UC Santa Barbara

UC Santa Barbara Electronic Theses and Dissertations

Title

Text as Controller and Crutch: Generative Control and Over-reliance in Vision-Language Models

Permalink

https://escholarship.org/uc/item/13v8z3t1

ISBN

9798186385462

Author

Wu, Qiucheng

Publication Date

2026-05-26

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

University of California
Santa Barbara

Text as Controller and Crutch: Generative Control and Over-reliance in Vision-Language Models

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Qiucheng Wu

Committee in charge:

Professor Shiyu Chang, Chair
Professor William Wang
Professor Lingqi Yan, Mohamed bin Zayed University of Artificial Intelligence
Dr. Yang Zhang
Dr. Handong Zhao

June 2026

The Dissertation of Qiucheng Wu is approved.

Professor William Wang

Professor Lingqi Yan

Dr. Yang Zhang

Dr. Handong Zhao

Professor Shiyu Chang, Committee Chair

May 2026

Text as Controller and Crutch: Generative Control and Over-reliance in Vision-Language Models

Copyright © 2026

by

Qiucheng Wu

To my parents and family, for their unconditional love and encouragement.

To Professor Chang and all my mentors, for guiding me in the pursuit of rigorous research.

To my friends, for their support and companionship.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Shiyu Chang. His broad vision across machine learning, sharp research insights, and rigorous attitude toward scholarship have continually inspired me to grow beyond my own expectations. It has been a great privilege and honor to develop as a researcher under his supervision and guidance. The journey of working with him throughout my five years of Ph.D. study has shaped the way I think, conduct research, and face challenges. I am profoundly grateful for his guidance, trust, and support, and I cannot thank him enough for the lasting impact he has had on my academic and personal growth.

I am also deeply grateful to Dr. Yang Zhang at MIT-IBM Watson AI Lab and Dr. Handong Zhao at Adobe Research. I have been fortunate to have the opportunity to work closely with and learn from both of them throughout my Ph.D. study. Their encouragement, patience, and guidance have shaped me as a researcher and continually motivated me to improve. I would also like to thank my committee members, Professor William Wang and Professor Lingqi Yan, for their constructive feedback and valuable suggestions on my dissertation and research path.

I would also like to express my sincere appreciation to the mentors who have guided and supported my research beyond my committee. I have greatly enjoyed working with Professor Sijia Liu at Michigan State University, whose encouragement and advice helped me navigate many challenges in my research journey. I was also fortunate to work with Professor Atlas Wang and Professor Humphrey Shi during the first year of my Ph.D. study and my internship at Picsart in 2022. Their deep expertise and research insights broadened my perspective and opened new directions for me at an early stage of my Ph.D. During my internships at Adobe Research in 2023, 2024, and 2025, I had the privilege of working closely with Handong Zhao, whose continuous encouragement and guidance

have pushed me toward more impactful research. Through my collaborations with Adobe Research, I am also grateful to Jing Shi, Zhixin Shu, Zhaowen Wang, Trung Bui, Scott Cohen, Ajinkya Kale, Zhe Lin, Tong Yu, Simon Jenni, Kushal Kafle, Tianyu Wang, Seunghyun Yoon, Quan Tran, and many other wonderful researchers at Adobe. Their guidance, collaboration, and willingness to share their expertise made my time at Adobe Research deeply rewarding and helped me grow as a researcher. In addition, I would like to thank Professor Shimeng Yu, Professor Cong Hao, Professor Ying Xu, Professor Lu Ding, Professor Q. Vera Liao, and Dr. Kaizhi Qian for their valuable suggestions and encouragement on several research projects.

I would also like to express my sincere thanks to my collaborators: Yujian Liu, Xinyi Gao, Guanyu Yao, Michael Saxon, Yifan Jiang, Junru Wu, Yimeng Zhang, Akshay Karkal Kamath, Zhiwen Fan, Wuyang Chen, Xuechen Liu, Wantong Li, Janak Sharda, Kai Wang, Gong Zhang, Victor Kulikov, Vidit Goel, and Nikita Orlov. I feel truly fortunate to have worked with them on many research projects, sharing the challenges, progress, and excitement of doing research together. These collaborations have been an important part of my Ph.D. journey and have helped me grow both technically and intellectually. In particular, I would like to thank Yifan Jiang and Junru Wu for generously sharing their research experience with me as senior researchers. Their guidance and support helped me take faster steps toward professional research. I learned a great deal from their ability to identify meaningful research problems, design careful experiments, and communicate ideas clearly through writing and presentations.

I am also grateful to my former and current labmates: Guanhua Zhang, Bairu Hou, Yujian Liu, Jiabao Ji, Xinyi Gao, Guanyu Yao, Jingbo Yang, Li An, Ziyan Jiang, and Congrui Du. I deeply appreciate the time we spent discussing research questions, developing ideas, and supporting one another through the challenges of Ph.D. study. I will also cherish the moments we shared outside of research, which made this journey much

more meaningful and memorable. In addition, I would like to sincerely thank my friends throughout these years: Ziping Liu, Lin Li, Daiqing Qi, Yichi Zhang, Yuchen Tang, Zichen Chen, Bowen Zhang, Zonglin Di, Weixiang Yan, Yuanbo Nie, Yimeng Zhang, Yihua Zhang, Yuguang Yao, Jinghan Jia, and many other friends who have supported me along the way. Your support, companionship, and the time we spent together have meant a great deal to me throughout this journey.

Finally, to my parents and family: your unconditional love and unwavering support have carried me through every step of this journey.

Curriculum Vitæ

Qiucheng Wu

Education

2026	Ph.D. in Computer Science (Expected), University of California, Santa Barbara.
2021	M.S. in Computer Science and Engineering, University of Michigan.
2019	B.S. in Computer Science and Engineering, University of Michigan.
2019	B.S. in Electrical and Computer Engineering, Shanghai Jiao Tong University.

Publications

‡ indicates co-first author.

1. **Qiucheng Wu**, Jing Shi, Simon Jenni, Kushal Kafle, Tianyu Wang, Shiyu Chang, and Handong Zhao. "RetouchIQ: MLLM Agents for Instruction-Based Image Retouching with Generalist Reward". **CVPR, 2026.**
2. **Qiucheng Wu**, Handong Zhao, Zhixin Shu, Jing Shi, Yang Zhang, and Shiyu Chang. "VividCam: Learning Unconventional Camera Motions from Virtual Synthetic Videos". **ICML, 2026.**
3. Guanyu Yao‡, **Qiucheng Wu**‡, Yang Zhang, Zhaowen Wang, Handong Zhao, and Shiyu Chang. "Rethinking the Text-Vision Reasoning Imbalance in MLLMs through the Lens of Training Recipes". **Arxiv.**
4. **Qiucheng Wu**, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. "VSP: Diagnosing the Dual Challenges of Perception and Reasoning in Spatial Planning Tasks for MLLMs". **ICCV, 2025**.
5. Xinyi Gao‡, **Qiucheng Wu**‡, Yang Zhang‡, Xuechen Liu, Kaizhi Qian, Ying Xu, and Shiyu Chang. "A Hierarchical Probabilistic Framework for Incremental Knowledge Tracing in Classroom Settings". **Arxiv.**
6. **Qiucheng Wu**‡, Yifan Jiang‡, Junru Wu‡, Kai Wang, Gong Zhang, Humphrey Shi, Zhangyang Wang, and Shiyu Chang. "Decoding Micromotion in Low-dimensional Latent Spaces from StyleGAN". **CPAL, 2024.**
7. **Qiucheng Wu**‡, Yujian Liu‡, Handong Zhao, Trung Bui, Zhe Lin, Yang Zhang, and Shiyu Chang. "Harnessing the spatial-temporal attention of diffusion models for high-fidelity text-to-image synthesis". **ICCV, 2023.**
8. **Qiucheng Wu**, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. "Uncovering the disentanglement capability in text-to-image diffusion models". **CVPR, 2023.**
9. **Qiucheng Wu**‡, Yifan Jiang‡, Junru Wu‡, Victor Kulikov, Vidit Goel, Nikita Orlov, Humphrey Shi, Zhangyang Wang, and Shiyu Chang. "Broad Spectrum Image Deblurring via An Adaptive Super-Network". **IEEE TIP, 2023.**

10. Yimeng Zhang[‡], Akshay Karkal Kamath[‡], **Qiucheng Wu**[‡], Zhiwen Fan[‡], Wuyang Chen, Zhangyang Wang, Shiyu Chang, Sijia Liu, and Cong Hao. "Data-Model-Circuit Tri-Design for Ultra-Light Video Intelligence on Edge Devices". **ASP-DAC, 2023.**
11. Janak Sharda, Wantong Li, **Qiucheng Wu**, Shiyu Chang, and Shimeng Yu. "Temporal frame filtering for autonomous driving using 3D-stacked global shutter CIS with IWO buffer memory and near-pixel compute". **IEEE TCAS-I, 2023.**
12. Wantong Li, **Qiucheng Wu**, Janak Sharda, Shiyu Chang, and Shimeng Yu. "Temporal Frame Filtering with Near-Pixel Compute for Autonomous Driving". **AICAS, 2022.**
13. Nina Zhou, **Qiucheng Wu**, Zewen Wu, Simeone Marino, and Ivo D Dinov. "Data SifterText: Partially synthetic text generation for sensitive clinical notes". **Journal of Medical Systems, 2022.**
14. Yijie Guo, **Qiucheng Wu**, and Honglak Lee. "Learning action translator for meta reinforcement learning on sparse-reward tasks". **AAAI, 2022.**
15. Simeone Marino, Nina Zhou, Yi Zhao, Lu Wang, **Qiucheng Wu**, and Ivo D Dinov. "DataSifter: statistical obfuscation of electronic health records and other sensitive datasets". **Journal of Statistical Computation and Simulation, 2019.**

Abstract

Text as Controller and Crutch: Generative Control and Over-reliance in Vision-Language Models

by

Qiucheng Wu

Text has become a central interface in modern vision-language models. In visual generative models, text provides an intuitive mechanism for controlling image and video synthesis. In multimodal large language models, however, text can also become a shortcut that causes models to underutilize visual information. This dissertation studies these two roles of text: as a controller for visual generation, and as a crutch that reveals limitations in multimodal understanding.

The first part of this dissertation investigates text-based controllability in visual generation. I study how natural-language instructions can guide image editing, image generation, and video synthesis across different forms of visual control. For image editing, I develop a multimodal large language model agent that translates user requests into executable editing operations in professional software. Since ground-truth edits are difficult to define, I further introduce a generalist reward model that evaluates editing results through multimodal reasoning and provides scalar feedback for reinforcement learning. For video generation, I study text-guided camera control, where natural-language camera descriptions specify complex camera motions. By synthesizing controllable training data and disentangling visual appearance from camera motion, this work enables more flexible video generation guided by both content and camera instructions. I also examine diffusion-based image generation and editing, showing how manipulating the diffusion process and attention mechanisms can improve content preservation, visual attribute

control, and compositional faithfulness.

The second part of this dissertation studies the limitations of multimodal understanding. Although multimodal large language models have achieved strong performance on many vision-language tasks, they often rely heavily on textual cues and struggle when essential information must be inferred from images. I analyze this imbalance through spatial reasoning and planning tasks, showing that current models can fail even in visually simple scenarios when textual descriptions do not provide sufficient guidance. I further explore training strategies that reduce excessive textual hints and encourage models to reason more directly from visual inputs.

Together, these studies show that text can serve as a powerful control interface in vision-language models, while also revealing the risks of excessive reliance on text in multimodal understanding. This dissertation highlights the importance of building systems that use text effectively while developing stronger visual reasoning capabilities.

Contents

Curriculum Vitae **viii**

Abstract **x**

1 Introduction **1**
1.1 Overview . . . 1
1.2 Challenges and Problem Statement . . . 3
1.3 Dissertation Contributions . . . 5
1.4 Organization of the Dissertation . . . 7

2 Tool-Assisted Text-Guided Image Editing **9**
2.1 Introduction . . . 9
2.2 Background and Prior Work . . . 12
2.3 RetouchIQ . . . 13
2.4 Generalist Reward Model . . . 18
2.5 Experiments . . . 21
2.6 Conclusion . . . 27

3 Text-Guided Camera Control in Video Generation **28**
3.1 Introduction . . . 28
3.2 Background and Prior Work . . . 32
3.3 Preliminaries . . . 33
3.4 VividCam . . . 34
3.5 Experiments . . . 39
3.6 Conclusion . . . 46

4 Text-Guided Visual Attribute Control in Diffusion Models **47**
4.1 Introduction . . . 47
4.2 Background and Prior Work . . . 50
4.3 Attribute Disentanglement in Stable Diffusion Models . . . 52
4.4 Experiments . . . 58
4.5 Conclusion . . . 65

5 Region-Aware Attention Control for High-Fidelity Text-to-Image Generation **66**
5.1 Introduction . . . 66
5.2 Background and Prior Work . . . 69
5.3 Method . . . 70
5.4 Experiments . . . 76
5.5 Conclusion . . . 85

6 Measuring Visual and Textual Understanding in MLLMs **86**
6.1 Introduction . . . 86
6.2 Background and Prior Work . . . 89
6.3 The Visual Spatial Planning Benchmark . . . 91
6.4 Experiments . . . 97
6.5 Conclusion . . . 105

7 Enhancing Visual Understanding in MLLMs **106**
7.1 Introduction . . . 106
7.2 Background and Prior Work . . . 108
7.3 Modality Gap in MLLMs . . . 109
7.4 Mitigating Modality Gap . . . 111
7.5 Conclusion . . . 115

8 Conclusion and Future Directions **116**
8.1 Conclusion . . . 116
8.2 Future Directions . . . 117

Bibliography **120**

Chapter 1

Introduction

1.1 Overview

Vision-language models have become a central direction in modern artificial intelligence, enabling machines to connect natural language with visual perception and generation. Recent progress has led to two major lines of development. The first focuses on visual generation, where text serves as a natural interface for controlling images [1–3] and videos [4–7]. Through textual prompts or instructions, users can guide generative models to synthesize visual content [8–10], modify existing images [11–13], and adjust styles and spatial attributes [14–16]. The second focuses on multimodal understanding, where multimodal large language models receive interleaved visual and textual inputs and produce language responses based on both modalities [17–20]. These models have shown strong capabilities in visual question answering, reasoning, planning, and interactive tasks.

Despite this progress, the role of text in vision-language models remains complex. In visual generation, text is a powerful controller. It provides an accessible and expressive way for users to communicate high-level intent without directly manipulating pixels, layouts, editing parameters, or camera trajectories. A single natural-language instruc-

tion can specify desired visual attributes, editing goals, or dynamic camera behavior. However, achieving precise and reliable text-based control remains challenging. Textual instructions are often abstract, ambiguous, or underspecified, while visual generation requires concrete decisions about appearance, structure, motion, and editing operations. This creates a modality gap between user intent expressed in language and the visual transformations produced by the model.

In multimodal understanding, text can play different roles. Multimodal large language models are often trained on data where textual information provides rich descriptions of visual content. As a result, these models may learn to rely heavily on text while underusing the image. When essential information is available only through visual inputs, especially in tasks involving spatial reasoning, visual planning, or diagram understanding, models can fail even when the visual scene appears simple to humans. This imbalance suggests that strong language reasoning does not necessarily imply strong visual reasoning.

This dissertation studies these two facets of text in vision-language models. The first part (Chapters 2 – 5) investigates how text can be used to improve controllability in visual generation and editing. It studies instruction-based image editing with professional software tools, text-guided camera control in video generation, and text-guided control in diffusion-based image generation and editing. The second part (Chapters 6 – 7) examines how over-reliance on text limits multimodal understanding. It analyzes the gap between visual and textual reasoning in multimodal large language models and explores training strategies that encourage models to reason more directly from visual inputs.

Through these studies, this dissertation aims to better understand both the strengths and limitations of text-centric vision-language modeling. Text provides a powerful interface for controlling visual systems, but excessive dependence on text can weaken visual grounding and reasoning. By examining text as both “controller” and “crutch”, this

dissertation provides insights toward building vision-language models that are more controllable, visually grounded, and reliable.

1.2 Challenges and Problem Statement

Although vision-language models have made rapid progress, open challenges remain on both the generation side and the understanding side. On the generation side, text is expected to serve as a flexible control interface for visual content. The central questions are what aspects of images and videos language can control, and how such control can be achieved through model design, training pipelines, or external visual editing tools. On the understanding side, multimodal large language models are expected to reason jointly over text and images. However, recent observations suggest an imbalance between textual and visual understanding, where models may rely heavily on language while underusing visual inputs. This dissertation studies these two challenges.

The first challenge is text controllability in visual generation and editing. Text instructions provide an intuitive way for users to express visual intent, but language and visual content belong to different modalities. Human users can interpret an instruction and infer the relevant visual changes, such as object layout, attribute faithfulness, aesthetic quality, style, video content, and camera motion. For visual generative models, achieving this form of control is substantially more difficult. The model must ground textual descriptions in concrete visual features, determine which aspects of the image or video should change, preserve irrelevant content, and produce outputs that remain visually coherent and faithful to the instruction. This challenge appears across multiple technical routes. In diffusion-based image generation and editing, the question is how to guide the generative process so that text controls the intended attributes, layouts, and compositions. In video generation, the question extends to temporal content and camera

behavior, where language must specify not only what appears in the scene but also how the camera moves through it. In tool-assisted image editing, the question becomes how to translate open-ended user requests into structured operations in professional editing software. These routes differ in architecture and execution mechanism, but they share a common problem: how to translate natural-language instructions into visual results through controllable transformations.

The second challenge is over-reliance on text in multimodal understanding. Multimodal large language models have demonstrated strong capabilities in visual question answering, reasoning, planning, and interactive tasks. However, their apparent multimodal competence can be strongly influenced by the textual information provided during training and evaluation. Many datasets contain rich textual descriptions that summarize or reveal important visual content, allowing models to solve tasks through language shortcuts rather than direct visual reasoning. As a result, when critical information is available only from the image, these models may fail to interpret spatial relations, visual structures, or diagrammatic information, even in cases that are straightforward for humans. This raises a fundamental question about whether current MLLMs truly integrate visual and textual information in a balanced way. Addressing this challenge requires diagnostic benchmarks that expose the gap between textual and visual reasoning, as well as training strategies that reduce excessive textual hints and encourage models to rely more directly on visual inputs.

The core problem addressed in this dissertation is how to use text effectively in vision-language models while maintaining strong visual grounding. On the generation side, this requires methods that improve the ability of language to control images and videos across different control targets and technical mechanisms. On the understanding side, this requires methods that diagnose and mitigate excessive dependence on text. Together, these challenges motivate a unified study of how text can serve as a powerful

control interface in visual generation, while also revealing and addressing its role in creating imbalance in multimodal understanding.

1.3 Dissertation Contributions

This dissertation makes contributions in two main directions: text-controlled visual generation and editing, and visual reasoning in multimodal understanding.

Text-Controlled Visual Generation and Editing.

- **Instruction-based image editing with professional tools.** I develop an MLLM agent for instruction-based image editing that translates natural-language requests into executable operations in professional editing software. This work explores a tool-assisted route for visual editing, where the model leverages mature editing tools rather than directly generating edited images as a black box. To train the agent, I introduce a two-stage learning framework with supervised fine-tuning and reinforcement learning, supported by a generalist reward model that evaluates editing results through multimodal reasoning.

- **Text-guided camera control in video generation.** I study how natural language can control camera motion in video generation. Instead of requiring users to specify explicit camera trajectories, this work enables users to describe both scene content and camera behavior through text. I construct synthetic training data with controllable object and camera motion, and propose a training strategy that disentangles visual appearance from camera control, enabling more flexible and realistic text-guided video generation.

- **Text-guided visual attribute control in diffusion models.** I study the in-

herent disentanglement capability of text-to-image diffusion models for image editing. This work shows that partially modifying text embeddings during denoising can change target attributes while preserving semantic content, and develops a lightweight optimization method for attribute editing without fine-tuning the diffusion model.

- **Region-aware attention control for high-fidelity text-to-image generation.** I study how to improve prompt-image fidelity in text-to-image diffusion models, especially when prompts involve multiple objects, attributes, or spatial layouts. This work introduces an attention-control method that predicts object regions and combines global prompt attention with object-specific local attention across denoising steps, reducing missing objects, mismatched attributes, and misslocated objects without fine-tuning the diffusion model.

Over-Reliance on Text in Multimodal Understanding.

- **Benchmarking visual reasoning limitations in MLLMs.** I study the imbalance between visual and textual understanding in multimodal large language models. Through spatial reasoning and planning tasks, I show that current MLLMs often struggle when essential information must be inferred from visual inputs rather than textual descriptions. This work provides diagnostic evidence that strong language reasoning does not necessarily imply strong visual reasoning.

- **Training MLLMs to rely more on visual inputs.** I explore training strategies that reduce excessive textual hints and encourage multimodal models to reason more directly from images. Using geometry reasoning as a representative setting, this work investigates how changing the structure of multimodal supervision can improve visual understanding and reduce dependence on text.

