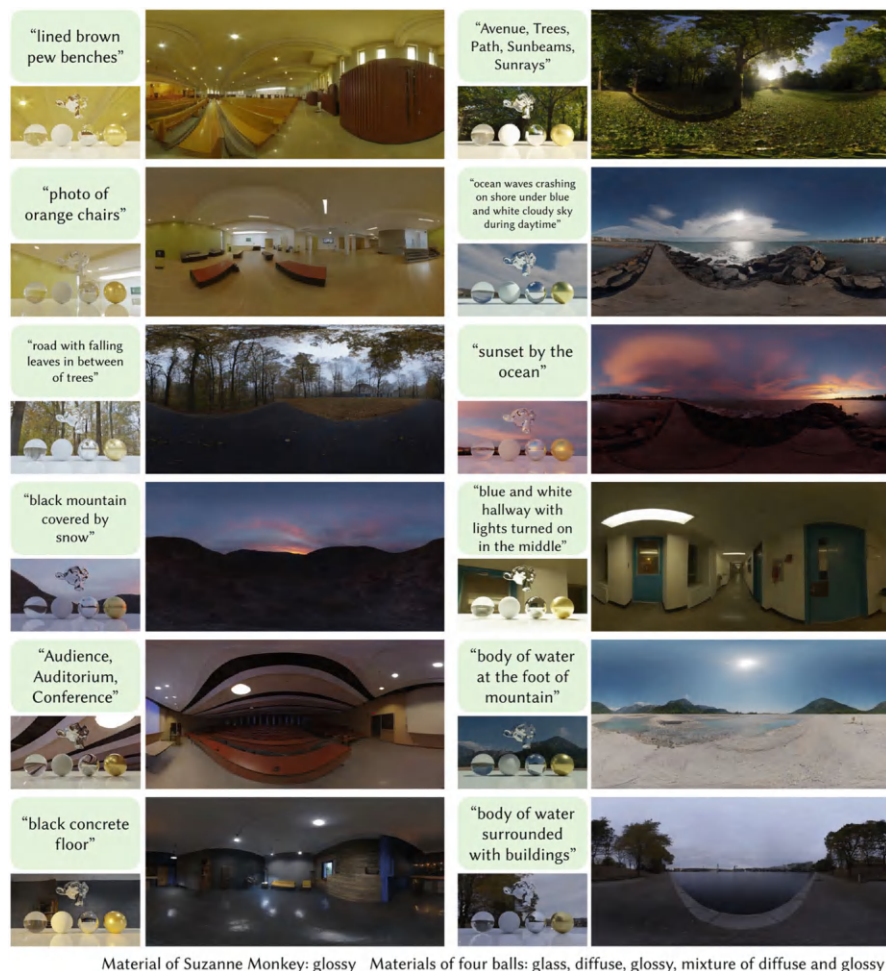


Fig. 17.6 Downstream applications. (1) The generated HDR panoramas can be directly used in lighting 3D scenes to produce photorealistic reflections and illuminations. (2) The model can use text to re-render 3D assets with high fidelity. (3) The panoramic images can be used for immersive virtual reality in 360°

17.6.2 Realistic Rendering

The generated panoramas can seamlessly integrate into modern graphics pipelines to produce photorealistic renderings on 3D assets. Additional results are shown in the top two rows of Fig. 17.6, featuring both indoor and outdoor scenes. Thanks to the careful design of our framework for generating high-quality textures and applying inverse tone mapping, the proposed model can create high-fidelity visuals

and deliver realistic 360° reflections and illumination, all driven by simple text prompts.

17.6.3 *Text-Based Scene Editing*

With the flexibility of our hierarchical generation framework, users can manipulate and edit generated scenes using text-based inputs, as demonstrated in Fig. 17.7. For instance, adding the phrase “with brick road” to “cityscape at night” modifies the road’s appearance in the panoramic scene. Similarly, by altering the holistic condition in a specific area, we can introduce additional “trees” into a rocky landscape.