Together, these contributions provide a systematic study of the dual role of text in vision-language models. On the generation side, they improve the ability of text to control images and videos across tools, diffusion models, and camera-aware video generation. On the understanding side, they reveal and address the risks of excessive dependence on text, pointing toward multimodal models that are more visually grounded and reliable.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows.

- **Chapter 2: Tool-Assisted Text-Guided Image Editing** studies instruction-based image editing with professional editing tools. This chapter presents an MLLM agent that translates natural-language requests into executable editing operations. It further introduces a training framework based on supervised fine-tuning and reinforcement learning, where a generalist reward model provides feedback for open-ended visual edits.

- **Chapter 3: Text-Guided Camera Control in Video Generation** focuses on controlling camera motion in video generation through natural language. Instead of requiring users to specify explicit camera trajectories, this chapter studies how text can describe both scene content and camera behavior. It presents a data synthesis pipeline and a training strategy that disentangles visual appearance from camera motion, enabling more flexible text-guided video generation.

- **Chapter 4: Text-Guided Visual Attribute Control in Diffusion Models** studies the inherent disentanglement capability of text-to-image diffusion models for image editing. This chapter shows how partially modifying text embeddings

during the denoising process can change target attributes while preserving semantic content, and presents a lightweight optimization method for attribute editing without fine-tuning the diffusion model.

- **Chapter 5: Region-Aware Attention Control for High-Fidelity Text-to-Image Generation** investigates how to improve prompt-image fidelity in text-to-image diffusion models, especially when prompts involve multiple objects, attributes, or spatial layouts. This chapter presents an attention-control method that predicts object regions and combines global prompt attention with object-specific local attention across denoising steps, reducing missing objects, mismatched attributes, and mislocated objects without fine-tuning the diffusion model.

- **Chapter 6: Measuring Visual and Textual Understanding in MLLMs** examines the imbalance between visual and textual understanding in multimodal large language models. This chapter introduces diagnostic tasks for spatial reasoning and planning, showing that current MLLMs may struggle when essential information must be inferred from visual inputs rather than textual descriptions.

- **Chapter 7: Enhancing Visual Understanding in MLLMs** studies how to reduce excessive dependence on text and encourage stronger visual reasoning. Using geometry reasoning as a representative setting, this chapter explores training strategies that provide less textual information and require models to infer key relationships directly from images.

- **Chapter 8: Conclusion and Future Directions** summarizes the main findings of the dissertation and discusses future directions for building vision-language models that are more controllable, visually grounded, and reliable.

Chapter 2

Tool-Assisted Text-Guided Image Editing

2.1 Introduction

The recent progress of multimodal large language models (MLLMs) has unlocked a vast landscape of applications that bridge visual understanding and linguistic reasoning. Beyond perception and captioning, MLLMs are now capable of acting as general-purpose agents that can invoke external tools, combining the broad reasoning capabilities of language models with the precision and domain expertise of specialized systems. Such MLLM agents have demonstrated their potential in a wide range of domains, from retrieval [21, 22] and image generation [23–25] to autonomous workflows [18, 26–28], showcasing an emerging paradigm of reasoning-driven tool use.

Among these applications, one particularly creative and artistically meaningful direction is to empower MLLM agents to interact with professional image editing software, such as Adobe Lightroom and PicsArt [29, 30]. However, mastering such software demands deep technical and creative expertise, posing a barrier for both casual users and

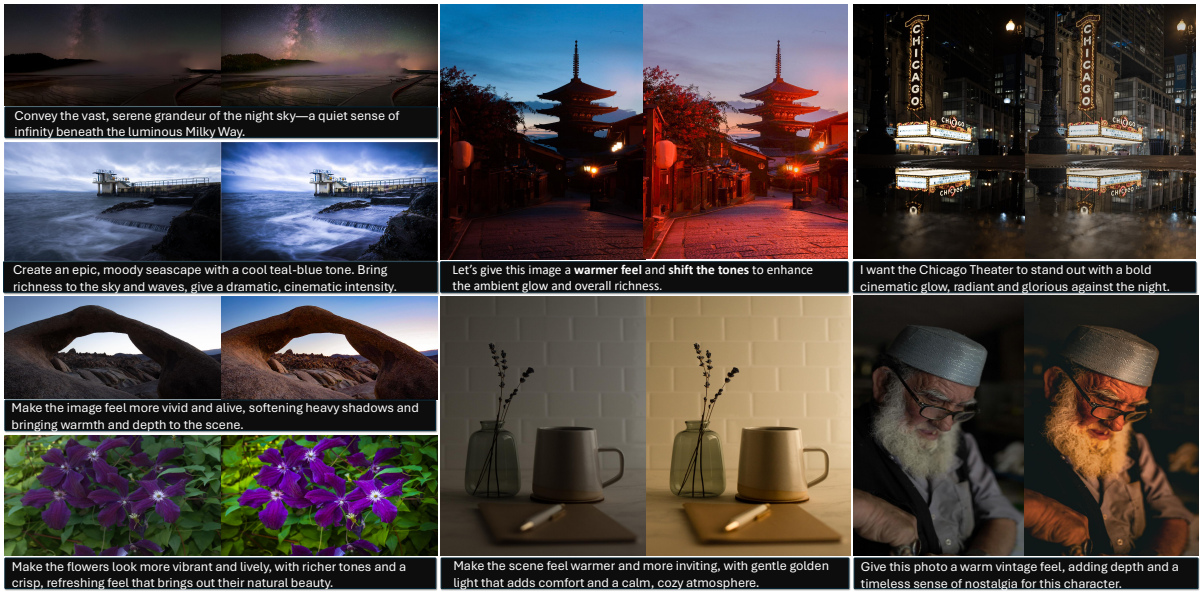


Figure 2.1: We present RetouchIQ, an MLLM agent that performs customized image retouching using Lightroom. Given a user instruction and an image input, RetouchIQ produces high-quality results across various quality enhancement and style transformation scenarios.

professional photographers who may excel at capturing images but lack the specialized knowledge for effective retouching. Existing solutions [31–33] adopt various post-training strategies to equip MLLMs with tool-use capabilities, including reinforcement learning (RL) aimed at reproducing human-edited results. However, due to the inherently subjective nature of creative photo retouching, multiple edits can be equally good, and different users may produce distinct yet valid outcomes for the same request. Consequently, existing rule-based pixel-level rewards (See Figure 2.3) become unreliable when model training is anchored to a single reference edit.

To address this challenge, we introduce RetouchIQ, an MLLM-based agent that performs instruction-based executable image editing, converting user requests into interpretable reasoning and actionable steps. RetouchIQ leverages a professional image processing platform, Lightroom, to execute fine-grained parameter adjustments that align with natural language instructions (e.g., "create an epic, moody seascape, give a

dramatic, cinematic intensity" as shown in Figure 2.1). Crucially, we introduce a generalist reward model (GRM), which is an RL fine-tuned MLLM that evaluates edited results with a series of generated metrics on a case-by-case basis. To construct pairwise (strong/weak) data for RL training of the GRM, a natural approach is to generate weak edits via inverse-edit perturbations—using predefined tools and parameter offsets [34]. However, we find that this seemingly reasonable strategy yields sub-optimal performance due to a distribution shift between these synthetic weak edits and the real weak edits produced by the policy editing model. We further introduce a new training scheme (Policy-Guided Reward Training, PGRT) designed to align the data distributions of the policy editing model and the reward critic model, thereby yielding stronger and more stable performance.

Experiments show that RetouchIQ exhibits strong capabilities in producing aesthetically consistent, instruction-aligned, and publication-quality image edits. Compared with existing agent and diffusion-based editing systems, our framework offers better semantic consistency and perceptual quality, while providing explainable reasoning throughout the editing process. Our contributions are threefold:

- We introduce RetouchIQ, the first framework to address the inherently subjective nature of image retouching by introducing a *generalist reward model* that provides flexible, context-aware supervision beyond conventional verifiable rewards.
- We propose PGRT, a RL paradigm that leverages signals from the policy editing model itself to refine reward estimation and better capture nuanced perceptual differences between before- and after-edit images.
- RetouchIQ demonstrates strong performance in both semantic alignment and perceptual quality, surpassing state-of-the-art general purpose MLLMs, MLLM

agents and diffusion-based systems, and producing edits that closely approach professional-level retouching.

2.2 Background and Prior Work

Image Retouching Image retouching aims to enhance visual appeal by adjusting tone, color, and illumination while preserving realism and semantic consistency. Early works focused on learning to predict image editing operations to achieve aesthetically pleasing results [35–39]. While these methods provide step-wise and explainable editing procedures, these models typically have little or no capability to capture nuanced, user-specific goals, which prevents them from handling diverse textual instructions. More recently, diffusion-based models [40, 41] have been explored for image enhancement and editing according to user prompts [13, 42–44]. Although these models can interpret natural-language instructions, they often unintentionally alter the original image content and environment due to the stochastic nature of the diffusion process. To address these limitations, recent research has begun exploring multimodal LLM agents that interact with external editing tools to achieve desired retouching effects [31–33]. These studies demonstrate that MLLMs possess potential to understand user intentions and reason about proper editing steps, even in a zero-shot setting [45, 46].

Generalist Reward in Reinforcement Learning Precise reward signals are central to fine-tuning LLMs for reasoning capabilities. Previous approaches mainly rely on verifiable rewards [47–49], which provide rule-based scores determined by the correctness of the final answers. While such methods have proven effective in objective domains such as mathematics and programming [50–52], recent studies show that they struggle in more complex and subjective tasks [53], such as the agentic scenario. One way to

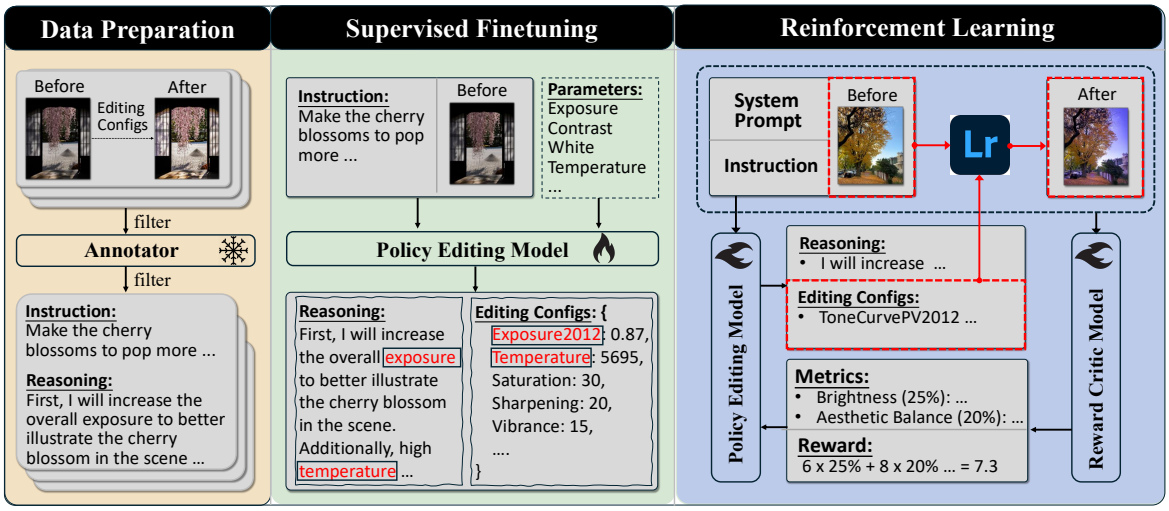


Figure 2.2: Overview of RetouchIQ. **Left:** We annotate the user instruction and reasoning for training data. Generated data are filtered to ensure quality. **Middle:** The supervised fine-tuning stage. Based on the user instruction (*e.g.*, pop more), the policy model needs to reason the correct parameters (*e.g.*, `exposure`) and change them accordingly. **Right:** The reinforcement learning stage. We leverage a generalist reward model to propose metrics and provide scalar reward guidance for policy model. Details of the reward model is introduced in Sec 2.4.

overcome this is through a *generalist reward model* [53, 54], where an LLM reward model is trained to score the answers. In this chapter, we extend this concept by training an MLLM-based reward model that assesses output images through a series of dynamically generated evaluation metrics.

2.3 RetouchIQ

In this section, we introduce RetouchIQ, an MLLM agent that performs instruction-based image retouching leveraging Lightroom. We first introduce the overall setup in Sec. 2.3.1. Then, we introduce the training pipeline, including data preprocessing (Sec. 2.3.2), supervised fine-tuning (Sec. 2.3.3), and reinforcement learning (Sec. 2.3.4).

2.3.1 Overview

The RetouchIQ pipeline is illustrated in Figure 2.2. RetouchIQ includes a policy model that translates a user's natural-language instruction into two outputs:

- A **reasoning trace**, describing the semantic interpretation and aesthetic intent of the request, and
- A sequence of **parameterized editing tool-use operations** that can be executed in Lightroom runtime environment (e.g., {`exposure=+0.9`; `contrast=-30`}).

To achieve this, we adopt a two-stage training strategy. In the first stage, supervised fine-tuning (SFT), the model learns to simulate *gold* reasoning processes and corresponding editing operations from real user edits. In the second stage, reinforcement learning (RL), the model self-explores diverse reasoning paths and editing plans, gradually discovering better editing strategies under different input conditions. However, collecting a suitable training dataset that includes images, user editing intentions, and detailed editing steps required for SFT and RL remains challenging. To address this, we design a data collection pipeline to curate and preprocess high-quality data for subsequent training.

2.3.2 Data Preparation

An ideal training datum should contain the following elements: a before-and-after image pair (I_0, I), a user editing goal g expressed in natural language, a reasoning process q describing how to plan the optimal editing strategy, and an editing step sequence e consisting of parameter configurations within Lightroom.

We present the data preparation pipeline in the left panel of Figure 2.2. We start by collecting initial data from user editing histories in Lightroom, where the before-and-after image pairs (I_0, I) along with the corresponding editing sequences e are recorded. Note

that, unlike existing work [32] that synthesizes post-edit images by applying predefined editing presets, our before–after image pairs are collected from real human users, making them naturally closer to real-world editing scenarios. However, two essential elements are missing from the source data: ❶ the specific editing goal g that the user had when modifying the image, and ❷ the reasoning process q that guided the user's use of Lightroom tools during editing.

As such, we seek to annotate g and q by leveraging an MLLM-based annotator. Specifically, the annotator takes as input the triplet (I_0, I, e) and first infers the user's editing intention g based on the differences between the images and the applied editing steps. It is then prompted to generate a reasoning process q, simulating how an MLLM agent would reason and produce the final editing sequence e accordingly. To ensure the quality of the generated instructions and reasoning processes, we apply a systematic filtering procedure to remove image pairs with unclear editing intentions or inconsistencies between the generated intentions and reasoning. The MLLM annotator remains fixed throughout the entire process.

2.3.3 Supervised Fine-Tuning Stage

With the prepared training data, we now perform supervised fine-tuning of the MLLM policy model in RetouchIQ. We initialize the policy model from a pretrained MLLM backbone and fine-tune it using the instruction–reasoning–edit corpus. Given an input image I_0 and user instruction g, the model is trained to generate both a textual reasoning trace q and structured editing steps e. The supervised fine-tuning objective is an autoregressive loss that minimizes the negative log-likelihood over the target token sequence:

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{t} \log p_\theta(y_t \mid y_{<t},\, I_0, g), \tag{2.1}$$

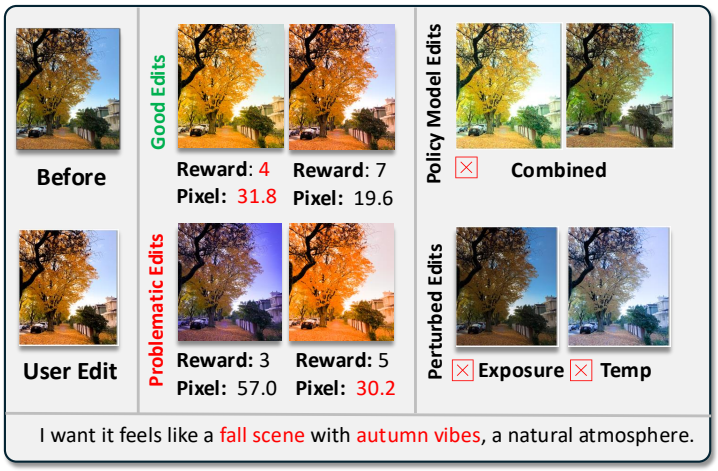


Figure 2.3: Problematic rewards in image retouching tasks. Given a before–after image pair (**left**), ❶ **verifiable rewards** (**middle**) rely on metrics between the edited image and ground truth, such as pixel differences. However, since multiple valid edits can satisfy user intent, these rewards become imprecise. ❷ The **reward model**'s precision strongly depends on its **training data distribution** (**right**). When trained to distinguish good user edits from randomly perturbed images, it may later struggle to assess results from the policy model that produces combined, complex edits.

where θ denotes the model parameters, y_t is the t-th output token, and $y_{<t}$ represents all previously generated tokens. The output sequence contains both the natural-language reasoning q and the editing operations e in a structured format. This stage enables the model to learn semantically grounded mappings between linguistic aesthetic intent and executable editing parameters, providing a strong initialization for downstream reinforcement learning.

2.3.4 Reinforcement Learning Stage

While supervised fine-tuning aligns the policy editing model with demonstration data, it cannot guarantee optimal aesthetic alignment with user intent across diverse images and instructions. To further improve the model's ability to generalize and adapt, we refine the policy through RL, allowing it to explore various reasoning paths and editing strategies beyond the demonstration data.

A conventional practice in RL is to develop a verifiable reward, which gives scores based on pre-defined rules such as similarities to a ground truth reference image [32]. However, this strategy faces a fundamental challenge in image editing, where *the inherently subjective nature of aesthetics makes it impossible to define a single verifiable ground truth.* As illustrated in Fig. 2.3, a given instruction (e.g., "make it feel like a fall scene with autumn vibes") can correspond to multiple acceptable results, each expressing the intent differently through tone, warmth, or color balance. In this setting, conventional reward definitions based on pixel-wise or reference-based comparisons fail to capture semantic quality or artistic preference. Such metrics are suitable for deterministic reconstruction tasks but are inadequate for open-ended, instruction-driven retouching, where subjective aesthetic judgments play a central role.

We therefore further refine the policy via RL using the *reward model* r_ϕ, which evaluates how well an edited result satisfies the given instruction. For a policy $\pi_\theta(q, s \mid I_0, g)$ that predicts editing actions given the image and instruction, the objective is to maximize the expected reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{q,s\sim\pi_\theta}\left[r_\phi(g,\, I_0, \text{Execute}(I_0, e)) + r_{\text{format}}(q, s)\right], \tag{2.2}$$

where Execute$(\cdot)$ denotes applying the predicted edits in the Lightroom engine to obtain the resulting image, and r_{format} is a format reward [47]. During training, the policy samples candidate edits, receives feedback, and updates its parameters via gradient ascent on $\mathcal{J}(\theta)$, progressively improving instruction alignment and aesthetic quality.

As such, the success of this RL stage heavily depends on the quality of the reward model. Ideally, the reward model should provide a scalar signal that accurately reflects the quality of the generated edit. However, the criteria for a successful edit vary across different input images and user instructions. To flexibly adapt to diverse scenarios and

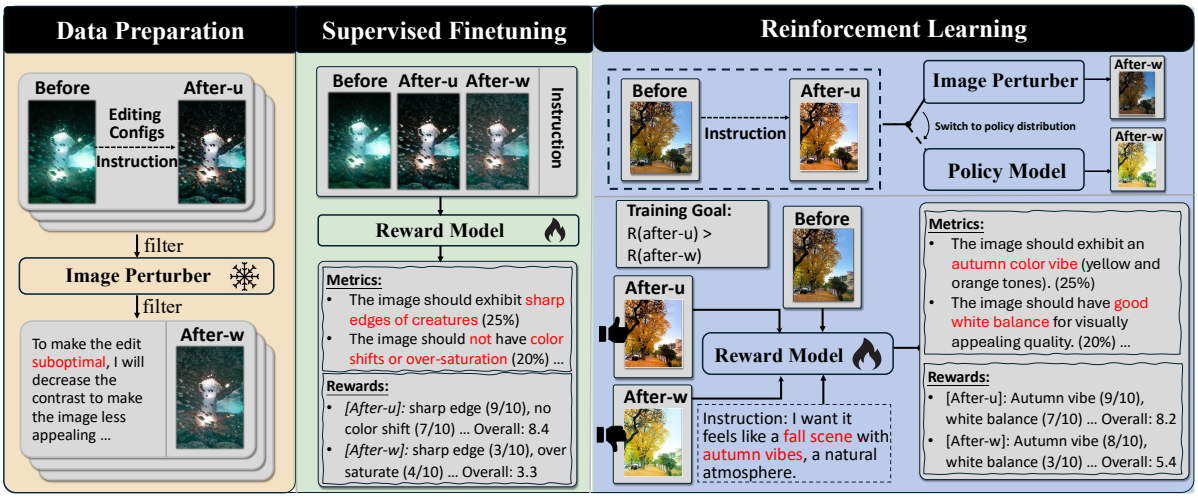


Figure 2.4: Overview of generalist reward model. **Left**: Given a before-edited image and a user-edited after image `After-u` ("u" for "user"), we perturb the editing process to obtain a suboptimal image, `After-w` ("w" for "weak"). **Middle**: In the SFT stage,the reward model first learns to generate metrics and then produces scalar rewards for the two input images. **Right**: In the RL stage, we introduce policy-guided reward training (PGRT), where the suboptimal image `After-w` is provided by the policy model rather than generated through perturbation. The goal of the RL stage is to assign a higher scalar reward to the user-edited (strong) image `After-u`.

deliver precise reward guidance, we propose a *generalist reward model* that first generates a set of metrics describing the key aspects of a successful edit and then scores the output image accordingly. The following section provides details of the generalist reward model.