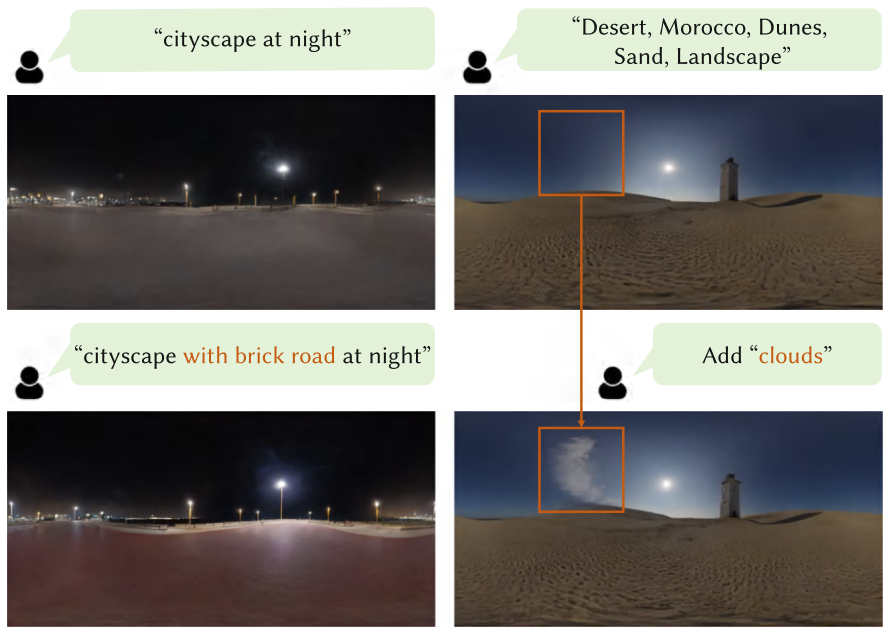


Fig. 17.7 Scene editing from texts. The hierarchical design of our model allows for flexible editing of generated scenes. Users can enhance the scene by appending descriptive phrases or modifying specific regions using additional textual inputs

17.7 Conclusion

In this chapter, we introduced a novel hierarchical framework for text-driven scene generation, with a focus on generating high-fidelity panoramic content. Through our carefully designed global and local codebooks, spherical positional encoding, and continuous latent representation, we demonstrated significant improvements in structural coherence, texture quality, and semantic alignment. Our method excels in various tasks, including zero-shot text-driven synthesis, inverse tone mapping, and photorealistic HDR panorama generation, achieving state-of-the-art performance in both quantitative and qualitative evaluations.

We showcased diverse applications of our framework, from enabling photorealistic rendering in modern graphic pipelines to immersive virtual reality experiences and text-based scene editing, underscoring the versatility and robustness of our approach. By addressing key challenges in panorama generation and HDR synthesis, our model provides a powerful and scalable solution that bridges the gap between natural language and high-quality 3D scene creation, paving the way for more intuitive and creative tools in virtual content generation.

Further Reading

The framework we discussed still has limitations: **(1) Text Bias.** Since we do not use any paired data for text-driven synthesis, the performance relies on CLIP model [33]. Interestingly, we observe that CLIP tends to be biased on some word-image pairs in our framework. **(2) Uncommon Scenery.** The model cannot synthesize scene-level contents that are uncommon in the dataset. Therefore, when the sentence “tree and aurora rays” is given, our model only focuses on the term “tree.” We attribute it to the use of KNN during training the text-align sampler. One possible solution would be further exploiting the joint image-text space of CLIP [33] or using diffusion priors [13, 32, 44], which is an open and challenging problem in zero-shot text-driven synthesis tasks. Solving the problem may also involve using more powerful generative priors [38, 39] or large-scale text-image paired data [42].

References

1. Banterle, F., Ledda, P., Debattista, K., Chalmers, A.: Inverse tone mapping. In: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia - GRAPHITE '06, p. 349. ACM Press, Kuala Lumpur (2006)
2. Blattmann, A., Rombach, R., Oktay, K., Ommer, B.: Retrieval-augmented diffusion models (2022). <https://doi.org/10.48550/ARXIV.2204.11824>
3. Bond-Taylor, S., Hessey, P., Sasaki, H., Breckon, T.P., Willcocks, C.G.: Unleashing Transformers: Parallel Token Prediction with Discrete Absorbing Diffusion for Fast High-Resolution