2.4 Generalist Reward Model

In this section, we present the Generalist Reward Model (GRM) used in RetouchIQ, describing the preparation of its training data and the procedures for model optimization.

2.4.1 Overview

The overall GRM preparation pipeline is shown in Figure 2.4. The proposed MLLM-based reward model evaluates image retouching quality by taking a before–after image

pair $(I_0, \text{Execute}(I_0, e))$ and the corresponding editing instruction g. Then, it produces two outputs sequentially:

- A self-generated set of **metrics** that describe the key characteristics a successful edit should exhibit, and
- A scalar **reward** value reflecting the overall quality of the retouching according to these metrics.

Similar to the policy model, we first prepare dedicated training data for the GRM. We then adopt SFT and RL to ensure that the GRM provides accurate and consistent guidance across diverse retouching scenarios.

2.4.2 Data Preparation

When preparing the training data, the key challenge lies in constructing accurate and reliable *gold* signals. Our primary objective is to ensure the correct ordering in supervision: a better retouching result should receive a higher reward than a suboptimal one under the same instruction.

To achieve this, we construct a paired comparison dataset by augmenting the training corpus with *strong* and *weak* edits for each original image, as shown in the left panel of Figure 2.4. Starting from the initial data pool containing before-after pairs (I_0, I), we treat the reference after image I as a strong edit corresponding to the annotated instruction g. To obtain a weak edit, we intentionally *perturb* the editing configuration s to produce an alternative version that omits or misadjusts several key parameters during the editing process, resulting in a suboptimal outcome I_w. This perturbation is guided by a frozen MLLM to maintain edits that remain plausible yet perceptually inferior. We further apply filtering mechanisms to ensure that each generated I_w is indeed weaker than I, providing reliable ranking consistency for reward learning.

Finally, we annotate each (I, I_w) pair using an MLLM. Since the ground-truth ordering between the strong and weak edits is known, we retain only annotations that correctly assign a higher score to I, obtaining annotated metrics m and scores for the strong and weak response (s, s_w) correspondingly.

2.4.3 Training

Supervised Fine-Tuning Stage We first perform supervised fine-tuning so the GRM learns to produce metrics and corresponding scores consistent with annotations. The training examples contain input image I_0, editing instruction g, two after-edit images I and I_w, and annotated metric m and scores (m, s, s_w). The training objective is an autoregressive loss that fits the annotation:

$$\mathcal{L}_{\text{SFT}}^{\text{reward}} = -\sum_{t} \log p_\phi(y_t \,|\, y_{<t},\, I_0, I, I_w, g). \tag{2.3}$$

This stage trains the model to explain aesthetic metrics in natural language and to produce scalar scores accordingly.

Policy-Guided Reward Training (PGRT) in RL The previous SFT stage relies on perturbed sub-optimal images, which introduces intrinsic bias toward a limited set of edit categories. As illustrated in Figure 2.3, we observe that many perturbations involve single adjustments to `exposure` and `temperature`, while the results generated by policy model often involve combined and complex edits. This reveals a training distribution shift, which may cause inferior performance of the reward model when predicting real outputs from policy models.

Our goal is to align the reward model with the policy distribution during training. To this end, we propose *policy-guided reward training* (PGRT), which trains the

reward model on data consistent with the policy's editing distribution. As shown in Figure 2.4 (right), the RL stage replaces perturbed images with policy-generated results, shifting I_w from the perturbed to the policy-generated distribution. Given paired examples (I_0, I, I_w), the reward function is defined as

$$\mathcal{J}(\phi) = \mathbb{E}_{m,r,r_w \sim \pi_\phi} \left[\mathbb{I}[r > r_w] + r_{\text{format}}(m, r, r_w) \right], \tag{2.4}$$

where $\mathbb{I}[\cdot]$ is a 0-1 indicator, and r_{format} is a format reward ensuring that both metrics and scores are properly generated. Essentially, this reward function penalizes the model when it incorrectly assigns a higher score to a weaker edit.

PGRT relies on the outputs of the policy model. Therefore, when training RetouchIQ, we adopt an alternating training strategy in which the policy model and the reward model are optimized in turn to progressively enhance each other's performance. It is worth noting that when training the reward model for the first time, the RL stage uses only perturbed edits, since the policy model is not yet trained.

2.5 Experiments

In this section, we evaluate RetouchIQ under various retouching settings and compare it with state-of-the-art methods. We focus on the following questions:

- Can RetouchIQ consistently provide high-quality image retouching results?
- How does the generalist reward model overcome the challenges in conventional verifiable reward?
- How does the policy-guided reward training improves the reward model and policy model in RetouchIQ?

2.5.1 Experiment Settings

Datasets We curate large-scale, high-quality datasets for training both the policy model and the reward model. For the policy model, the final training set contains 190K image–instruction pairs covering a wide range of *global* and *local* retouching tasks. Each input image is associated with three instruction variants that differ in length and complexity, providing diversity during training. For the reward model, we construct a dataset of 10K perturbed samples, where each datum consists of an original image, its ground-truth editing, and a corresponding sub-optimal version. During alternating training, an additional 5K samples are generated to further fine-tune the reward model under the RL objective.

Implementation The policy and reward models are built upon `Qwen2.5-VL-7B` [55]. For the MLLM annotator and perturber, we employ `GLM-4.5V` [48]. During training, we first fine-tune the reward model using the perturbed dataset. We then perform alternating fine-tuning of the policy and reward models, as described in Sec. 2.3 and 2.4.

Benchmarks To the best of our knowledge, there is no existing public benchmark for evaluating image retouching quality using professional editing software such as Lightroom. Therefore, we introduce RetouchEval, a dedicated benchmark designed to assess retouching quality. RetouchEval consists of 300 instruction–image pairs curated from real user editing histories. The dataset is divided into three splits: *quality enhancement*, which focuses on enhancing the overall visual quality of the input image; *style transformation*, which aims to alter the artistic or tonal style of the image; and *local retouching*, which targets localized adjustments rather than global edits. In addition, we also evaluate on MIT-Adobe5K [56], a public benchmark for general image enhancement in Lightroom. It is worth noting that, unlike our setting, MIT-Adobe5K does not include

explicit user intentions; as its primary objective is to improve perceptual quality rather than to achieve a customized editing style. Following previous work [31], we conduct evaluations on 400 randomly selected test images.

Baselines We compare RetouchIQ with three categories of state-of-the-art baselines: ❶ **General-purpose MLLMs**: We carefully design editing instructions and prompt leading MLLMs, including `Gemini-2.5` [57] and `GPT-5` [17], to evaluate their zero-shot ability to interpret user instructions and generate valid editing plans. ❷ **MLLM Agents**: `JarvisArt` [32] serves as a representative state-of-the-art agent that leverages Lightroom for image editing. We also include `MonetGPT` [31], a strong baseline focusing on image quality enhancement within Lightroom. ❸ **Diffusion-based Models**: We further evaluate how advanced diffusion-based models, `flux-pro-1.1` [58], respond to editing instructions and perform corresponding modifications. Note that Google's NanaBanana and Adobe Firefly are popular image editing solutions but are closed-source. Since their underlying architectures (e.g., pure diffusion or hybrid systems) remain undisclosed, we instead compare against the leading open-source diffusion model, `flux-pro-1.1`.

Metrics Following prior work [32], we adopt the following metrics: ❶ L1 and L2 differences between the output and the ground-truth images; ❷ Perceptual quality and semantic consistency scores evaluated by GLM-4.5V [48]. For the MIT-Adobe5K dataset, we report PSNR, LPIPS, and SSIM, following the original benchmark setting.

2.5.2 Qualitative Comparisons

Figure 2.5 presents side-by-side comparisons between RetouchIQ and representative baselines across three scenarios: quality improving, style changing, and local retouching. We observe that each baseline exhibits distinct deficiencies. First, general-purpose

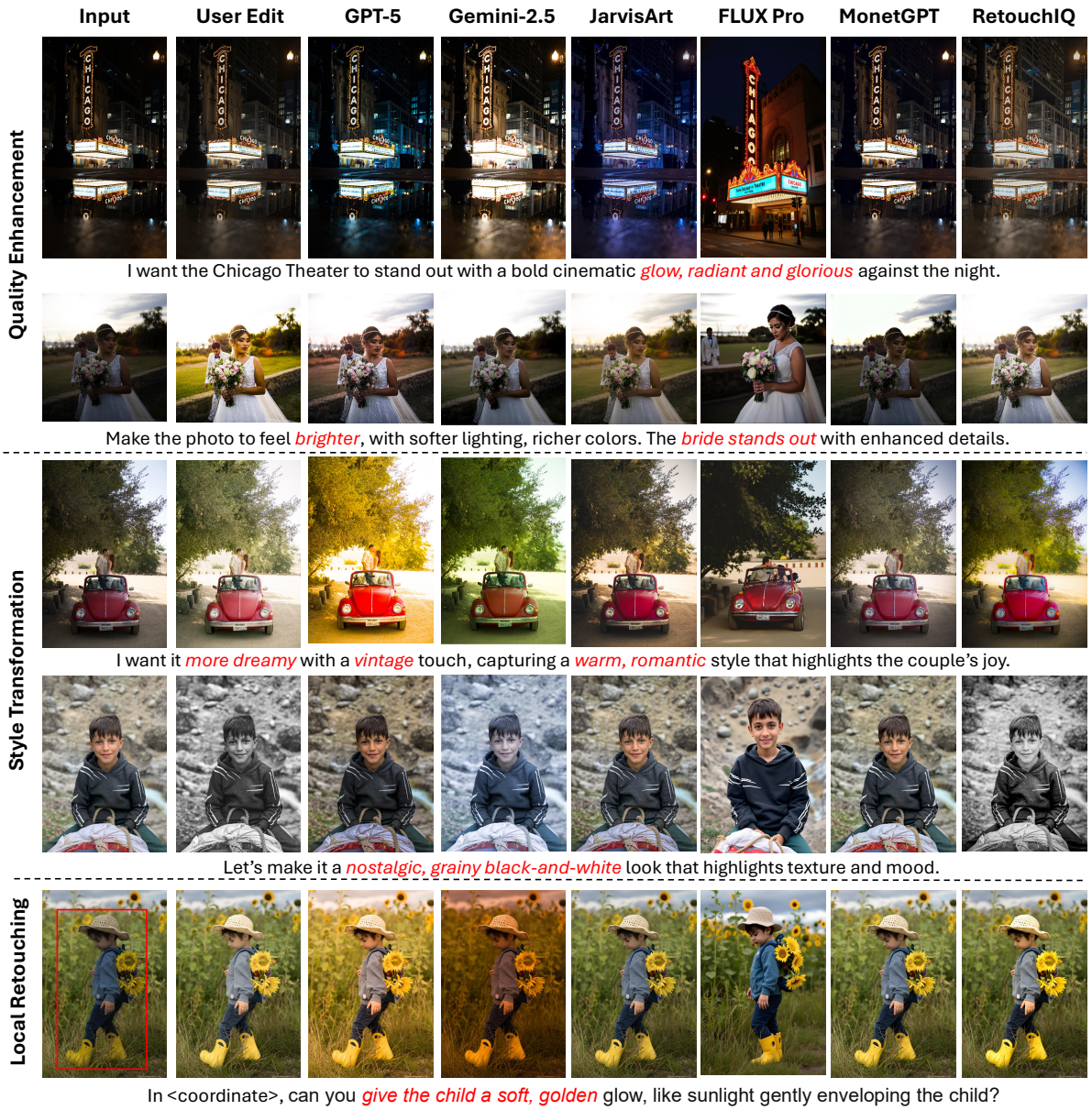


Figure 2.5: Qualitative results across diverse image retouching scenarios, including quality enhancement (top), style transformation (middle), and local retouching (bottom). For each example, the input image and the corresponding user-edited result are shown on the left.

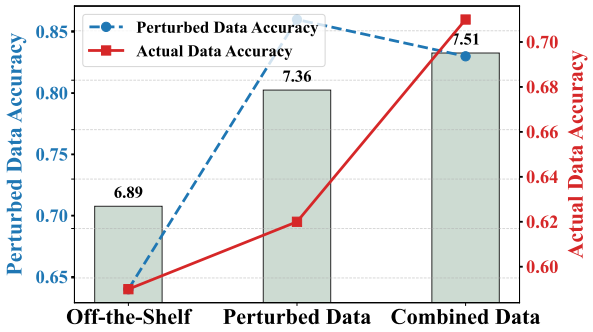


Figure 2.6: Comparison of reward model and policy model performance under different reward model configurations. The lines show the accuracies of the reward model, while the bars indicate the scores of the corresponding policy model.

MLLMs such as `GPT-5` tend to over-edit images, as shown in the 3rd and 5th rows. This is likely because these models struggle to infer coherent and natural parameter configurations in a zero-shot manner. Second, the diffusion-based method `Flux Pro` fails to preserve the original image structure, leading to noticeable identity and environmental distortions in all examples. Among MLLM agents, `MonetGPT` struggles with style changing tasks, while `JarvisArt` fails on customized requests, such as incorrectly adjusting the temperature in the first row and missing the black-and-white conversions in the fourth row. In contrast, RetouchIQ consistently produces results that are more semantically aligned and professionally styled, particularly in challenging scenarios such as nighttime balancing and tone harmonization.

2.5.3 Quantitative Evaluation

Next, we quantitatively evaluate performance on RetouchEval and MIT-Adobe5K. The results are shown in Table 2.1 and 2.2. We observe that RetouchIQ achieves superior results across various scenarios, outperforming diffusion-based and MLLM-agent baselines by large margins. Similar gains are observed on MIT-Adobe5K, demonstrating that RetouchIQ generalizes beyond the curated instruction dataset and transfers well to

Table 2.1: Quantitative comparisons on the RetouchEval benchmark. We consider three categories of baselines: general-purpose MLLMs (GPT-5, Gemini-2.5); MLLM agents (MonetGPT, JarvisArt); and diffusion-based methods (Flux-Pro). Meanwhile, we report the performance of RetouchIQ under both the SFT (RetouchIQ-SFT) and RL (RetouchIQ-GRM) stages. We also include results from a variant trained with rule-based reward (RetouchIQ-Rule). The best results are shown in **bold**, and the second-best results are underlined.

	Quality Improving					Style Changing					Local Retouching				
	L1 ↓	L2 ↓	SC ↑	PQ ↑	O ↑	L1 ↓	L2 ↓	SC ↑	PQ ↑	O ↑	L1 ↓	L2 ↓	SC ↑	PQ ↑	O ↑
Flux-Pro [58]	64.97	98.14	6.12	6.09	6.10	68.20	70.01	6.48	6.36	6.42	67.93	82.86	6.19	6.26	6.22
GPT-5 [17]	36.21	47.53	6.74	6.51	6.62	39.17	50.36	6.89	6.76	6.82	37.20	47.56	6.07	6.89	6.47
Gemini-2.5 [57]	39.58	51.60	6.56	6.72	6.64	41.54	53.01	5.82	6.28	6.05	36.11	46.72	6.01	6.63	6.31
MonetGPT [31]	32.05	**44.98**	6.52	7.05	6.78	41.03	52.75	5.87	6.03	5.95	30.08	44.98	6.27	6.70	6.48
JarvisArt [32]	32.17	46.23	7.22	6.59	6.90	**33.90**	48.23	**7.39**	6.88	7.13	38.19	49.75	**6.45**	6.39	6.42
RetouchIQ-SFT	35.03	46.63	6.71	6.67	6.69	38.80	51.55	7.04	6.96	7.00	**26.41**	**42.10**	5.98	6.84	6.40
RetouchIQ-Rule	32.65	46.47	7.14	6.61	6.87	35.74	49.01	7.17	6.91	7.04	28.22	43.89	6.20	6.83	6.51
RetouchIQ-GRM	**31.41**	44.99	**7.57**	**7.48**	**7.51**	34.08	**47.71**	7.29	**7.34**	**7.31**	27.03	42.73	6.39	**6.92**	**6.65**

standard aesthetic editing benchmarks. Notably, the inclusion of RL fine-tuning with the generalist reward contributes a consistent improvement across all metrics, confirming the value of multimodal reward alignment. Meanwhile, we observe that JarvisArt often achieves relatively high semantic consistency scores.

2.5.4 The Role of Reward Model and PGRT

In this section, we analyze the role of the reward model in RetouchIQ. Specifically, we examine how policy-guided reward training (PGRT) improves the reward model itself, and how this improvement, in turn, benefits the policy model training. We consider two variants of the generalist reward model: ❶ an untrained version, *i.e.*, an off-the-shelf MLLM; and ❷ a version trained only on perturbed data generated by the image perturber introduced in Sec. 2.4.

The results are in Figure 2.6. We first report the accuracy of the reward model (line plot). Here, accuracy measures whether the reward model correctly assigns lower rewards to sub-optimal images either from perturbed synthesis data (blue line) or from actual policy-generated data (red line). As shown in the figure, PGRT shifts the reward

Table 2.2: Quantitative results on the MIT-Adobe5K benchmark.

	SSIM	LPIPS	PSNR
GPT-5 [17]	0.72	0.26	20.82
MonetGPT [31]	0.82	0.17	23.10
JarvisArt [32]	0.76	0.23	21.03
RetouchIQ-SFT	0.84	0.20	22.37
RetouchIQ-GRM	**0.86**	**0.16**	**23.14**

model's distribution toward actual policy-generated data, achieving the highest accuracy on that set. Next, we use these reward models to train policy models and report their performance on the RetouchEval benchmark, illustrated by the bar plot. We observe that PGRT yields the best overall performance, indicating that policy-guided reward training effectively produces a reward model that better supports the policy model's learning process.

2.6 Conclusion

This chapter introduced RetouchIQ, an instruction-based MLLM agent that performs executable image editing via professional software interfaces. By converting natural-language instructions into parameterized edits, it enables controllable, transparent manipulation beyond black-box diffusion methods. A generalist reward model assesses editing quality, providing human-like feedback for reinforcement learning. This enhances instruction consistency and demonstrates the potential of reasoning-based policies with multimodal rewards for intelligent visual editing.

Chapter 3

Text-Guided Camera Control in Video Generation

3.1 Introduction

In creative video generation, camera motion is pivotal for conveying intent, enhancing expressivity, and adding artistic value. As a result, recent work in video generation has focused heavily on equipping text-to-video models with camera control. Camera control in video generation is commonly approached via two paradigms: *text-based control,* where motion is described directly in the input prompt [59–61], and *trajectory-based control*, where explicit 3D motion trajectories are provided as additional conditioning [62–64]. With sufficient training data labeled with camera motion, both paradigms can effectively reproduce similar motion patterns in generated videos.

However, truly creative video generation demands more than just replicating conventional camera techniques like panning or dollying. It requires inventing stylized, intricate motions (as exemplified by the dramatic impact of the *Dolly Zoom* in Hitchcock's *Vertigo*), or crafting scene-specific movements tailored to expressive content (e.g., tracking



Figure 3.1: State-of-the-art method [63] fails to generate unconventional camera motions.

an unconventional car race). In such cases, collecting enough training data that embodies these avant-garde or bespoke camera motions is infeasible.

Unfortunately, without enough training data support, neither the text-based nor trajectory-based camera control can generalize well to unseen camera motions. For example, Figure 3.1 illustrates results from state-of-the-art baseline method [63]: while it can reproduce conventional motions such as a forward push (1^{st} row), it fails on more complex, expressive ones (2^{nd} row). In this case, the intended motion was *the camera pans left and right, seeking a sunflower in a glass vase, then locking focus and pushing in upon finding it.* As shown, the generated videos fail to follow the delicate control of the camera panning left/right process, resulting in videos with large perturbations and losing focus.

In short, there is a fundamental paradox between the data-intensive nature of training generative AI models and the creative nature of camera control. So our question is: Can we enable learning out-of-distribution camera motions without real training data?

In this chapter, we explore an alternative solution: instead of collecting real videos with uncommon camera motions, we *generate synthetic videos* with the intended motions as training data for generative models. However, while synthetic videos can cover arbitrary motions, they often exhibit virtual styles that diverge significantly from realistic ones. Directly training models on such videos would seriously degrade generation



Figure 3.2: VividCam learns diverse unconventional camera motions from synthetic videos.

quality. Of course, such drawbacks could potentially be alleviated by generating super-high-quality, real-like synthetic videos [65], but this would incur tremendous manual efforts and professional expertise to prepare these videos.

Essentially, the problem boils down to a *disentanglement problem*, *i.e.,* separating appearance and camera motion information in synthetic videos and guiding models to learn only the latter. To this end, this chapter presents **ViVidCam**, which uses synthetic **Vi**rtual **Vid**eos to fine-tune models for producing correct **Cam**era motions. ViVidCam focuses on disentanglement mechanisms. First, inspired by AnimateDiff [5], we adopt a dual-adaptation training scheme: we first learn the appearance of synthetic videos through a LoRA and then learn the camera motion; at inference, the appearance LoRA is discarded so the outputs no longer carry undesirable virtual styles. However, this technique has only been used to bridge minor appearance gaps, and is insufficient to resolve

the drastic appearance differences between realistic and virtual videos. To further mitigate virtual appearance, ViVidCam employs a training recipe with two complementary components: (i) *Data*: we synthesize two sets of videos for training, one without camera motions (to train the appearance LoRA) and one with motions (to train the motion module); and (ii) *Training signals*: we introduce an optical-flow based loss for the motion module, providing appearance-invariant supervision that stabilizes motion learning and strengthens disentanglement. Finally, we use special text prompts to *anchor* virtual appearance, enabling the model to better distinguish virtual from realistic styles.

We find that even when the generated videos are of very low visual quality, comprising only basic geometries rendered in low-poly 3D scenes (top row in Fig. 3.2), ViVidCam can still effectively disentangle the synthetic appearance from motion and generate realistic videos with the camera motion learned from the virtual videos (bottom four rows in Fig. 3.2). Our experiments demonstrate that models trained with ViVidCam can handle a wide variety of complex, compound camera motions while maintaining realistic visual quality comparable to models trained on real footage.

In summary, our main contributions are as follows:

1. We introduce ViVidCam, a novel framework for generating realistic videos with diverse camera motions by leveraging synthetic data efficiently rendered from engines like Unity.

2. Despite the significant domain gap, ViVidCam effectively mitigates artifacts present in low-quality synthetic videos and focuses on learning their complex camera motions.

3. We demonstrate that our framework can synthesize a wide range of camera motions with precise, consistent control and high visual quality.

3.2 Background and Prior Work

Video Generative Models Early explorations in video generation tasks focused on GANs [66,67]. Recently, leveraging advancements in diffusion models [40,41,68,69], video generative models have rapidly evolved. Early diffusion-based video generative models originated from adapting existing image generative models by incorporating additional motion modules [5,70–73]. More recently, many end-to-end video generation models have achieved superior results in terms of video quality, resolution, and duration [4, 74–76]. For conditional video generation, various works have explored using text or images to guide the content in generated videos. Rapid progress in video generation has led to several groundbreaking works [6, 61, 77, 78].

Camera Control in Video Generation Recently, a line of research has focused on enhancing the controllability of video generative models [79–81]. An important aspect is managing camera motion [82–85]. Early works learned simple, fixed movements (e.g., zooming, panning) from reference videos [5, 74], while later methods conditioned generation on input trajectories [86, 87], representing cameras through camera matrices [88] or Plücker embedding [62, 63]. These approaches, however, depend on large annotated datasets, which are scarce and offer only limited motion diversity. Recent works [64, 89, 90] have devoted significant effort to constructing and curating large-scale realistic video datasets with camera trajectory annotations.

Improving Video Generation Models Using Synthetic Data Training on realistic video datasets often faces limitations, such as the absence of camera motion annotations and the limited diversity of motion patterns [62,88]. As such, several studies turned to synthetic datasets, many of which focus on multi-view generation [91] and human animation [92–94]. For camera motion editing, recent work has synthesized new training

videos by modifying camera trajectories in existing sequences [95], though these efforts are generally restricted to simple motions. For camera control in video generation, recent approaches generate training videos with explicit camera trajectories in virtual scenes rendered by 3D engines [65, 96]. However, these methods typically demand substantial manual effort to generate complex and diverse scenes and objects. In contrast, our work leverages simple, low-poly 3D environments, enabling diverse camera motions without reliance on large-scale annotation or labor-intensive scene design.

3.3 Preliminaries

We first provide a brief overview of text-to-video diffusion models, which serve as the base model of our work. To train a diffusion model, we first generate a set of corrupted videos, denoted as $\boldsymbol{X}_{1:T}$, by adding progressively increasing Gaussian noises $\boldsymbol{\epsilon}_{1:T}$, to the clean video, $\boldsymbol{X}_0$. The diffusion model then learns to predict the additive noise and denoise noisy videos into cleaner videos. We focus on two types of diffusion models with slightly different training objectives.

Text-Based Control Only Standard text-to-video diffusion models condition the denoising process only on a text input, denoted as $\boldsymbol{c}$. The training loss can be written as

$$\mathcal{L}(\boldsymbol{\theta}_d) = \mathbb{E}_{\boldsymbol{X}_0,t,\boldsymbol{\epsilon}_t}[\boldsymbol{\epsilon}_t - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}_d}(\boldsymbol{X}_t, \boldsymbol{c}, t)], \tag{3.1}$$

where $\boldsymbol{\theta}_d$ denotes the parameters of the noise predictor. During inference, clean videos are progressively denoised from pure-noise videos using the trained noise predictor conditional on text $\boldsymbol{c}$. We denote the generation process as $\boldsymbol{g}_{\boldsymbol{\theta}_d}(\boldsymbol{c})$.

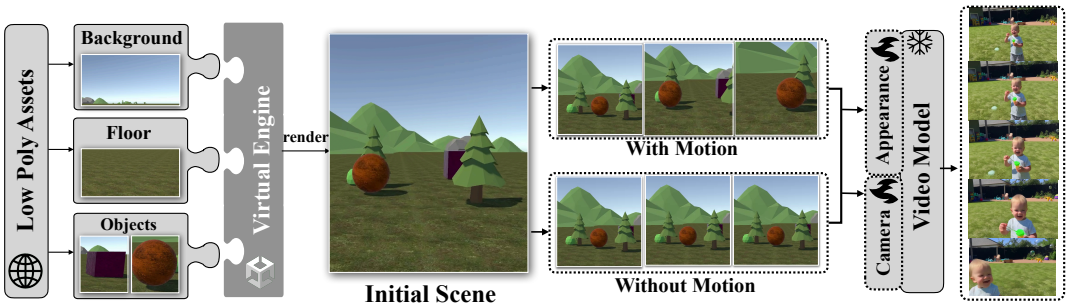


Figure 3.3: VividCam. We first render initial scene using publicly available assets. Then, we render videos with and without camera motion. These videos are used to train the camera and appearance modules, generating videos with desired camera motions. Details of training are in Sec. 3.4.3.

Trajectory-Based Control Recent works control camera poses using trajectories represented as Plücker embeddings $\boldsymbol{p}$, which are encoded via $E_{\boldsymbol{\theta}_e}(\boldsymbol{p})$ with parameters $\boldsymbol{\theta}_e$. The training loss becomes

$$\mathcal{L}(\boldsymbol{\theta}_d, \boldsymbol{\theta}_e) = \mathbb{E}_{\boldsymbol{X}_0, t, \boldsymbol{\epsilon}_t}[\boldsymbol{\epsilon}_t - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}_d}(\boldsymbol{X}_t, \boldsymbol{c}, E_{\boldsymbol{\theta}_e}(\boldsymbol{p}), t)]. \tag{3.2}$$

The inference generation process becomes $\boldsymbol{g}_{\boldsymbol{\theta}_d}(\boldsymbol{c}, E_{\boldsymbol{\theta}_e}(\boldsymbol{p}))$.

3.4 VividCam

In this section, we first formulate the research problem of synthesizing unconventional camera motions. Then, we introduce **VividCam**, a framework to generate them leveraging completely virtual synthetic videos. The overall pipeline is shown in Fig. 3.3.

3.4.1 Problem Formulation

We aim to fine-tune pre-trained text-to-video diffusion models to follow certain camera motions, which can be unconventional so the pre-trained models do not generalize well on them.

Specifically, we consider two camera control paradigms. For **text-based control**, the camera motion instructions are specified as additional input prompts, denoted as $\boldsymbol{c}_m$, such as *'The camera pans around, searching for a bird.'* We fine-tune a *text-only diffusion model*, parameterized as $\boldsymbol{\theta}_d$, to integrate the additional camera instructions, so the generation process becomes $\boldsymbol{g}_{\boldsymbol{\theta}_d+\Delta\boldsymbol{\theta}_{cd}}(\boldsymbol{c}_m \oplus \boldsymbol{c})$, where $\oplus$ denotes text concatenation, and $\Delta\boldsymbol{\theta}_{cd}$ denotes the fine-tuning weight difference.

For **trajectory-based control**, the camera motions are out-of-distribution camera trajectories $\boldsymbol{p}$. We fine-tune only the *trajectory encoder* of a text-to-video diffusion model, since the encoder already provides basic camera control abilities [63]. The adapted generation process becomes $\boldsymbol{g}_{\boldsymbol{\theta}_d}(\boldsymbol{c}, E_{\boldsymbol{\theta}_e+\Delta\boldsymbol{\theta}_{ce}}(\boldsymbol{p}))$, where $\Delta\boldsymbol{\theta}_{ce}$ are fine-tuning weight difference (only the trajectory encoder weights are updated; the diffusion model weights are kept frozen).

Since the camera motions are unconventional, realistic video datasets often lack sufficient examples for training. We propose to leverage synthetic videos rendered by a physics engine, which allows for arbitrary and diverse camera trajectories. As such, we consider two research questions:

- What types of synthetic videos are most effective for camera motion learning?
- How to ensure the model learns camera motion independently of video appearance?

To answer these questions, we first detail the process for rendering training videos in Sec. 3.4.2. Then, wepresent our pipeline for disentangling camera motion from artificial appearances in Sec. 3.4.3.

3.4.2 Render Training Videos

Our first step is to prepare synthetic videos for camera motion training. Leveraging the rendering engine, we are able to generate arbitrary camera motions within a virtual

Table 3.1: Summary of camera motion categories. We classify them into three categories based on their complexity. These motions are derived from realistic applications in everyday videos and filmmaking.

Category		**Examples**	**Videos/Filmmaking Applications**
Simple		Push in, Tilt up	❒ Emphasize an object
Composed		Push in → Truck left	❒ Emphasize, then reveal surroundings
Complex	Expressive	Seek object (pan/tilt search)	❒ Simulate searching for a target
		Switch focus between objects	❒ Highlight relational dynamics
		Orbit shot	❒ Showcase all sides of objects
		Handheld shake	❒ Simulate instability or realism
	Stylized	Dolly zoom	❒ Create dramatic perspective shift
		Explosive shake	❒ Convey impact or chaos
		Camera rotation (90/180°)	❒ Disorient viewer, mark transition

scene. Note that while it is possible to include a variety of synthetic objects in the scene, we focus on constructing the scene with **minimal** numbers and categories of objects to reduce human efforts. Below, we describe the key details of the rendering process.

The rendered scene All synthetic videos are rendered in a low-poly 3D scene using Unity. The scene consists of a *background*, *floor*, and *objects*. Examples of these elements are shown in Fig. 3.3. Notably, these elements are created using basic geometries. When preparing to synthesize videos, we first randomly sample a background, floor texture, and arbitrarily determine the positions of the objects. Fig. 3.3 also illustrates an example of the initial settings of the scene.

Videos with camera motions Next, we define the camera motion for the videos. Unity allows users to specify camera movement through code, enabling the simulation of arbitrary camera motions in the scene with just a few lines of code. We consider diverse simple and complex motions listed in Table 3.1. We denote these synthetic videos with camera motions as $\mathcal{X}_c$ (c stands for 'camera').

Videos without camera motions VividCam also requires a set of videos without camera motions to aid the disentanglement between appearance and camera motion. For this purpose, as shown in Fig. 3.3, we synthesize another set of training videos, denoted as $\mathcal{X}_a$ (a stands for 'appearance'), which consist of identical appearance styles but with a static camera.

3.4.3 Dual Adaptation Training Scheme

Next, we introduce the pipeline to learn camera motions without introducing synthetic side effects by leveraging the rendered data $\mathcal{X}_c$ and $\mathcal{X}_a$. Prior work suggests that a LoRA module trained specifically to capture the appearance of videos can help mitigate the domain gap in realistic video training datasets [5]. Motivated by this insight, we investigate whether similar techniques can be applied to absorb synthetic artifacts in *fully virtual-style videos*. Our training involves two steps.

Step 1: Appearance Adaptation We first train a LoRA, called the *appearance LoRA*, to model the visual characteristics of synthetic scenes without entangling motion, using the static videos $\mathcal{X}_a$ that contain no camera movement. The training objective is thus formulated as

$$\begin{aligned} &\textbf{Text-Based Control:} && \mathcal{L}_1(\Delta\boldsymbol{\theta}_a) = \mathbb{E}_{\boldsymbol{X}_0\sim\mathcal{X}_a,t,\boldsymbol{\epsilon}_t}[\boldsymbol{\epsilon}_t - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}_d+\Delta\boldsymbol{\theta}_a}(\boldsymbol{X}_t, \boldsymbol{c}, t)], \\ &\textbf{Trajectory-Based Control:} && \mathcal{L}_1(\Delta\boldsymbol{\theta}_a) = \mathbb{E}_{\boldsymbol{X}_0\sim\mathcal{X}_a,t,\boldsymbol{\epsilon}_t}[\boldsymbol{\epsilon}_t - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}_d+\Delta\boldsymbol{\theta}_a}(\boldsymbol{X}_t, \boldsymbol{c}, E_{\boldsymbol{\theta}_e}(\boldsymbol{p}_0), t)], \end{aligned} \tag{3.3}$$

where the $\Delta\boldsymbol{\theta}_a$ denote the appearance LoRA parameters, $\boldsymbol{c}$ only contains scene descriptions (and no camera motion descriptions), and $\boldsymbol{p}_0$ denotes a static camera trajectory.

Step 2: Camera Control Learning We then further fine-tune the models on the new camera motion, in the presence of the trained appearance LoRA (which is kept frozen), using the dataset $\mathcal{X}_c$. The training objective consists of two components. The first is the

standard diffusion loss:

$$\begin{aligned}
&\textbf{Text-Based Control:} && \mathcal{L}_2(\Delta\boldsymbol{\theta}_{cd}) = \mathbb{E}_{\boldsymbol{X}_0\sim\mathcal{X}_c,t,\boldsymbol{\epsilon}_t}[\boldsymbol{\epsilon}_t - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}_d+\Delta\boldsymbol{\theta}_a+\Delta\boldsymbol{\theta}_{cd}}(\boldsymbol{X}_t, \boldsymbol{c}_m \oplus \boldsymbol{c}, t)],\\
&\textbf{Trajectory-Based Control:} && \mathcal{L}_2(\Delta\boldsymbol{\theta}_{ce}) = \mathbb{E}_{\boldsymbol{X}_0\sim\mathcal{X}_c,t,\boldsymbol{\epsilon}_t}[\boldsymbol{\epsilon}_t - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}_d+\Delta\boldsymbol{\theta}_a}(\boldsymbol{X}_t, \boldsymbol{c}, E_{\boldsymbol{\theta}_e+\Delta\boldsymbol{\theta}_{ce}}(\boldsymbol{p}), t)],
\end{aligned} \tag{3.4}$$

where $\boldsymbol{c}_m$ is the camera motion description. $\Delta\boldsymbol{\theta}_{\boldsymbol{cd}}$ is a LoRA module only used for text-based model. For trajectory-based control, instead of LoRA fine-tuning, we perform full fine-tuning on the camera encoder, denoted by $\Delta\boldsymbol{\theta}_{ce}$.

In addition, we introduce an *optical-flow-based loss* to encourage camera motion learning by aligning frame-to-frame differences between predictions and ground truth:

$$\mathcal{L}_{\text{flow}} = \frac{1}{K-1}\sum_{k=1}^{K-1} \left\| \left(\hat{\boldsymbol{X}}_0^{(k+1)} - \hat{\boldsymbol{X}}_0^{(k)}\right) - \left(\boldsymbol{X}_0^{(k+1)} - \boldsymbol{X}_0^{(k)}\right) \right\|_1, \tag{3.5}$$

where $\hat{\boldsymbol{X}}_0$ is the predicted video reconstructed from the noisy input $\boldsymbol{X}_t$; (k) denotes the frame index. The final training loss in step 2 is therefore $\mathcal{L} = \mathcal{L}_2 + \lambda\mathcal{L}_{\text{flow}}$, where $\lambda > 0$ balances two losses.

It is important to note that the camera control adaptation is performed on different model components for text- and trajectory-based methods. For text-based control, the camera LoRA is performed on the diffusion model weights, superimposed on top of the appearance LoRA. For trajectory-based control, the adaptation is performed on the trajectory encoder, which is separated from the appearance LoRA on the diffusion model, because the former is responsible for processing the trajectory information.

The fundamental idea behind dual adaptation is that since the appearance LoRA already learns the virtual appearance information, the camera control learning no longer needs to learn the same information, and can focus on what the appearance LoRA does not learn – camera motion.

Inference After the training, we only deploy camera modules $\Delta\boldsymbol{\theta}_{cd}$ and $\Delta\boldsymbol{\theta}_{ce}$ during inference, while the appearance LoRA $\Delta\boldsymbol{\theta}_a$ is discarded. This would largely remove the undesirable synthetic appearance acquired during training. Specifically, given an input text prompt $\boldsymbol{c}$ with camera instruction $\boldsymbol{c}_m$ or camera pose $\boldsymbol{p}$, the video can be synthesized using $\boldsymbol{g}_{\boldsymbol{\theta}_d+\Delta\boldsymbol{\theta}_{cd}}(\boldsymbol{c}_m \oplus \boldsymbol{c})$ for text-based model or $\boldsymbol{g}_{\boldsymbol{\theta}_d}(\boldsymbol{c}, E_{\boldsymbol{\theta}_e+\Delta\boldsymbol{\theta}_{ce}}(\boldsymbol{p}))$ for trajectory-based model, respectively.

Style-aligned Prompt While the appearance LoRA helps address domain gaps, we observe that relying on it alone still introduces synthetic artifacts (see Sec. 3.5.4). To further disentangle the appearance and camera motion learning, during both training stages, which are trained on virtual videos, we append a virtual indicator to the input text prompt, $\boldsymbol{c}$, in the form of *'In this low-poly* `<VIRTUAL>` *scene.'* This would help the model differentiate the virtual style. During inference, this virtual indicator is dropped, which further removes the virtual appearance quality.

3.5 Experiments

In this section, we evaluate VividCam under various camera motions and compare it with state-of-the-art methods. We focus on the following questions: ❶ What types of camera motions can VividCam generate? ❷ Can VividCam provide precise camera control? ❸ Despite being trained exclusively on low-poly synthetic videos, can VividCam synthesize high-quality realistic videos?

3.5.1 Experiment Settings

Implementation For all experiments, we use `CogVideoX-5B` [4] as the base model. For *text-based control*, the LoRA rank is 128 and 512 for appearance and camera learning,

respectively. Notably, different camera motion types are trained using a single camera LoRA module. For *trajectory-based control*, we fine-tune the `AC3D` model built on `CogVideoX-5B`. Similarly, a single trajectory encoder is trained to handle multiple motion types.

Camera Motions As shown in Table 3.1, we systematically consider three broad categories of camera motions: *simple*, *composed*, and *complex* movements. *Simple* camera movements refer to basic motions in six different directions. *Composed* movements combine two simple motions. Specifically, when motion 1 is combined with motion 2, the first half of the video follows motion 1, and the second half motion 2. We consider combinations of {push in, pull out} $\times$ {truck left, truck right}, resulting in four combinations. Beyond these, *complex* camera motions include more unconventional practices, divided into *expressive* motions that convey semantic meaning (e.g., seeking objects, handheld shake) and *stylized* motions that create dramatic visual effects (e.g., explosive shake, camera rotation). All of these camera motions are motivated by realistic applications in everyday videos and filmmaking.

3.5.2 Qualitative Results

We present qualitative results of text-based control in Fig. 3.4. We highlight the following three features:

Stable camera motion We observe that the generated videos exhibit stable camera motion. For example, in the first row, the camera steadily pushes forward at a constant speed, as evidenced by the predictable changes in the sizes and positions of objects, such as the stone annotated in the frames.



Camera: The camera **pushes forward, focusing** on a roaring campfire.
Content: The campfire is surrounded by logs and stones, with a forest backdrop.



Camera: The camera **pushes forward**, highlighting a toddler blowing bubbles. Then, the camera **trucks right**.
Content: A toddler giggles as bubbles float away in a sunny backyard, with toys and a garden visible in the background.



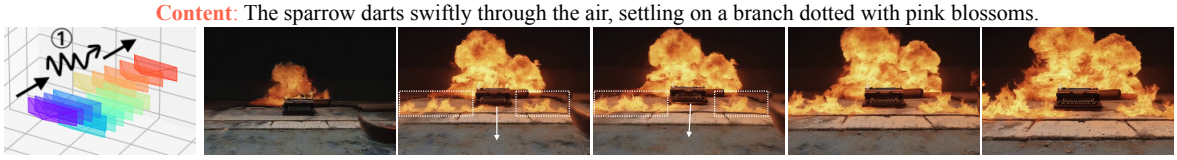
Camera: The camera **orbits** the static snow globe.
Content: A snow globe featuring a tiny winter village sits on a mantelpiece, surrounded by festive decorations.