- Image Generation from Vector-Quantized Codes. arXiv:2111.12701 [cs] (2021). ArXiv: 2111.12701
4. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: MaskGIT: Masked Generative Image Transformer. arXiv:2202.04200 [cs] (2022). ArXiv: 2202.04200
 5. Chen, Z., Liu, Z.: Relighting4D: neural relightable human from videos. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
 6. Chen, G., Chen, C., Guo, S., Liang, Z., Wong, K.Y.K., Zhang, L.: HDR video reconstruction: A coarse-to-fine network and a real-world benchmark dataset. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2482–2491. IEEE, Montreal (2021)
 7. Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al.: CogView: mastering text-to-image generation via transformers. Adv. Neural Inf. Proces. Syst. **34**, 19822–19835 (2021)
 8. Eilertsen, G., Kronander, J., Denes, G., Mantiuk, R.K., Unger, J.: HDR image reconstruction from a single exposure using deep CNNs. ACM Trans. Graph. **36**(6), 1–15 (2017)
 9. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12873–12883 (2021)
 10. Gur, S., Benaim, S., Wolf, L.: Hierarchical patch VAE-GAN: generating diverse videos from a single sample. Adv. Neural Inf. Proces. Syst. **33**, 16761–16772 (2020)
 11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc., New York (2017)
 12. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. arXiv:2006.11239 [cs, stat] (2020). ArXiv: 2006.11239
 13. Höllein, L., Cao, A., Owens, A., Johnson, J., Nießner, M.: Text2room: Extracting textured 3D meshes from 2D text-to-image models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7909–7920 (2023)
 14. Hong, F., Zhang, M., Pan, L., Cai, Z., Yang, L., Liu, Z.: Avatarclip: zero-shot text-driven generation and animation of 3D avatars. ACM Trans. Graph. **41**(4), 1–19 (2022)
 15. Huang, X., Mallya, A., Wang, T.C., Liu, M.Y.: Multimodal Conditional Image Synthesis with Product-of-Experts GANs. arXiv:2112.05130 [cs] (2021). ArXiv: 2112.05130
 16. Jiang, Y., Yang, S., Qiu, H., Wu, W., Loy, C.C., Liu, Z.: Text2Human: text-driven controllable human image generation. ACM Trans. Graph. **41**(4), 1–11 (2022). <https://doi.org/10.1145/3528223.3530104>
 17. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4401–4410 (2019)
 18. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: Advances in Neural Information Processing Systems, vol. 33, pp. 12104–12114 (2020)
 19. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)
 20. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems, vol. 34, pp. 852–863. Curran Associates, Inc., New York (2021)
 21. Kim, G., Kwon, T., Ye, J.C.: DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation. arXiv:2110.02711 [cs] (2022). ArXiv: 2110.02711
 22. Lee, S., An, G.H., Kang, S.J.: Deep chain HDRI: reconstructing a high dynamic range image from a single low dynamic range image. IEEE Access **6**, 49913–49924 (2018). ArXiv: 1801.06277

23. Lin, C.H., Chang, C.C., Chen, Y.S., Juan, D.C., Wei, W., Chen, H.T.: COCO-GAN: generation by parts via conditional coordinating. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
24. Lin, C.H., Lee, H.Y., Cheng, Y.C., Tulyakov, S., Yang, M.H.: InfinityGAN: Towards Infinite-Pixel Image Synthesis. arXiv:2104.03963 [cs] (2021). ArXiv: 2104.03963
25. Liu, Y.L., Lai, W.S., Chen, Y.S., Kao, Y.L., Yang, M.H., Chuang, Y.Y., Huang, J.B.: Single-Image HDR Reconstruction by Learning to Reverse the Camera Pipeline. arXiv:2004.01179 [cs, eess] (2020). ArXiv: 2004.01179
26. Liu, X., Gong, C., Wu, L., Zhang, S., Su, H., Liu, Q.: FuseDream: Training-Free Text-to-Image Generation with Improved CLIP+GAN Space Optimization. arXiv:2112.01573 [cs] (2021). ArXiv: 2112.01573
27. Marnerides, D., Bashford-Rogers, T., Hatchett, J., DeBattista, K.: ExpandNet: A Deep Convolutional Neural Network for High Dynamic Range Expansion from Low Dynamic Range Content. arXiv:1803.02266 [cs] (2019). ArXiv: 1803.02266
28. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. arXiv:2112.10741 [cs] (2022). ArXiv: 2112.10741
29. Oord, A.v.d., Vinyals, O., Kavukcuoglu, K.: Neural Discrete Representation Learning. arXiv:1711.00937 [cs] (2018). ArXiv: 1711.00937
30. Pandey, R., Escolano, S.O., Legendre, C., Häne, C., Bouaziz, S., Rhemann, C., Debevec, P., Fanello, S.: Total relighting: learning to relight portraits for background replacement. ACM Trans. Graph. **40**(4) (2021). <https://doi.org/10.1145/3450626.3459872>
31. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. arXiv:2103.17249 [cs] (2021). ArXiv: 2103.17249
32. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: text-to-3D using 2D diffusion. In: The Eleventh International Conference on Learning Representations (2023). <https://openreview.net/forum?id=FjNys5c7VyY>
33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs] (2021). ArXiv: 2103.00020
34. Raipurkar, P., Pal, R., Raman, S.: HDR-cGAN: Single LDR to HDR Image Translation using Conditional GAN. arXiv:2110.01660 [cs, eess] (2021). ArXiv: 2110.01660
35. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Sutskever, I.: Zero-shot text-to-image generation. In International conference on machine learning, pp. 8821–8831 (2021)
36. Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with VQ-VAE-2. Adv. Neural Inf. Proces. Syst. **32** (2019)
37. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs] (2021). ArXiv: 2112.10752
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684–10695 (2022)
39. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding (2022). <https://arxiv.org/abs/2205.11487>
40. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Adv. Neural Inf. Proces. Syst. **29** (2016)
41. Schaldenbrand, P., Liu, Z., Oh, J.: StyleCLIPDraw: Coupling Content and Style in Text-to-Drawing Synthesis. arXiv:2111.03133 [cs] (2021). ArXiv: 2111.03133
42. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C.W., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S.R., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5b: an open large-scale dataset for training next generation image-text models. In: Thirty-Sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2022). <https://openreview.net/forum?id=M3Y74vmsMcY>