Camera: The camera **pans around**, **searching** for a static steaming loaf of bread. Upon finding it, the camera **locks focus** and **pushes in** on the golden crust. **Content**: A loaf of bread cools on a wooden cutting board, with a knife and a small dish of butter.



Camera: The camera first **tracks** a flying sparrow. Then, the camera **switches focus** to a black-feathered crow.
Content: The sparrow darts swiftly through the air, settling on a branch dotted with pink blossoms.



Camera: The camera **pushes forward** with **intermittent explosive tremors, focusing** on a typewriter covered in ash.
Content: Nearby a typewriter, a wall of flame roars as a gas line explodes, shaking the entire area.



Camera: The camera **rotates 180 degrees in place**.
Content: An astronaut in a white spacesuit slowly spins weightlessly in the silence of outer space in the spaceship.

Figure 3.4: Qualitative results of diverse camera motions. From top to down: ❶ Push forward; ❷ Push forward, then truck right; ❸ Orbit shot; ❹ Pan around, then focus on one object; ❺ Switch focus between objects; ❻ Camera shaking; ❼ Camera rotating. The three panels correspond to simple, composed, and complex camera motions.

Table 3.2: Camera pose precision measurements. The best TransErr and RotErr values are in **bold**, and the second-best are underlined.

	Simple Motion			Composed Motion			Complex Motion		
	TransErr ↓	RotErr ↓	FVD ↓	TransErr ↓	RotErr ↓	FVD ↓	TransErr ↓	RotErr ↓	FVD ↓
CameraCtrl [62]	0.3578	0.1358	2577.35	0.3835	0.1989	2000.37	0.5753	0.7042	1503.76
CogVideoX [4]	0.3113	**0.0297**	1781.90	0.5107	**0.1338**	2168.99	0.4327	0.5067	1488.36
AC3D [63]	0.2639	0.0973	1996.32	0.4389	0.1958	2241.88	0.4271	0.5864	1719.82
VividCam-Cog	**0.1704**	0.0407	1808.30	0.2208	0.1593	2162.60	0.4011	0.5013	1866.40
VividCam-AC3D	0.2502	0.1162	2007.15	**0.1908**	0.1968	2280.71	**0.3376**	**0.3619**	1721.45

Ability to handle dynamic objects VividCam can generate precise camera motions for both static and dynamic objects. For example, in rows 1, 2, 5, and 6, we demonstrate that VividCam can synthesize high-quality moving objects, such as children, birds, and fire effects.

Mastery in unconventional camera motions VividCam can synthesize unconventional camera motions, many of which semantically depict story-like camera shots. For example, in row 4, VividCam simulates a common scenario where a person looks around for bread and locks focus once they find it. This capability suggests broad applications beyond simple camera movement controls. Additionally, we note that some delicate motions, like shaking in row 6, are difficult to convey through static images. To illustrate this, we annotate the shaking frames to highlight the blur and camera motion.

3.5.3 Quantitative Evaluations

In this section, we quantitatively compare our framework with existing models and methods. We compare our work with the following representative baselines:

CameraCtrl [62] enables trajectory-based camera control for U-Net diffusion models via Plücker embedding. In experiments, we provide both prompts and camera poses as inputs.

CogVideoX [4] demonstrates text-based camera motion control, likely due to ex-

Table 3.3: Human study results. The best scores are shown in **bold**, and the second-best are underlined.

	Simple Motion		Composed Motion		Complex Motion	
	Action Correctness ↑	Realism ↑	Action Correctness ↑	Realism ↑	Action Correctness ↑	Realism ↑
CameraCtrl [62]	0.79	0.68	0.90	0.74	0.62	0.60
CogVideoX [4]	0.80	0.74	0.53	0.65	0.61	**0.68**
AC3D [63]	0.74	0.72	0.75	0.66	0.68	0.65
VividCam-Cog	**0.86**	0.77	**0.93**	0.74	0.77	**0.68**
VividCam-AC3D	0.74	**0.78**	0.90	**0.78**	**0.81**	0.65

posure to relative motion data during pre-training. In experiments, we provide prompts that combine both camera motion instructions and content descriptions as inputs.

AC3D [63] achieves trajectory-based camera control using ControlNet [14] on DiT [68] based models. Similar to CameraCtrl, we provide the same trajectory and prompt description as input.

For all methods, we use the official implementations and checkpoints from their repositories. We conduct experiments across three levels of motion: *simple*, *composed*, and *complex*. Each category consists of 100 input prompts.

Evaluation Metrics Following prior work [62, 97], we use *FVD* [98] to assess visual quality, and report *TransErr* and *RotErr* to evaluate camera action accuracy. However, in the text-based control setting, the model does not have access to the ground-truth trajectory, making these metrics potentially unfair. To address this, we conduct a *human study* to evaluate both the correctness of camera actions and the realism of the generated videos. We report *Action Correctness* and *Realism* scores from 88 participants.

Automated Evaluation Results We present the automated metric results in Table 3.2. Our text-based and trajectory-based methods are denoted as VividCam-Cog and VividCam-AC3D, respectively. As shown, our method demonstrates strong camera motion precision across diverse motion categories, as indicated by the low TransErr and RotErr values. Additionally, we highlight that our method preserves video quality,

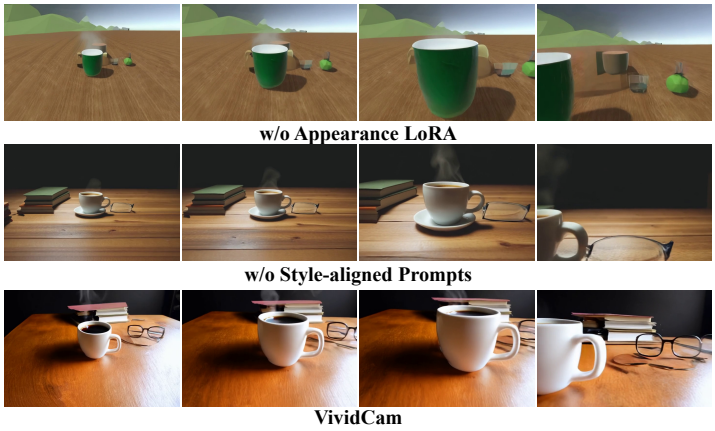


Figure 3.5: Visual examples illustrating the effects of appearance LoRA and style-aligned prompts.

as evidenced by the small FVD difference compared to the vanilla CogVideoX and AC3D. Notably, the original CogVideoX achieves good RotErr performance for simple and composed motions. We find this is because such motions involve minimal camera rotation, whereas vanilla CogVideoX typically produces videos with imprecise camera translation (high TransErr) and lacks camera rotation altogether. The following human evaluation further highlights its limitations.

Human study results We present human study results in Table 3.3. We observe that VividCam-Cog consistently generates more precise camera motion compared to the baselines across all categories of camera motion, and VividCam-AC3D shows advantages on more complex motions. Additionally, VividCam produces high-quality, realistic-style videos, as evidenced by the comparable realism scores given by participants.

3.5.4 Ablation Study

In this section, We conduct ablation studies to examine two key design choices: ❶ **Algorithm side:** Is it necessary to incorporate appearance LoRA and style-aligned

prompts? ❷ **Data side:** Does including realistic videos in the training data improve performance?

The role of appearance adaptation and style-aligned prompts We first examine how appearance LoRA and style-aligned prompt design help mitigate the negative effects of synthetic appearance. For ablation, we train one model without the appearance adaptation step. In parallel, we train another model in which the style-aligned prompt is omitted from the input text during training.

We conduct a human evaluation and show results in Table 3.4. Overall, we find the absence of appearance adaptation and style-aligned prompts leads to a clear decline in the model's ability to synthesize realistic videos. As shown in Figure 3.5, without appearance adaptation and style-aligned prompts, the generated appearance closely resembles the synthetic data, resulting in significantly degraded video quality (e.g., synthesized texture in 1^{st} row and distorted glasses in the 2^{nd} row). In contrast, our full model configuration effectively mitigates these visual artifacts, producing clean, realistic videos with the desired camera motion.

Add realistic videos in training data Second, we examine whether adding realistic videos improves training. We use the RealEstate10K [99] dataset, which provides annotated camera trajectories. Thus, experiments are conducted in the trajectory-based camera control setting. The training set includes 500 synthetic and 500 realistic videos. Results in Table 3.5 indicate that using only virtual training data performs comparably to mixing in realistic videos. We hypothesize that realistic data offers little benefit due to the limited diversity of camera motions in RealEstate10K. Given the extra human effort required to collect large-scale realistic data, VividCam offers a cost-efficient training paradigm.

Table 3.4: Effects of appearance LoRA (A-LoRA) and style-aligned prompts (S-Prompt).

	Action Correctness	**Realism**
w/o A-LoRA	0.75	0.60
w/o S-Prompt	**0.93**	0.65
VividCam	**0.93**	**0.74**

Table 3.5: Effects of realistic data in training.

	Action Correctness	**Realism**
Mixed data	0.88	0.75
Virtual data	0.91	0.72

3.6 Conclusion

In this chapter, we propose **VividCam**, which uses synthetic **Vi**rtual **Vid**eos to fine-tune video generation models to generate correct **Cam**era motions. Notably, we show that *synthetic videos do not need to be realistic at all.* In fact, VividCam shows that video diffusion models can effectively learn camera motion from surprisingly simple synthetic data, often comprising basic geometries rendered in low-poly scenes. Experiments show that models trained with VividCam can master various compound and complex camera motions, while maintaining a level of realism comparable to baselines trained on real footage. Ultimately, our work offers an efficient approach to synthesizing realistic videos with precise camera motion control, especially for unconventional motions.

Chapter 4

Text-Guided Visual Attribute Control in Diffusion Models

4.1 Introduction

Image generation has been a widely-studied research problem in computer vision, with many competitive generative models proposed over the last decade, such as generative adversarial networks (GANs) [100–104] and variational autoencoders (VAE) [105–108]. Recently, diffusion models [40,41,109,110], with their ability to generate high-quality and high-resolution images in different domains, have soon attracted wide research attention.

One important research direction regarding image generative models is the ability to *disentangle* different aspects of the generated images, such as semantic contents and styles, which is crucial for image editing and style transfer. A generative model with a good disentanglement capability should satisfy the following two desirable properties. First, it should permit separate modification of one aspect *without* changing other aspects. As an example shown in Fig. 4.1, in text-to-image generation, when the text input changes from "a photo of person" to "a photo of person with smile", the generative model should

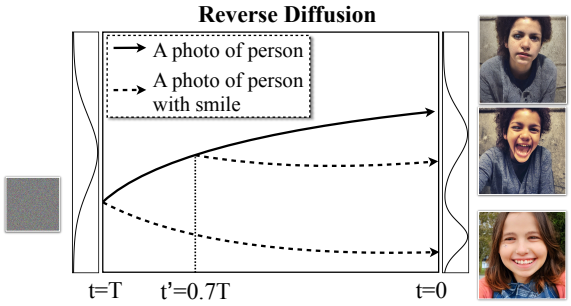


Figure 4.1: The disentanglement property of stable diffusion models. The top image is generated conditioned on "a photo of person". The bottom image is generated with all descriptions replaced with "a photo of person with smile", and changes the person's identity. The middle image is generated by partially replacing descriptions at later steps, and maintains the person's identity.

have the ability to modify just the expression of the person (*i.e.,* from the top image to middle image in Fig. 4.1) without changing the person's identity (the bottom image in Fig. 4.1). Second, the parameters learned from modifying one image should transfer well to other similar images. For example, the optimal parameters that can add smile to one person should also work for images of different people with different genders and races.

Previous studies have discovered that GANs are inherently endowed with a strong disentanglement capability. Specifically, it is found that there exist certain directions in the latent space separately controlling different attributes. Therefore, by identifying these directions, *e.g.,* via principal component analysis [111], GAN can achieve effective disentanglement *without* any re-training or fine-tuning. On the other hand, such an inherent disentanglement capability has yet to be found in diffusion models. Hence come our research questions: Do diffusion models also possess a disentanglement capability with the aforementioned nice properties? If so, how can we uncover it?

In this chapter, we seek to answer these research questions. Our finding is that for stable diffusion model [8], one of the diffusion models that can generate images based on an input text description, disentangled image modifications can be achieved by partial modifications in the text embedding space. In particular, if we fix the standard Gaussian

Table 4.1: Summarization of explored attributes. ✔ shows successfully disentangled attributes and ✘ shows failure cases. Small edits on the image are harder to be disentangled when the target attribute correlates with other parts of the image.

		Scenes	Person
✔	**Global**	**Styles** (children drawing, cyberpunk, anime), **Building appearance** (wooden, red brick), **Weather & time** (sunset, night, snowy)	**Styles** (renaissance, Egyptian mural, sketch, Pixar) **Appearance** (young, tanned, male)
	Local	Cherry blossom, rainbow, foothills	**Expressions** (smiling, crying, angry)
✘	**Small edits**	Cake toppings, remove people on the street	Hats, hair colors, earrings

noises introduced in the denoising process, and partially change the input text embedding from a neutral description (*e.g.,* "a photo of person") to one with style (*e.g.,* "a photo of person with smile"), the generated image will also shift towards the target style without changing the semantic content. Based on this finding, we further propose a simple, light-weight algorithm, where we optimize the mixing weights of the two text embeddings under two objectives, a perceptual loss for content preservation and a CLIP-based style matching loss. The entire process only involves optimizing over around 50 parameters and does not fine-tune the diffusion model.

Our experiments show that the inherent disentanglement capability in stable diffusion model can already disentangle a wide range of concepts and attributes, ranging from global styles such as painting styles to local styles like facial expressions, as shown in Table 4.1. As shown in Fig. 4.2, by learning the optimal mixing weights of the two descriptions, stable diffusion models can generate convincing image pairs that only modify the target attribute, and the optimal weights can generalize well to different images. The experiment results also show that our proposed image editing algorithm, without fine-tuning the diffusion model, can match or outperform the more sophisticated diffusion-model-based image-editing baselines that require fine-tuning. The findings of this chapter can shed some light on how diffusion models work and how they can be applied to image editing tasks.

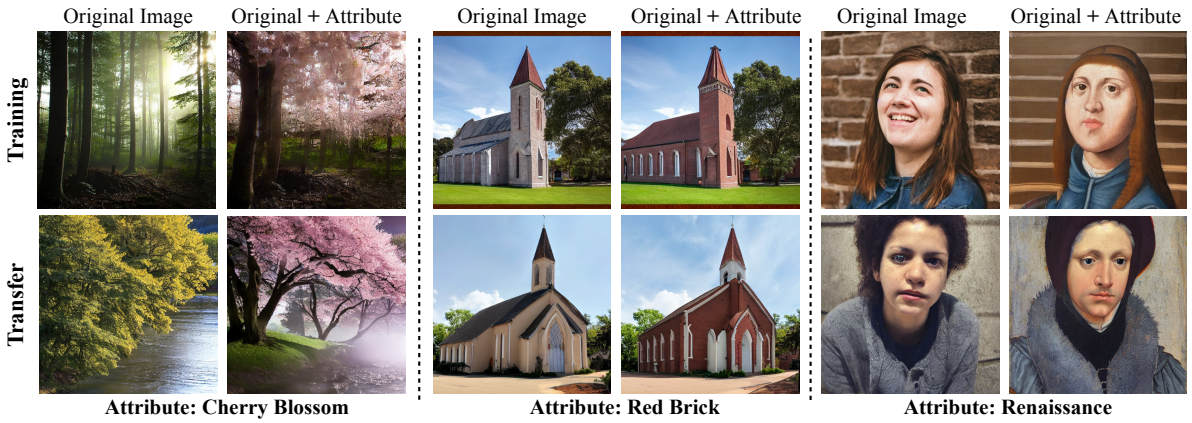


Figure 4.2: Example attributes disentangled from the stable diffusion model. Based on a *fixed* stable diffusion model, we disentangle the target attribute from a single training image. The learned parameters can then be applied to an unseen image and achieve the same edit.

4.2 Background and Prior Work

Disentanglement in Generative Models The ability to disentangle different attributes is a key desired property of generative models. Previous work that studies disentanglement mainly aims to learn parameters that allow modifications on a target aspect without changing other aspects, and the learned parameters should generalize to different images [104, 112, 113]. For pre-trained GANs [103, 104, 114, 115], it has been shown that the disentanglement can be achieved by moving towards particular directions in its latent space [111, 116–118], which will lead to attribute-only changes [119–121]. Multiple methods have been proposed to discover these latent directions, which leverage auxiliary classifiers [116, 117], principal component analysis [111] , contrastive learning [122], and information maximization [123]. Besides GANs, disentanglement has also been studied in VAE and flow-based models [124, 125]. Recently, two works study disentanglement in diffusion models. The first work disentangles attributes by learning a shift in the embedding space of an intermediate layer of U-Net [126, 127], such that applying the shift satisfies the disentanglement criteria. Particularly, they use a neural network to

generate such shifts. However, disentangling in the hidden layer representation of U-Net might be sub-optimal, as can be observed that their method struggles at disentangling holistic styles of the image. Moreover, their search space and number of parameters are much larger than ours. By contrast, we consider the text embedding space, which is more natural for text-to-image diffusion models and achieves comparable or better results with only 1.2% parameters of theirs. Another work that studies disentanglement is [128]. They train an encoder to generate an image-specific representation, which is later used as input to diffusion models to reconstruct the original image. Disentanglement is done by finding corresponding directions in this representation space similar to methods in GANs [116, 117]. However, their method requires re-training a diffusion model from scratch, whereas we fix the pre-trained diffusion model.

Diffusion Models Diffusion models [40, 41, 109, 110] are a family of generative models that have achieved state-of-the-art performance in image synthesis and have advanced research in super-resolution [129, 130], inpainting [131, 132], density estimation [133], video synthesis [10, 134], and areas beyond computer vision [135–139]. Building on top of diffusion models, various methods have been proposed to control the generation process through external models [140, 141] or additional inputs [142]. One type of conditional generation models is the text-to-image diffusion models [8, 9, 143, 144], which take text descriptions as inputs and generate images that match the text descriptions. Due to the expressiveness of text and superior generation quality of diffusion models, these models allow unprecedented control over generated images and have inspired many novel applications.

Image Editing Image editing is a widely-studied task [145, 146]. Many GAN-based editing works [121, 147–150] have demonstrated strong controllability. Recently, diffusion

models have been broadly adapted to image editing task [11, 13, 151–153]. With the CLIP encoder [154] that bridges text and image, generation process can be guided by arbitrary text descriptions [141]. To preserve the contents in a local region, [155] relies on an auxiliary mask, such that contents in the unmasked region are largely kept unchanged during generation. Moreover, some works [156–158] propose to invert the input image to find text embeddings that can synthesize the same object but in different scenes and views. Although these works have demonstrated successful edits, there are two limitations. First, most of the methods require fine-tuning diffusion models [156,157,159]. For each editing task, they have to fine-tune and store the whole diffusion model, making them unscalable to a large amount of edits. Second, many methods rely on auxiliary inputs such as image masks [155,160,161] or multiple examples of the edited object [157, 158], which are not always available. Besides, only editing masked region may cause incoherence between masked and unmasked regions. Leveraging the disentanglement in stable diffusion, we propose to perform image editing without auxiliary inputs and the need to fine-tune diffusion models, which is more practical to use.

4.3 Attribute Disentanglement in Stable Diffusion Models

In this section, we will explore the disentanglement properties inherent in diffusion models, and then propose an approach for disentangled image modification and editing utilizing these properties.

4.3.1 Preliminaries on Diffusion Models

We first provide a brief overview of the denoising diffusion implicit model (DDIM) [110] conditioned on input text descriptions that is used in the stable diffusion model [8], which we will be primarily studying in this chapter. Given a text embedding, denoted as $\boldsymbol{c}$, the goal of the text-conditioned DDIM is to generate an image, denoted as $\boldsymbol{X}_0$, that conforms to the text description. DDIM defines a set of noisy images, $\boldsymbol{X}_{1:T}$, by adding Gaussian noise to $\boldsymbol{X}_0$ according to a predefined diffusion process. Each $\boldsymbol{X}_t$ is corrupted with a larger noise than $\boldsymbol{X}_{t-1}$, and $\boldsymbol{X}_T$ is very close to standard Gaussian noise. The generation process of DDIM tries to denoise from $\boldsymbol{X}_T$ all the way back to $\boldsymbol{X}_0$. Specifically, in the first step, $\boldsymbol{X}_T$ is randomly drawn from a standard Gaussian distribution. Then, each $\boldsymbol{X}_{t-1}$ is inferred from $\boldsymbol{X}_t$ via the following denoising process:

$$\boldsymbol{X}_{t-1} = \gamma_{t0}\boldsymbol{X}_t + \gamma_{t1}\boldsymbol{\epsilon}_\theta(\boldsymbol{X}_t, t, \boldsymbol{c}_t), \tag{4.1}$$

where $\boldsymbol{c}_t$ is the text embedding used at step t. In most common settings, generating one image only requires one text description $\boldsymbol{c}$, so $\boldsymbol{c}_t = \boldsymbol{c}$ for all t. Here we make it dependent upon t to accommodate the discussions in the following subsections. $\boldsymbol{\epsilon}_{\theta^*}(\boldsymbol{X}_t, t, \boldsymbol{c}_t)$ is a pre-trained denoising network that infers $\boldsymbol{X}_0$ given the input of $\boldsymbol{X}_t$ and $\boldsymbol{c}_t$. The parameters of the denoising network, θ^*, are considered fixed throughout this section. γ_{t0} and γ_{t1} are defined as

$$\gamma_{t0} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}}, \quad \gamma_{t1} = \sqrt{1-\alpha_{t-1}} - \sqrt{\frac{\alpha_{t-1}}{\alpha_t} - \alpha_{t-1}}, \tag{4.2}$$

and $\alpha_{0:T}$ are hyperparameters that govern diffusion process.