43. Skorokhodov, I., Sotnikov, G., Elhoseiny, M.: Aligning latent and image spaces to connect the unconnectable. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 14124–14133. IEEE, Montreal (2021)
44. Tang, S., Zhang, F., Chen, J., Wang, P., Furukawa, Y.: Mvdifussion: Enabling holistic multi-view image generation with correspondence-aware diffusion. arXiv (2023)
45. 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. Proces. Syst.* 30 (2017)
46. Wang, L., Yoon, K.J.: Deep Learning for HDR Imaging: State-of-the-Art and Future Trends. arXiv:2110.10394 [cs, eess] (2021). ArXiv: 2110.10394
47. Wang, G., Yang, Y., Loy, C.C., Liu, Z.: Stylelight: HDR panorama generation for lighting estimation and editing. In: European Conference on Computer Vision (ECCV) (2022)
48. Wang, Z., Liu, W., He, Q., Wu, X., Yi, Z.: CLIP-GEN: Language-Free Training of a Text-to-Image Generator with CLIP. arXiv:2203.00386 [cs] (2022). ArXiv: 2203.00386
49. Wei, W., Guan, L., Liu, Y., Kang, H., Li, H., Wu, Y., Hua, G.: Beyond Visual Attractiveness: Physically Plausible Single Image HDR Reconstruction for Spherical Panoramas. arXiv:2103.12926 [cs, eess] (2021). ArXiv: 2103.12926
50. Xu, Y., Xiao, T., Zhang, J., Yang, K., Zhang, Z.: Scale-invariant convolutional neural networks. Preprint. arXiv:1411.6369 (2014)
51. Yu, H., Liu, W., Long, C., Dong, B., Zou, Q., Xiao, C.: Luminance Attentive Networks for HDR Image and Panorama Reconstruction. arXiv:2109.06688 [cs, eess] (2021). ArXiv: 2109.06688
52. Zhang, J., Lalonde, J.F.: Learning high dynamic range from outdoor Panoramas. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 4529–4538. IEEE, Venice (2017)
53. Zhou, Y., Zhang, R., Chen, C., Li, C., Tensmeyer, C., Yu, T., Gu, J., Xu, J., Sun, T.: LAFITE: Towards Language-Free Training for Text-to-Image Generation. arXiv:2111.13792 [cs] (2022). ArXiv: 2111.13792

Index

A

Adaptive layer normalization (AdaLN), 298
Affinity matrix, 173
AlexNet, 3
ALIGN, 5
APIs, 9
Artificial general intelligence (AGI), 23
Aspect ratio, 33, 34
Autoregressive, 410
Autoregressive learning, 4
Autoregressive modeling, 97

B

Back-projection block, 269
BERT, 4
Binary matching, 230
Black-box, 9
Bounding box, 232
Byte pair encoding (BPE), 120

C

Cache model, 167
Caltech101, 116
Classifier-free guidance, 385
CLIP, 5
CLIP-score, 326
Clustering, 343
Continual learning, 8
Contrastive learning, 29, 30, 96
Cosine similarity, 346