Note that in the general DDIM framework, a Gaussian noise is added to each denoising step in Eq. (4.1), but we follow the convention [159] to set the variance to 0 for better stability. Therefore, the generated image is a *deterministic* function of initial random noise $\boldsymbol{X}_T$ and the text descriptions $\boldsymbol{c}_{1:T}$. Thus we introduce the following notation,

$$\boldsymbol{X}_0 = \boldsymbol{g}(\boldsymbol{X}_T, \boldsymbol{c}_{1:T}), \tag{4.3}$$

to summarize the image generation process. The stable diffusion model follows the same setup, except that the diffusion process is performed on a hidden embedding space, so Eq. (4.3) can also summarize its generation process.

4.3.2 The Disentanglement Properties

Now we are ready to study whether the stable diffusion model is inherently capable of disentangling styles from semantic content. For concreteness, we will present our findings based on one specific example, but the findings are consistent across different cases.

Consider two text embeddings, $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$. $\boldsymbol{c}^{(0)}$ is the embedding of a style-neutral description, "a photo of person", and $\boldsymbol{c}^{(1)}$ is the embedding of a description with an explicit style, "a photo of person with smile". When $\boldsymbol{c}^{(0)}$ is fed to the model, the generated image is a person with neutral expressions (top image in Fig. 4.1). We investigate whether the model can generate an image of the *same person* with only the facial expression changed, when we fix $\boldsymbol{X}_T$ (hence controlling all the randomness) and replace $\boldsymbol{c}^{(0)}$ with $\boldsymbol{c}^{(1)}$.

Case 1: Full Replacement In our first attempt, we replaced text embeddings at all denoising steps with $\boldsymbol{c}^{(1)}$. The resulting generated image undesirably changes the identity of the person, as shown in the bottom image in Fig. 4.1.

Case 2: Partial Replacement In our second attempt, we replaced the text embeddings to $\boldsymbol{c}^{(1)}$ only at the later denoising steps, *i.e.,*

$$\boldsymbol{c}_t = \boldsymbol{c}^{(1)}, \forall t \leq t', \quad \boldsymbol{c}_t = \boldsymbol{c}^{(0)}, \forall t > t', \tag{4.4}$$

where $t' = 0.7T$ in this example. In this case, the image can successfully maintain the identity of the person while only changing the smile, as shown in the middle image in Fig. 4.1.

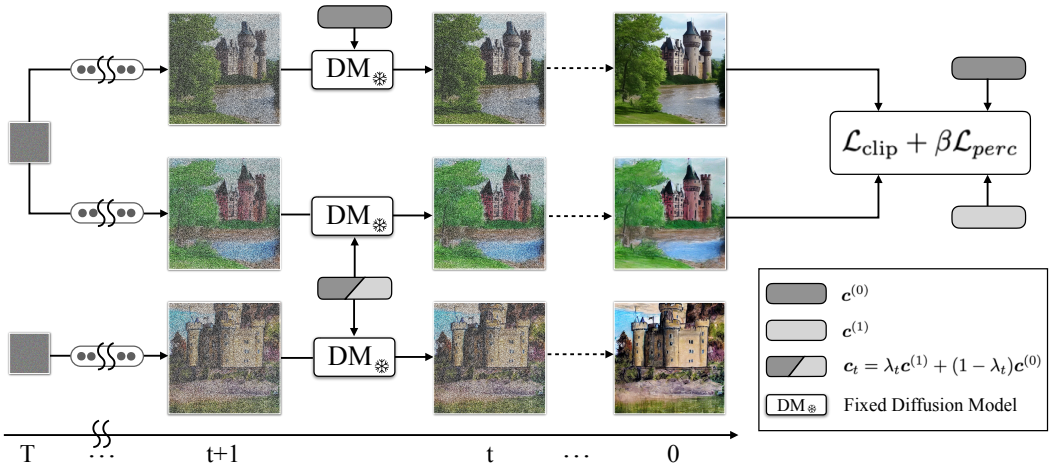


Figure 4.3: Overview of our method that finds optimal text embedding for disentanglement. In this example, $\boldsymbol{c}^{(0)}$ is the embedding of "A castle", and $\boldsymbol{c}^{(1)}$ is the embedding of "A children drawing of castle". **First two rows:** optimization process that finds the best soft combination of $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$, such that the modified image (the second row) changes the attribute without affecting other contents. **Last row:** the learned text embedding can be directly applied to a new image, which leads to the same editing effect.

As can be concluded, there exists an inherent disentanglement capability in stable diffusion models, which can be triggered by partially replacing the text embeddings, but not with full replacement.

4.3.3 Optimizing for Disentanglement

In Sec. 4.3.2, we have seen one way of realizing a disentangled generation of image using $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$. However, this is not necessarily the optimal way of combining the two embeddings in terms of disentanglement. In this subsection, we will propose a principled and tractable optimization scheme to combine a given pair of $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$ to achieve the best disentanglement, illustrated in Fig. 4.3.

The key relaxation for the optimization framework is the *soft combination* of the text embeddings. Specifically, instead of feeding either $\boldsymbol{c}^{(0)}$ or $\boldsymbol{c}^{(1)}$ at each denoising step t, we

feed a soft combination of the two, namely

$$\boldsymbol{c}_t = \lambda_t \boldsymbol{c}^{(1)} + (1 - \lambda_t)\boldsymbol{c}^{(0)} \equiv \boldsymbol{c}_t^{(\lambda)}, \tag{4.5}$$

where λ_t is a learnable combination weight. The soft combination offers a much richer representation power, with both cases discussed in Sec. 4.3.2 being its special cases (by setting λ_t to either 1 or 0).

Given a random noise $\boldsymbol{X}_T$ and the text description pair $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$, the optimization procedure for $\lambda_{1:T}$ is as follows. First, two images are generated, a style-neutral one generated with $\boldsymbol{c}^{(0)}$ and the other generated with $\boldsymbol{c}_{1:T}^{(\lambda)}$:

$$\boldsymbol{X}_0^{(0)} = \boldsymbol{g}(\boldsymbol{X}_T, \boldsymbol{c}^{(0)}), \quad \boldsymbol{X}_0^{(\lambda)} = \boldsymbol{g}(\boldsymbol{X}_T, \boldsymbol{c}_{1:T}^{(\lambda)}). \tag{4.6}$$

Our goal is to find an optimal $\lambda_{1:T}$ such that $\boldsymbol{X}_0^{(\lambda)}$ maintains the same semantic content as $\boldsymbol{X}_0^{(0)}$ but conforms to the style described in $\boldsymbol{c}^{(1)}$, which is achieved by solving the following optimization problem similar to [121, 126]:

$$\min_{\lambda_{1:T}} \mathcal{L}_{\text{clip}}(\boldsymbol{X}_0^{(0)}, \boldsymbol{X}_0^{(\lambda)}, \boldsymbol{c}^{(0)}, \boldsymbol{c}^{(1)}) + \beta \mathcal{L}_{perc}(\boldsymbol{X}_0^{(0)}, \boldsymbol{X}_0^{(\lambda)}). \tag{4.7}$$

β is the hyperparameter that balances the two loss terms. $\mathcal{L}_{\text{clip}}$ is the directional CLIP loss [162] that encourages the image change from $\boldsymbol{X}_0^{(0)}$ to $\boldsymbol{X}_0^{(\lambda)}$ matches the text change from $\boldsymbol{c}^{(0)}$ to $\boldsymbol{c}^{(1)}$ in the CLIP embedding space.[1] $\mathcal{L}_{perc}$ is the perceptual loss [163] that prevents drastic changes in semantic content:

$$\mathcal{L}_{perc}(\boldsymbol{X}_0^{(0)}, \boldsymbol{X}_0^{(\lambda)}) = \|\boldsymbol{h}(\boldsymbol{X}_0^{(0)}) - \boldsymbol{h}(\boldsymbol{X}_0^{(\lambda)})\|_1, \tag{4.8}$$

where $\boldsymbol{h}(\cdot)$ denotes a perceptual network that encodes a given image. In stable diffusion models, the number of denoising steps can be as few as 50 and so are the corresponding $\lambda_{1:T}$, so this optimization problem involves very few parameters and does not need to fine-tune the diffusion model itself.

[1] In practice, we use a different text embedder for $\mathcal{L}_{clip}$, but for brevity we still use $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$ to denote the embeddings.

4.3.4 Extension to Image Editing

With the disentangled image modification algorithm developed, we can now extend the approach to achieve disentangled image editing. The only difference of the image editing setting compared to the settings in the previous subsections is that rather than having the diffusion model generate the neutral image conditioned on $\boldsymbol{c}^{(0)}$, the neutral image is now externally given, which we denote as $\boldsymbol{I}$ to show the distinction. Therefore, if we could find a value for initial random variable $\boldsymbol{X}_T$ such that the stable diffusion model can generate exactly the same image as $\boldsymbol{I}$ when conditioned on $\boldsymbol{c}^{(0)}$, we can then use the same approach in Sec. 4.3.3 for the image editing task.

To this end, we adapt the image inversion approach proposed in [13, 159], which recursively generates a set of noisy images, $\hat{\boldsymbol{X}}_{1:T}$ based on $\boldsymbol{I}$ as follows:

$$\hat{\boldsymbol{X}}_0 = \boldsymbol{I}, \quad \hat{\boldsymbol{X}}_{t+1} = \gamma'_{t0}\hat{\boldsymbol{X}}_t + \gamma'_{t1}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\boldsymbol{X}}_t, t, \boldsymbol{c}^{(0)}), \tag{4.9}$$

where

$$\gamma'_{t0} = \sqrt{\frac{\alpha_{t+1}}{\alpha_t}}, \quad \gamma'_{t1} = \sqrt{1-\alpha_{t+1}} - \sqrt{\frac{\alpha_{t+1}}{\alpha_t} - \alpha_{t+1}}. \tag{4.10}$$

It has been shown [159] that by setting $\boldsymbol{X}_T = \hat{\boldsymbol{X}}_T$ and following through the generation process in Eq. (4.1) with $\boldsymbol{c}_t = \boldsymbol{c}^{(0)}$, the resulting $\boldsymbol{X}_{0:T-1}$ would satisfy $\boldsymbol{X}_t \approx \hat{\boldsymbol{X}}_t, \forall t \in [0, T-1]$, and therefore $\boldsymbol{X}_0 \approx \boldsymbol{I}$.

To further close the approximation gap, we introduce a new diffusion process, where the approximation error is added as a correction term. Formally, the new diffusion process starts with $\tilde{\boldsymbol{X}}_T = \hat{\boldsymbol{X}}_T$, and then

$$\tilde{\boldsymbol{X}}_{t-1} = \gamma_{t0}\tilde{\boldsymbol{X}}_t + \gamma_{t1}\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{X}}_t, t, \boldsymbol{c}_t) + \boldsymbol{E}_t, \tag{4.11}$$

where the $\boldsymbol{E}_t$ is the correction term defined as

$$\boldsymbol{E}_t = \hat{\boldsymbol{X}}_{t-1} - \gamma_{t0}\hat{\boldsymbol{X}}_t - \gamma_{t1}\boldsymbol{\epsilon}_\theta(\hat{\boldsymbol{X}}_t, t, \boldsymbol{c}^{(0)}). \tag{4.12}$$

Again, for notational brevity, we summarize this new generation process as

$$\tilde{\boldsymbol{X}}_0 = \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{X}}_T, \boldsymbol{c}_{1:T}, \boldsymbol{E}_{1:T}). \tag{4.13}$$

It can be easily shown that $\tilde{\boldsymbol{g}}(\hat{\boldsymbol{X}}_T, \boldsymbol{c}^{(0)}, \boldsymbol{E}_{1:T}) = \boldsymbol{I}$.

Now that we have developed a generation process that can reconstruct $\boldsymbol{I}$, we can now follow the same procedure in Sec. 4.3.3 to perform image editing, with the image generation in Eq. (4.6) replaced with

$$\tilde{\boldsymbol{X}}_0^{(0)} = \tilde{\boldsymbol{g}}(\hat{\boldsymbol{X}}_T, \boldsymbol{c}^{(0)}, \boldsymbol{E}_{1:T}) = \boldsymbol{I}, \;\; \tilde{\boldsymbol{X}}_0^{(\lambda)} = \tilde{\boldsymbol{g}}(\hat{\boldsymbol{X}}_T, \boldsymbol{c}_{1:T}^{(\lambda)}, \boldsymbol{E}_{1:T}). \tag{4.14}$$

It is worth emphasizing that when generating $\tilde{\boldsymbol{X}}_0^{(\lambda)}$, the error correction terms $\boldsymbol{E}_{1:T}$ are still fixed to the ones computed for reconstructing $\boldsymbol{I}$. To further enhance the quality of the edited image, we adopt the re-diffusion approach in [11].

4.4 Experiments

We will perform experiments to explore the inherent disentanglement capability in the stable diffusion model and evaluate the performance of our method.

Implementation Details For all experiments, we use the diffusion model `stable-diffusion-v1-4` [8], which is pre-trained on `laion` dataset [164]. The pre-trained model is *frozen* throughout all experiments, and we keep the default hyperparameters of the model. All images generated by our method are in size 512×512. We use a variant of the DDIM sampler [165] to synthesize images with 50 total backward diffusion steps. To optimize $\lambda_{1:T}$, we use Adam [166] optimizer with learning rate 0.03. To balance two loss terms, we set β to 0.05 for all human face experiments and 0.03 for all experiments on scenes and buildings. Finally, when editing real images, the number of re-diffusion steps is 20.

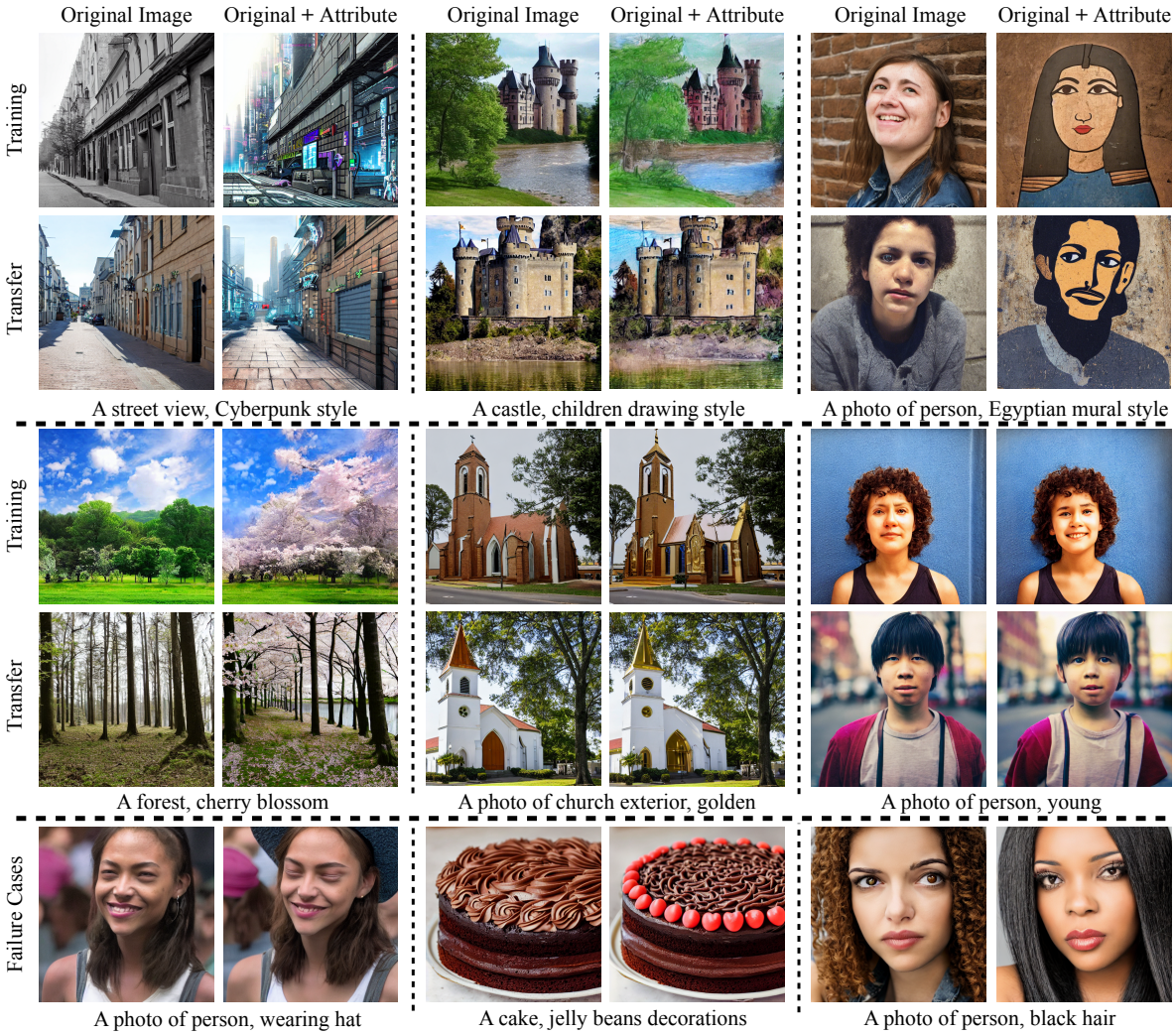


Figure 4.4: Example disentanglement results. Text description with style ($\boldsymbol{c}^{(1)}$) is shown below each row, which consists of the style-neutral text description ($\boldsymbol{c}^{(0)}$) and target attribute description, separated by comma. **Top two panels:** successful cases. **Bottom panel:** failure cases. Within each attribute, **first row:** results on optimization images; **second row:** results of transferring to unseen images; **left column:** source images; **right column:** modified images.

4.4.1 Exploring the Disentanglement Capability

Since the foundation of our method hinges on the inherent disentanglement capability in the stable diffusion model, we would like to first explore the strength and the limit of this capability. In particular, the research question we would like to answer is what are the attributes and objects that the stable diffusion model can inherently disentangle well, and what cannot. To this end, we perform a comprehensive qualitative study where we first compile a list of objects and attributes that have been tested upon in existing work on image disentanglement [111, 116, 126], as shown in Table 4.1. For each object-attribute pair in the list, we craft a text description pair, the neutral description $\boldsymbol{c}^{(0)}$ and the description with style $\boldsymbol{c}^{(1)}$. In particular, $\boldsymbol{c}^{(0)}$ includes just the name of the object, *e.g.,* "A castle", and $\boldsymbol{c}^{(1)}$ is constructed by appending the description of the attribute to $\boldsymbol{c}^{(0)}$, separated by comma (*e.g.,* "A castle, children drawing style"). Next, we generate five style-neutral images for each $\boldsymbol{c}^{(0)}$ and use the optimization method in Sec. 4.3.3 to disentangle the attribute from these five images.

Results The results are shown in Table 4.1, and some generated image pairs with their text descriptions are shown in Fig. 4.4, where the top two panels show some successful cases. In each panel, the first row (marked "Training") shows the results where the optimal $\lambda_{1:T}$ are learned specific to that image pair; the second row (marked "Transfer") shows the results of applying the learned $\lambda_{1:T}$ to a new image with the same object and target attribute. As can be observed, using our method, the stable diffusion model, without any re-training, can disentangle a surprisingly wide range of objects and attributes already. In particular, our method is strong at disentangling global styles that cover largely the whole image, such as scenery styles, drawing styles, and architecture materials, where the scenery layout or building structure is largely maintained while only the target attribute is modified. Our method can also disentangle many local attributes like

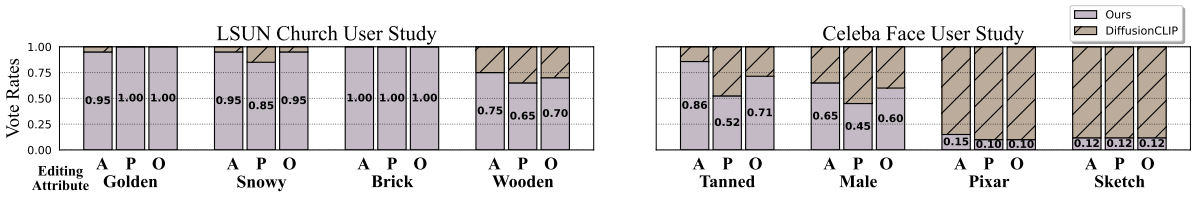


Figure 4.5: Human evaluations on **A (attribute similarity)**, **P (content preservation)**, and **O (overall quality)**. The number shown in each bar indicates the percentage of users who vote for our method is better than the baseline for the corresponding aspect.

facial expressions. Besides, the learned $\lambda_{1:T}$ has a great transferability to unseen images, thus satisfying both disentanglement criteria in Sec. 2.1.

On the other hand, stable diffusion has difficulties disentangling attributes that involve small objects, *e.g.,* adding small accessories, as shown in the bottom panel of Fig. 4.4. When the target attribute is added, the model tends to also change some other correlated attributes, such as the style of the cake or the identity of the person, which may be ascribed to the model's weaker control of finer-grained details. Nevertheless, these results suggest that the disentanglement capability in an unmodified stable diffusion model may be stronger than previously revealed, which provides a good justification for our method that applies it as is to image editing.

4.4.2 Evaluation on Disentangled Image Editing

In this section, we will evaluate our proposed method on the image editing task as described in Sec. 4.3.4.

Configurations Since there is no ground-truth for the image editing task, we conduct a subjective evaluation on Amazon Mechanical Turk. Specifically, we use the CelebA [167] and LSUN-church [168]. For each dataset, we use the first 20 images as the source image and perform 4 types of the edit used in [159] (*i.e.,* tanned, male, sketch, pixar for

human faces; and golden, wooden, snowy, red brick for churches). The main baseline is DiffusionCLIP, which is a state-of-the-art diffusion-based image editing approach that requires fine-tuning. Each subject was presented with a pair of images edited by the two methods from the same source image in a randomized order and asked three questions regarding the editing quality: (1) (*attribute similarity*) which image better incorporates the target attribute in a natural way; (2) (*content preservation*) which image better preserves other contents; and (3) (*overall*) which image has better overall quality.

Subjective Evaluation Results Fig. 4.5 shows the results of the percentage of answers that chose our method in each of the three questions and in each target attribute. As can be observed, in 6 out of the 8 attributes, our method outperforms the baseline in almost all three aspects, demonstrating the high quality of its disentanglement. We discovered a common failure case in DiffusionCLIP where the attributes are so over-optimized that some artifacts are introduced and some irrelevant parts of the image are modified. Our method, with its light-weight optimization, can avoid the over-optimization problem. On the other hand, our method is less competitive in human-related editing, underperforming the baseline in two attributes, "sketch" and "pixar". One potential cause is that the diffusion model in DiffusionCLIP is specifically trained to generate human images. Nevertheless, these results show that the inherent disentanglement capability in stable diffusion can enable powerful image editing that can even outperform the baseline that requires fine-tuning.

Qualitative Comparison with More Baselines Besides DiffusionCLIP, we have identified three other baselines that use diffusion models for image editing, which are Blended-Diffusion [155], Prompt-to-Prompt [13], and Imagic [156]. We generate qualitative comparisons by collecting the source and edited images shown on their



Figure 4.6: Visual comparisons. The source images and the corresponding baseline results are taken from the original papers. Each row indicates the comparison to one particular baseline.

papers and using our method to perform the edit. The edited images are displayed side-by-side in Fig. 4.6, with each row corresponding to one baseline. As can be observed, our method maintains the semantic content better than Blended Diffusion, which undesirably changes the details such as the grass. Our method performs relatively on par with DiffusionCLIP and Prompt-to-Prompt in terms of both attribute matching and content preservation. Finally, compared with Imagic, our method arguably changes slightly more semantic content (*e.g.,* adding stars) but the overall quality is very competitive. These results consistently verify the high quality of our edited images against each baseline's most representative results reported in their papers.

4.4.3 Ablation Study

As described in Sec. 4.3.3, the optimal $\lambda_{1:T}$ depends on specific values of $\boldsymbol{c}^{(0)}$, $\boldsymbol{c}^{(1)}$, and $\boldsymbol{X}^T$. We thus like to investigate whether our image editing algorithm is robust against variations in these inputs. To check the robustness against $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$ in the image editing task, we generate different edited images from the same source image by varying the text descriptions. We study three types of variations.

First, we study how the way that $\boldsymbol{c}^{(1)}$ appends the target attribute description can affect the results. Specifically, we fix $\boldsymbol{c}^{(0)}$ and the target attribute, and generate three $\boldsymbol{c}^{(1)}$'s with different ways to concatenate the target attribute: direct concatenation, concatenation separated by a comma, and concatenation separated by "with", as illustrated in the top row of Fig. 4.7. As can be seen, the resulting edit images are hardly impacted by such variations.

Second, we study whether varying the complexity of the target attribute description in $\boldsymbol{c}^{(1)}$ would affect the results. To this end, we again fix $\boldsymbol{c}^{(0)}$ and gradually add more modifiers describing the same target attribute to $\boldsymbol{c}^{(1)}$, as shown in the second row of Fig. 4.7. As can be seen, while having more modifiers can amplify the editing effects, having one modifier is sufficient to generate an effective edit.

Finally, we study whether variations in $\boldsymbol{c}^{(0)}$ would affect the results. Specifically, we fix the target attribute description and generate three versions of $\boldsymbol{c}^{(0)}$ that describe the same source object, a short description, a longer description by adding non-informative words, and a description generated by an image captioning model [169]. As shown in the bottom row of Fig. 4.7, the image editing is largely consistent in almost all cases, but fails with the $\boldsymbol{c}^{(0)}$ generated by the image captioning model. One possible cause is that the generated caption is usually lengthy and contains more details, which may overwhelm the target attribute description.



Figure 4.7: Text descriptions for image editing. The original image is shown in the first column. On the right columns, we demonstrate the results when changing text descriptions $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$.

4.5 Conclusion

In this chapter, we study the disentanglement property in the stable diffusion model. We find that stable diffusion inherently has the disentanglement capability, and it can be triggered by partially replacing the text embedding from a style-neutral description to one with desired style. Motivated by this finding, we propose a simple and light-weight disentanglement algorithm where the combination weights of the two text embeddings are optimized for style matching and content preservation. With only 50 parameters being optimized, our method demonstrates generalizable disentanglement ability and outperforms sophisticated baselines that require fine-tuning on image editing task.

Chapter 5

Region-Aware Attention Control for High-Fidelity Text-to-Image Generation

5.1 Introduction

Diffusion models [40, 41, 109, 110, 170] have recently revolutionized the field of image synthesis. Compared with previous generative models such as generative adversarial networks [100–103] and variational autoencoders [105, 106, 108], diffusion models have demonstrated superior performance in generating images with higher quality, more diversity, and better control over generated contents. Particularly, text-to-image diffusion models [8,9,143,144,171] allow generating images conditioned on a text description, which enables generation of creative images due to the expressiveness of natural language.

However, recent studies [172,173] have revealed that one critical limitation of existing diffusion-model-based text-to-image algorithms is the *low fidelity* with respect to the text descriptions – the content of the generated image is sometimes at odds with the



Figure 5.1: Example generation by stable-diffusion-v1-4 and our method. Stable diffusion model makes three types of errors that include *missing objects*, *mismatched attributes*, and *mislocated objects*. Errors are highlighted in red.

text description, especially when the description is complex. Specifically, typical errors made by stable diffusion models fall into three categories: *missing objects*, *mismatched attributes*, and *mislocated objects*. For example, in Fig. 5.1(a), stable diffusion model ignores the airplane even though it is mentioned in text; in Fig. 5.1(b), the model confuses "red car" and "black mailbox" and generates a red mailbox; in Fig. 5.1(c), the model locates teddy bear behind the toilet, despite the description "teddy bear is placed high above the toilet."

Such infidelity problems suggest that the cross-attention map on the text description may not be accurate. In particular, if we view the generation process of a diffusion model as a sequence of denoising steps, then the cross-attention on text descriptions can be considered as a function of both *spatial* (pixels) and *temporal* (denoising steps) information. Therefore, the inaccuracies of the cross-attention can result from the loose control over both the spatial and temporal dimensions. On one hand, spatial attention controls at what pixels the model should attend to each object and the corresponding

attributes mentioned in the text. If the spatial attention is incorrect, the resulting images will have incorrect object locations or miss-associated attributes. On the other hand, temporal attention controls when the models should attend to different levels of details in the text. As previous works have revealed, diffusion models tend to focus on generating object outlines at earlier denoising steps and on details at later [43]. Thus loose control over the temporal aspect of attention can easily lead to overlooking certain levels of the object details. In short, to improve the fidelity of text-to-image synthesis, one would need to explicitly control both spatial and temporal attention to follow an accurate and optimal distribution.

In this chapter, we propose a new text-to-image algorithm based on a pre-trained conditional diffusion model with explicit control over the spatial-temporal cross-attention map on text. The proposed algorithm introduces a layout predictor and a spatial-temporal attention optimizer. The layout predictor takes the text description as input and generates a spatial layout for each object mentioned in the text. Alternatively, the layout can also be provided by the user. Then the spatial-temporal attention optimizer imposes direct control over the spatial and temporal aspects of the attention according to the spatial layout. In particular, for the spatial aspect, we parameterize the attention map such that the attention outputs in the designated pixel region for an object are a weighted combination of attention over the entire text description and that over the local description that specifically describes the corresponding object. In this way, we manage to emphasize the attention over the object descriptions. For the temporal aspect, we allow the combination weights to change across time and optimized according to a CLIP objective that measures the agreement between the generated images and the text description. In this way, we allow the attention to focus more on the entire description at the early stage and gradually shift to the detailed local descriptions as the denoising process proceeds. The entire pipeline resembles a typical painting process of a human,

where each object's position is determined beforehand and the focus gradually shifts from global information to the local details of each object.

We conduct extensive experiments on datasets that contain real and template-based captions [174, 175] and our newly created synthetic dataset that contains complex text descriptions. Results show that our method generates images that better align with descriptions compared to other stable diffusion-based baselines. As shown in Fig. 5.1, our method effectively resolves the above-mentioned three errors. Particularly, controlling spatial attention locates objects at the desired position, and controlling temporal attention promotes the occurrence of objects with associated attributes. Our findings shed light on fine-grained control of diffusion models in text-to-image generation tasks.

5.2 Background and Prior Work

Diffusion Models Diffusion models are a class of generative models that have demonstrated strong performance on image synthesis tasks [40, 41, 109, 110, 140, 170]. These models synthesize images by sampling a noisy image from the standard Gaussian distribution and iteratively denoising it back to a clean image. Their impressive performance has advanced research in multiple computer vision areas including inpainting [131, 132, 176], image editing [11, 13, 43, 152], super-resolution [129, 130], video synthesis [10, 134, 177], and applications beyond computer vision [135, 178, 179]. Among these, text-to-image diffusion models have gained significant attention [3, 9, 144, 171, 180]. Taking text descriptions as inputs, these models generate high quality images that are semantically aligned with the input text, which have led to many creative and artistic applications.

Enhancing the Controllability of Text-to-Image Diffusion Models While text-to-image models have shown promising results, recent works have highlighted cases where models fail to generate high-fidelity images with respect to the input text [172, 173]. To this end, prior works have explored various ways to enhance the controllability of text-to-image diffusion models. One line of work enhances the controllability by improving diffusion models' ability to understand natural language, which includes using a more powerful text encoder that is separately trained on language modeling tasks [9,171], incorporating linguistic structures in the text to guide the cross-attention between image and text [172], and decomposing a complex text description into multiple components that are easy to generate [151]. Another line of work conditions diffusion models' generation on auxiliary inputs such as object layout [15, 171, 181, 182] and silhouette [14, 183–186]. By modifying diffusion models' attention operation according to these auxiliary information or directly fine-tuning diffusion models to take these auxiliary inputs, they are able to control the location and shape of the objects in the image. Finally, some work adds temporal aspect control on diffusion models by modifying the input text condition at each denoising step, which allows them to disentangle a desired attribute from other contents [43]. Different from prior works, our method imposes both spatial and temporal control in cross-attention layer. Moreover, our method does not require auxiliary inputs and does not fine-tune the diffusion model.

5.3 Method

5.3.1 Problem Formulation

We focus on the standard text-to-image problem. Given a text description, denoted as $\boldsymbol{D}$, our goal is to generate an image that is consistent with $\boldsymbol{D}$. For a concrete exposition,

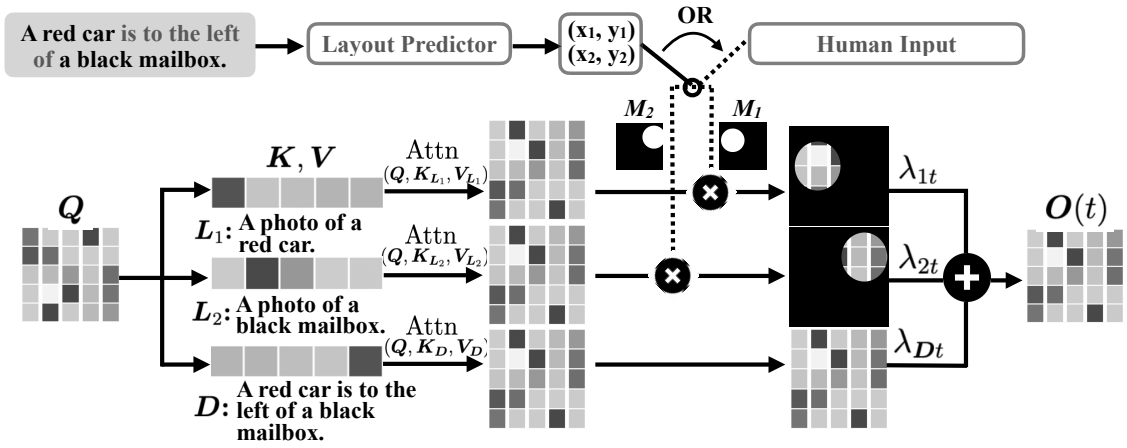


Figure 5.2: Overview of our text-to-image generation pipeline at one denoising step. Given input text $\boldsymbol{D}$, we first parse it and extract all objects mentioned, constructing local descriptions $\boldsymbol{L}_i$. Next, the layout predictor predicts the pixel region for each object in the text. The diffusion model attends to the global description $\boldsymbol{D}$ and additionally attends to the local description $\boldsymbol{L}_i$ in object i's region. The final attention output is a weighted combination of attention to both global and local descriptions, where the combination weights sum up to 1 for each pixel and are optimized for each denoising step to achieve a high fidelity with $\boldsymbol{D}$.

we will use an example description "*A red car is to the left of a black mailbox*" in the following. Our work aims to improve the fidelity of the generated image to text, which includes three requirements:

- **Object Fidelity:** The generated images should include all the objects mentioned in $\boldsymbol{D}$. In our example, the generated image should contain a car and a mailbox.
- **Attribute Fidelity:** The attributes of each object in the image should match those in $\boldsymbol{D}$. In our example, the car should be red; the mailbox should be black.
- **Spatial Fidelity:** The relative spatial positions of the object should match the description in $\boldsymbol{D}$. In our example, the car should be on the left and the mailbox on the right.

We will tackle these problems using a pre-trained, fixed stable diffusion model.

5.3.2 Method Overview

To achieve high-fidelity text-to-image generation, we propose an algorithm consisting of the following four steps.

Step 1: Object Identification We extract all the objects mentioned in $\boldsymbol{D}$, denoted as $O_{1:N}$, by eliciting the noun phrases using spaCy.[1] N is the total number of objects. In our sample, $O_1 =$ *"a red car"* and $O_2 =$ *"a black mailbox"*.

Step 2: Layout Prediction For each object O_i, we use a *layout predictor* to predict its pixel region, $\mathcal{R}_i$, which is a set of pixels roughly specifying where the object should be. The layout can also be provided by human users.

Step 3: Local Description Generation For each object O_i, we generate a local text description, $\boldsymbol{L}_i$, containing only that object information using a simple template. In our example, there are two local descriptions. $\boldsymbol{L}_1 =$ *"A photo of a red car"* and $\boldsymbol{L}_2 =$ *"A photo of a black mailbox"*.

Step 4: Attention Optimization During the generation process, we guide the diffusion model to combine attention to both the global description $\boldsymbol{D}$ and the local descriptions $\boldsymbol{L}_{1:N}$ according to the object layout. The attention combination weights are optimized for each input description.

The following subsections will provide further details about steps 2 and 4.

5.3.3 Layout Predictor

Our layout predictor is adapted from [187], which aims to predict the center coordinate $\boldsymbol{C}_i = [X_i, Y_i]$ for each object O_i. Specifically, the layout predictor is a transformer that

[1] `https://spacy.io/`.

takes the text description $\boldsymbol{D}$ as the input. At the output position where the input mentions the object O_i, we let the transformer output a set of Gaussian mixture model (GMM) parameters to fit the object's center coordinate $\boldsymbol{C}_i$. Formally, denote $\boldsymbol{f}_i(\boldsymbol{D};\boldsymbol{\theta})$ as the output of the layout predictor at the location of the mentioning object O_i. Then

$$\boldsymbol{f}_i(\boldsymbol{D};\boldsymbol{\theta}) = \bigcup_{k=1}^{K} \{\boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}, w_{ik}\}, \tag{5.1}$$

where $\boldsymbol{\mu}_{ik}$, $\boldsymbol{\Sigma}_{ik}$, and w_{ik} denote the mean, covariance matrix and the prior probability of mixture k; $\boldsymbol{\theta}$ denotes the network parameters.

To improve the fidelity of the object positions in the predicted layout, we introduce a hybrid training objective including an *absolute position objective* and a *relative position objective*.

Absolute Position Objective The absolute position objective provides direct supervision of the exact position of each object. Formally, we assume access to an image captioning dataset $\mathcal{D}_{\text{real}}$ with description $\boldsymbol{D}$ as well as the extracted labels of all the objects $\{O_i\}$ and their center coordinates $\{\boldsymbol{C}_i\}$. Then the training objective is to minimize the negative log-likelihood of the ground-truth coordinates under the predicted GMM distribution, *i.e.*

$$\mathcal{L}_{\text{abs}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{D},\{\boldsymbol{C}i\}\sim\mathcal{D}_{\text{real}}}\Big[\sum_{i=1}^{N} \ell_{\text{nll}}(\boldsymbol{C}_i; \boldsymbol{f}_i(\boldsymbol{D};\boldsymbol{\theta}))\Big], \tag{5.2}$$

where $\ell_{\text{nll}}(\boldsymbol{C}_i; \boldsymbol{f}_i)$ denotes the negative log-likelihood of ground truth $\boldsymbol{C}_i$ under the GMM specified by $\boldsymbol{f}_i$.

Relative Position Objective In many cases, the text description only mentions the relative positions of the objects, and thus it is more important to ensure the relative position of the predicted position is correct than the absolute position. To further enforce the fidelity of the relative positions, we introduce the following relative position objective.

To start with, we construct a synthetic dataset, $\mathcal{D}_{\text{syn}}$, which consists of text description $\boldsymbol{D}$ with explicit descriptions of relative spatial relations. We first randomly select N objects with attribute modifiers. We then select M pairs of objects to specify their relative spatial relation. For object pair (O_i, O_j), their spatial relation, R_{ij}, is randomly drawn from "left of", "right of", "above" and "below". Finally, we prompt GPT-3 [188] to generate the text description $\boldsymbol{D}$ that mentions all the objects and their relative positions. Our familiar "A red car is to the left of a black mailbox" is one such example.

Next, we introduce a loss to penalize violating the relative position. If object i is to the left of object j, we enforce that the rightmost mixture mean of object i is to the left of the leftmost mixture mean of object j. Formally,

$$\ell_{\text{rel}}(R_{ij} = \text{"left"}; \boldsymbol{f}_i, \boldsymbol{f}_j) = \max\{\max_k \boldsymbol{\mu}_{ik}(0) - \min_k \boldsymbol{\mu}_{jk}(0), -\delta\}, \tag{5.3}$$

where $\boldsymbol{\mu}_{ik}(0)$ denotes the zeroth element (the x-coordinate) of $\boldsymbol{\mu}_{ik}$, and δ is a pre-specified margin. The ℓ_{rel} of the other three types of relations are defined similarly. The relative position loss is thus the aggregation of ℓ_{rel} across the synthetic dataset:

$$\mathcal{L}_{\text{rel}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{D}, \{R_{ij}\} \sim \mathcal{D}_{\text{syn}}} \Big[\sum_{i,j: R_{ij} \neq \emptyset} \ell_{\text{rel}}(R_{ij}; \boldsymbol{f}_i, \boldsymbol{f}_j) \Big], \tag{5.4}$$

where $\boldsymbol{f}_i$ is short for $\boldsymbol{f}_i(\boldsymbol{D}; \boldsymbol{\theta})$.

Training and Inference The final training objective is the combination of both:

$$\mathcal{L}_{\text{layout}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{abs}}(\boldsymbol{\theta}) + \xi \mathcal{L}_{\text{rel}}(\boldsymbol{\theta}), \tag{5.5}$$

where ξ is a hyperparameter. During inference time, we randomly draw the center coordinate $\boldsymbol{C}_i$ from the predicted GMM. The pixel region for object O_i, $\mathcal{R}_i$, is defined as a circular region centered at the drawn $\boldsymbol{C}_i$ with a *fixed radius* r. $\mathcal{R}_i$ only roughly regulates the position of the generated objects, and the actual size of the object can go beyond or below the size of $\mathcal{R}_i$. Thus a fixed r is sufficient for this purpose.