D

Decoder, 315
Detection transformer, 230

DeViSE, 5

Diffusion timesteps, 305
Domain generalization, 189
Domain shifts, 203
Dual-codebook architecture, 403, 406
Dual-encoder, 6, 120

E

ELMo, 4
Embeddings, 4, 5, 13
Encoder, 7, 12
Euclidean distances, 393
Evolutionary search, 194
Expected Calibration Error (ECE), 208

F

Feature adapter, 171
Federated learning, 10
Few-shot learning, 117
Fine-grained classification, 122
Fine-tuning, 127
Flamingo, 5
Fourier-based, 411
Frechet inception distance (FID), 302

G

Gaussian kernel, 275
Global sampler, 403
GloVe, 4
GPT, 4
GPT-4o, 9
GPT-4V, 23

GPUs, 9
 Gradients, 127
 Grid-based voxels, 277

H

Hadamard product, 384
 HDR panoramas, 403
 Hierarchical discrete representation, 409
 High-frequency components, 316
 High-resolution strategy, 32

I

Image-conditioned text generation, 12
 Image-level recognition, 23
 ImageNet, 3
 Image-text data, 23
 Image-text matching, 12
 Image-text pair data, 7
 Immersive reflections, 403
 In-context learning, 94
 Influence function, 289
 Instruction tuning, 94
 InternVL, 23
 Inter-view adapter, 267
 Inverse linear blend skinning, 355

K

Kinematic similarity, 386
 KL-divergence, 358
 K-nearest neighbors, 410

L

LAION-5B, 10
 Large language models (LLMs), 23
 Latent space, 298
 Linear probe, 117
 Long-tail, 11
 Low-frequency components, 316
 LSTM, 4

M

Marching cubes algorithm, 348
 Markov chain, 382
 Masked language modeling, 4, 12
 Masked modeling, 6
 Modulation factors, 317
 Multimodal, 30
 Multi-modal dialog systems, 23
 Multimodal integration, 32

Multimodality, 393
 Multimodal Large language models (MLLMs), 23
 Multi-scale feature, 233

N

Neural architecture search, 187
 Next-token prediction, 4

O

Open-vocabulary, 14, 230
 Open-vocabulary zero-shot image recognition, 249

P

Parameter-efficient tuning, 188, 221
 Parameter size constraint, 194
 Parametric human models, 343
 Photorealistic lighting, 403
 Pixel-aligned feature map, 412
 Pixel-level dense prediction, 249
 Pixel-level recognition, 23
 Point clouds, 267
 Post-hoc calibration, 222
 Pretrained models, 175
 Pre-training, 3
 Prompt engineering, 116
 Prompt ensembling, 126
 Prompting, 3, 116
 Prompt learning, 13, 118
 Pseudo labeling, 249

R

Relative log amplitudes, 320
 Reparameterization trick, 358
 Representation learning, 11, 116, 118
 Residual connection, 171
 ResNet, 3

S

Scaling factors, 325
 Self-attention, 5, 6, 8
 Self-training, 249
 Semantic feature, 203
 Semantic segmentation, 249
 Seq2seq, 4
 Signed distance function, 352
 Sliding-window, 411
 Spectral modulation, 324

Spherical positional encoding, 411
Stable Diffusion, 315
State-of-the-art, 23
Subject-driven generation, 100
Supernet training, 194
Super-resolution reconstruction, 405

T

Temperature scaling, 211
Text-conditioned global sampler, 405
Textual, 221
Textual encoder, 173
TPUs, 9
Transfer learning, 166
Transformer, 12
Transformer-style multi-head attention, 254

U

Unified 3D open-world learning, 267

V

Variational autoencoder, 298
Video temporal grounding, 76
Vision, 23

Vision foundation model, 23
Vision-language, 23
Vision-language models, 2, 118
Vision-language pre-training, 116
Vision-language tasks, 94
Vision transformer (ViT), 4, 29, 63
Visual-linguistic, 23
Visual perception, 23
Visual question answering (VQA), 7
Visual tokens, 34
Vocabulary, 129
Volume rendering, 341

W

Weight entanglement, 194
Word embeddings, 117
Word vectors, 117
Word2Vec, 4
World knowledge, 2

Z

Zero-shot, 13, 96
Zero-shot classification, 23, 167
Zero-shot retrieval, 23