5.3.4 Spatial-Temporal Attention Optimization

Recall that $\boldsymbol{D}$ is the global description and $\{\boldsymbol{L}_i\}$ are local descriptions for each object. Our goal is to guide the diffusion model to attend to not only $\boldsymbol{D}$, but also $\boldsymbol{L}_i$ in object i's region, $\mathcal{R}_i$, so that the model is more strongly prompted to generate the specific object as specified in the layout.

Spatial-Temporal Attention Recall the standard cross-attention [189] is defined as

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\big(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d}\big)\boldsymbol{V}, \tag{5.6}$$

where $\boldsymbol{Q} \in \mathbb{R}^{h\times w,d}$ is the queries vectors for each *pixel*, and $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{l,d}$ are key and value vectors for each *text description token*; h and w represent the image height and width, l represents the text description length; d represents the dimension of each attention vector.

Since we have multiple text descriptions, we define $\boldsymbol{K_D}, \boldsymbol{V_D}$ as the key and value vectors for the global description $\boldsymbol{D}$, and $\boldsymbol{K}_{\boldsymbol{L}_i}, \boldsymbol{V}_{\boldsymbol{L}_i}$ for the local description $\boldsymbol{L}_i$. Further, we introduce a set of binary mask matrices $\{\boldsymbol{M}_i\}$ to indicate the region for each object, *i.e.*

$$\boldsymbol{M}_i(x, y) = 1, \text{ if } (x, y) \in \mathcal{R}_i, \quad \text{and } 0, \text{ otherwise.} \tag{5.7}$$

Then the output of the attention layer of the denoising network at time t is defined as

$$\begin{aligned}\boldsymbol{O}(t) = &\sum_{i=1}^{N} \lambda_{it}\boldsymbol{M}_i \odot \text{Attention}(\boldsymbol{Q}, \boldsymbol{K}_{\boldsymbol{L}_i}, \boldsymbol{V}_{\boldsymbol{L}_i}) \\ &+ \Big(1 - \sum_{i=1}^{N} \lambda_{it}\boldsymbol{M}_i\Big) \odot \text{Attention}(\boldsymbol{Q}, \boldsymbol{K_D}, \boldsymbol{V_D}),\end{aligned} \tag{5.8}$$

where λ_{it} are the attention combination weights, and $\odot$ denotes element-wise multiplication. Note that the combination weights are functions of time t, which is motivated by the observation that different denoising steps control the generation of different levels of details. When synthesizing images, the combination weights λ_{it} are optimized on the fly

for each corresponding mask and time. By introducing this time dependency, we allow the diffusion model to focus more on the global description at earlier time steps and shift to local descriptions later.

Optimization Objective All the attention combination weights, $\boldsymbol{\lambda} = \{\lambda_{it}\}$, are determined by maximizing the consistency between the generated image and the text description as measured by the CLIP similarity [154]. We introduce two CLIP similarities, a global CLIP similarity that compares the entire image and the global description, and a set of local CLIP similarities that compare the images at each object region and the corresponding local descriptions. Formally, denote the generated image as $\boldsymbol{I}(\boldsymbol{\lambda})$, and its local patch at each object's region as $\boldsymbol{I}_{O_i}(\boldsymbol{\lambda})$.[2] Note that we adopt the *deterministic* PLMS denoising process [165] (*i.e.*, with $\sigma = 0$), so both $\boldsymbol{I}$ and $\boldsymbol{I}_{O_i}$ are deterministic functions of $\boldsymbol{\lambda}$. Then the loss for optimizing $\boldsymbol{\lambda}$ is given by

$$\mathcal{L}_{\text{attend}}(\boldsymbol{\lambda}) = -\text{CLIP}(\boldsymbol{I}(\boldsymbol{\lambda}), \boldsymbol{D}) - \gamma \sum_{i=1}^{N} \text{CLIP}(\boldsymbol{I}_{O_i}(\boldsymbol{\lambda}), \boldsymbol{L}_i), \tag{5.9}$$

where $\text{CLIP}(\cdot)$ denotes the CLIP similarity, and γ is a hyperparameter.

5.4 Experiments

We conduct experiments to evaluate our method's performance and generalizability. We also perform ablation study on important design choices of our method.

Implementing Details We adopt RoBERTa-base [190] as the base model for layout predictor and use 5 mixtures in GMM. We use stable-diffusion-v1-4 [8] pre-trained on the laion dataset [164] and freeze it throughout all experiments. All generated images

[2] To standardize the image patch sizes, $\boldsymbol{I}_{O_i}$ is obtained by cropping the original image with a minimum square that encompasses $\mathcal{R}_i$ and resizing it to 224×224.

are in the size of 512×512. We use PLMS sampler [165] to synthesize images with 50 denoising steps, and we use Adam [166] to optimize the layout predictor and attention combination weights.

5.4.1 Evaluation on Fidelity of Generated Images

We first evaluate our method on object, attribute, and spatial fidelities as introduced in Sec. 5.3.1 using both objective and subjective metrics.

Baselines We identify four baseline methods on generating images from complex text descriptions. (1) VANILLA-SD [8] is the pre-trained text-to-image stable diffusion model that directly generates images conditioned on the text description. (2) COMPOSABLE-DIFFUSION [151] is a diffusion-based compositional generation method. To generate images from a text description, the text is first decomposed into the conjunction of multiple components (*e.g.,* for the example in Sec. 5.3.1, the components are "A red car is to the left of a black mailbox." AND "A red car" AND "a black mailbox"). Each component is separately modeled by the diffusion model and then composed to generate an image by merging the outputs of each denoising step. (3) STRUCTURE-DIFFUSION [172] improves VANILLA-SD on generating images with better fidelities. They do so by first extracting noun phrases at different levels of the parsing tree. The model then separately attends to each noun phrase and combines the attention outputs by taking their arithmetic mean. Different from our method, they perform cross-attention on the whole image instead of the region specific to an object. (4) PAINT-WITH-WORDS [171] assumes that users provide the pixel region of each object to be generated. Pixels in the region then increase their attention weights to the text that describes the corresponding object, and the amount of increase is determined by heuristic rules. For a fair comparison, we use the pixel region predicted by our layout predictor.

Datasets and Metrics We conduct experiments on three datasets. (1) **MS-COCO** [191] contains photos taken by photographers and manually annotates the caption for each photo. We use the caption as the input text description. (2) **VSR** [175] is proposed for probing spatial understanding of vision-language models. Constructed from a subset of MS-COCO, it uses templates to generate captions that describe spatial relations in the image (*e.g.,* "The horse is to the left of the person."). (3) **GPT-synthetic** is the synthetic dataset introduced in Sec. 5.3.3. We manually check the test set to filter out sentences that do not conform to the specified spatial relation. Compared with VSR, this dataset has more diverse and complex text descriptions. The final dataset contains 500 descriptions, and we downsample MS-COCO and VSR to have the same size.

For automatic evaluations, we consider two metrics. (1) **Object Recall** measures the percentage of successfully synthesized objects over objects mentioned in the text. We use DETR [192] to detect objects in generated images. To calculate recall, we divide the number of detected objects in the text by the total number of objects in the text that also belong to one of the MS-COCO categories. It measures the **object fidelity** of generated images.(2) **Spatial Relation Precision** (SPRel Precision) measures the percentage of the correct spatial relations among all the relations whose corresponding objects are successfully synthesized. This metric measures the **spatial fidelity** of generated images. We consider the relation of left, right, above, and below because their correctness can be evaluated by comparing the bounding box centers.

Results The results are shown in Table 5.1. As shown in the table, our method outperforms all baselines on both object recall and SPRel precision. Specifically, compared with baselines that do not have spatial attention control (all but Paint-with-Words), our method is significantly better on SPRel precision, which illustrates the effectiveness of spatial control and indicates that it is difficult for the stable diffusion model to gen-

Table 5.1: Automatic evaluation of our method and baselines. SPRel Precision: Spatial Relation Precision. **Best numbers** are in bold. SPRel precision is not available on MS-COCO since most captions do not have an explicit spatial relation.

	MS-COCO		**VSR**		**GPT-synthetic**	
	Object Recall	SPRel Precision	Object Recall	SPRel Precision	Object Recall	SPRel Precision
Vanilla-SD	58.0%	–	62.1%	56.0%	42.4%	56.8%
Composable-Diffusion	51.8%	–	60.8%	72.2%	34.4%	52.6%
Structure-Diffusion	61.7%	–	62.3%	58.3%	43.0%	59.3%
Paint-with-Words	57.3%	–	63.8%	47.1%	45.9%	53.5%
Ours	**69.6%**	–	**65.1%**	**75.0%**	**47.2%**	**66.7%**

erate correct spatial relations without guidance on the layout. Our method moves the burden of generating objects at the correct location from diffusion models to a separate layout predictor, allowing the whole pipeline to achieve better spatial fidelity. On the other hand, compared with Paint-with-Words that does not impose the temporal attention control, our method achieves better object recall, showing that our optimized combination weights across the temporal dimension strike a better balance between local and global descriptions.

Fig. 5.3 demonstrates examples of text descriptions and images generated by our method and baselines. We observe that our method resolves three types of errors in baselines. First, our method alleviates the missing object issue (in the top panel). Baselines tend to focus on one object in the text and ignore other objects (*e.g.,* focusing on the elephant and ignoring the man in the second row), whereas our method generates all objects. Second, as shown in the middle panel, our method mitigates the mismatched attribute issue. Particularly, baselines struggle when multiple objects are mentioned in the text, where they mismatch the attribute and object (*e.g.,* red truck in the first row). Finally, the bottom panel shows examples where our method reduces the number of mislocated objects.

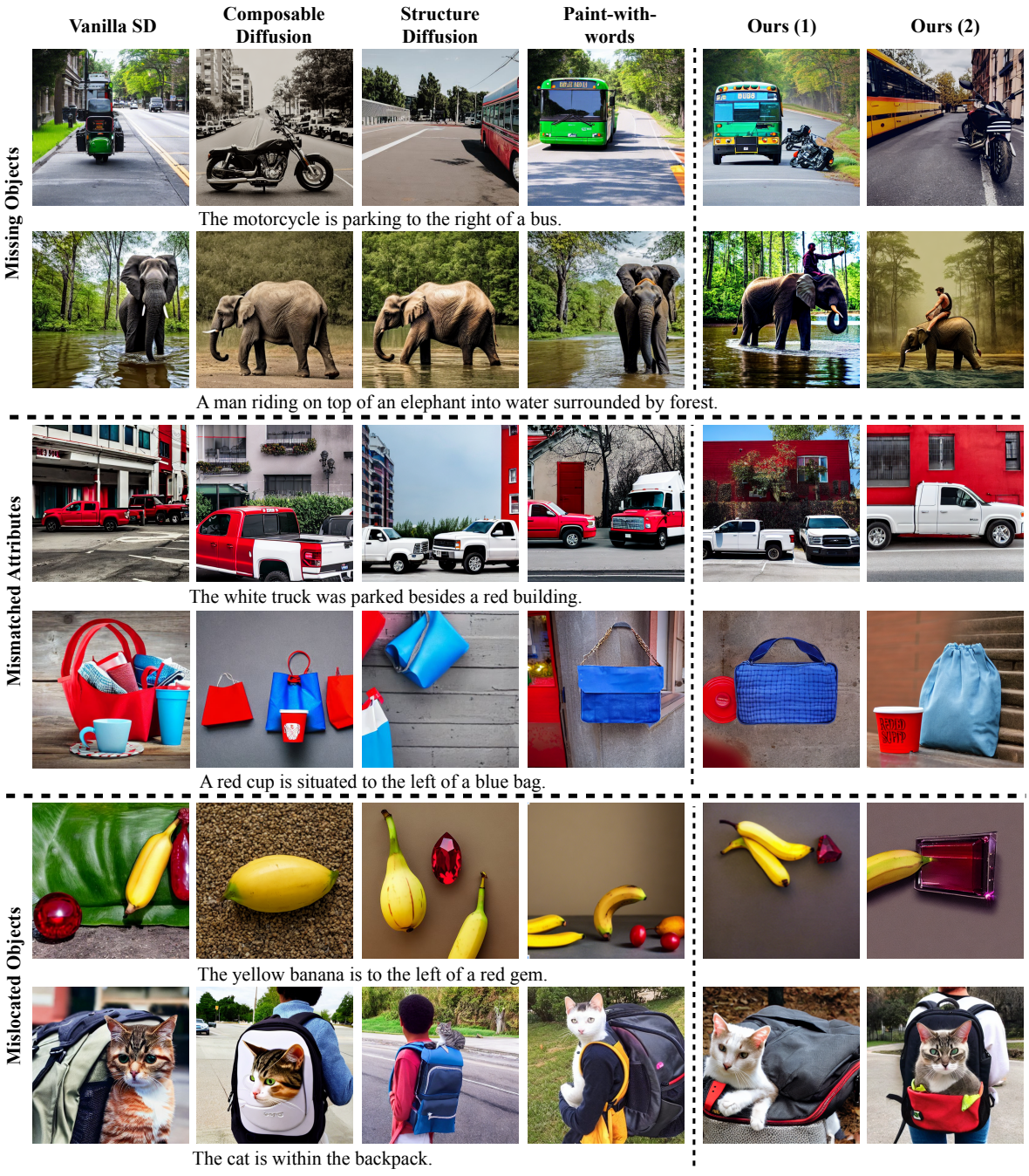


Figure 5.3: Example images generated by our method and baselines. Typical errors of baselines include missing objects, mismatched attributes, and mislocated objects. Ours (1)/(2) show the results with two different random seeds.

Table 5.2: Subjective evaluation of our method and baselines. **Best numbers** are in bold. Spatial relation is not available on MS-COCO because very few of its captions contain spatial relations. Attribute is not available on VSR as its captions do not consider attribute. Object, Attribute, and Spatial show the total score of 50 evaluations, where a model with the highest fidelity would achieve a score of 100. Overall denotes the percentage of generation of each method that is rated better than our method.

	MS-COCO			**VSR**			**GPT-synthetic**			
	Object (100) ↑	Attribute (100) ↑	Overall	Object (100) ↑	Spatial (100) ↑	Overall	Object (100) ↑	Attribute (100) ↑	Spatial (100) ↑	Overall
Vanilla-SD	66	62	42%	64	66	48%	60	58	60	38%
Composable-Diffusion	53	48	30%	63	60	18%	63	60	47	28%
Structure-Diffusion	60	63	48%	54	64	32%	58	54	58	32%
Paint-with-Words	**75**	**74**	52%	54	59	30%	65	52	57	32%
Ours	**75**	**74**	–	**68**	**77**	–	**84**	**83**	**86**	–

Subjective Evaluation To further evaluate the fidelity and quality of the generated images, we perform a subjective evaluation on Amazon Mechanical Turk. Specifically, we randomly sample 25 text descriptions for each dataset in Table 5.1. Each subject was presented with the text description and corresponding image generated by a single method or a pair of methods and asked the following four questions: (1) (*Object Fidelity*) Does the image contain all objects mentioned in the text? (2) (*Attribute Fidelity*) Are all synthesized objects consistent with their characteristics described in the text (*e.g.,* color and material)? (3) (*Spatial Fidelity*) Does the image locate all objects at the correct position such that the spatial relations in the text are satisfied (if an object in the relationship is missing, it is considered as an incorrect generation)? and (4) (*Overall*) Which image in the pair has higher fidelity with the text and has better quality? The first three questions are evaluated for each method individually with a score of 0, 1, or 2, where 2 denotes all objects/attributes/relations are correct and 0 denotes none of them is correct. The last question is evaluated on a pair of images, one generated by our method and the other by a baseline. Each generated image is evaluated by two subjects, so the total score of the first three questions is 100.

Table 5.2 shows the results. Our method achieves significantly better performance in most cases, especially on GPT-synthetic dataset that contains multiple objects and relations in the same description. The results show that our method is effective at

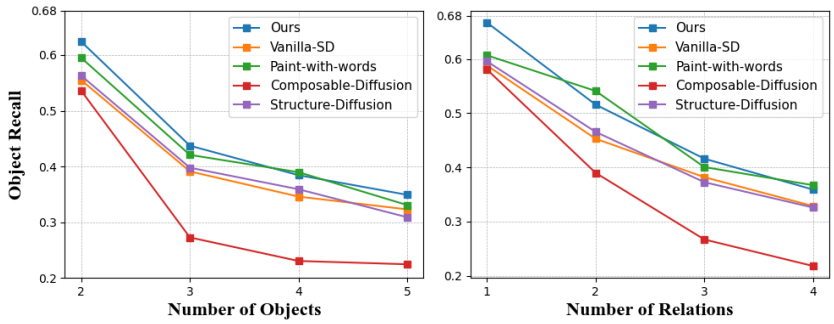


Figure 5.4: Performance when number of objects and spatial relations in the text increase.

generating images with high fidelity without sacrificing perceptual quality.

5.4.2 Additional Analyses

Performance on Each Text Complexity Level To explore the capability of our method, especially on complex text, we analyze the performance at each complexity level. Specifically, we use the GPT-synthetic dataset where multiple objects and relations can appear in the same description. We use the number of objects N and number of spatial relations M introduced in Sec. 5.3.3 as proxies for text complexity and plot the performance of each method as the text becomes more complex. *i.e.,* we plot the performance for each value of N regardless of M, and vise versa for M. As shown in Fig. 5.4, the performance of all methods decrease as the text becomes more complex, but our method still outperforms or is on par with others on complex descriptions. Note that Paint-with-Words uses the same layout predictor as ours, which partially explains its strong performance.

Generalizability to novel object combinations Generalizability is a critical requirement for text-to-image models. However, since MS-COCO dataset is collected from real photos, most of its captions contain object pairs and object-attribute pairs that are common in daily life, which are also more likely to overlap with stable diffusion's pre-

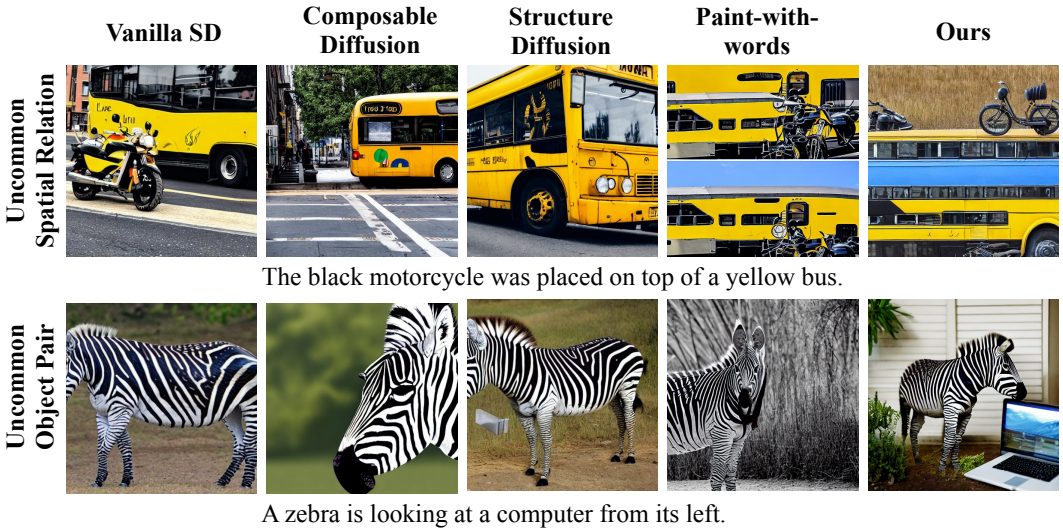


Figure 5.5: Example images generated by our method and baselines on novel object combinations.

training data. Thus we additionally evaluate our method on another synthetic dataset that is created similarly to Sec. 5.3.3, but contains *uncommon* object pairs, object-attribute pairs, and object spatial relations. Fig. 5.5 shows sample images generated by our method and baselines. It can be observed that our method successfully generates novel object pairs (*e.g.,* "zebra is looking at computer").

5.4.3 Ablation Study

In this section, we will investigate the influence of two important steps in our method, namely, the layout predictor and the spatial-temporal attention. We first study how spatial-temporal attention affects our performance. To do that, we consider three variants of attention.

First, we explore the attention **without spatial control**. Concretely, the diffusion model still attends to both global description and local descriptions. However, instead of only attending to a local description in the pixel region of the object, the model now attends to all local descriptions in the *whole image*. The combination weights at each

Table 5.3: Ablation performance on GPT-synthetic test set.

	Object Recall	SPRel Precision
Ours	**47.2%**	**66.7%**
No spatial control	39.6%	51.7%
No temporal control	43.8%	61.4%
No optimization	41.0%	55.6%

denoising step are optimized with only the global CLIP similarity in Sec. 5.3.4. Table 5.3 shows that its performance drops drastically, demonstrating the importance of spatial control.

Second, we explore the attention **without temporal control**, where combination weights remain the same for all denoising steps. Note that the weights can be different for each object and are optimized with $\mathcal{L}_{\text{attend}}(\boldsymbol{\lambda})$ in Sec. 5.3.4. The results in Table 5.3 show a significant degradation under both metrics, indicating the benefits of temporal dependency on attention.

Finally, we explore the attention **without optimization**. Specifically, we fix the combination weight $\lambda_{it} = \frac{1}{N}$ for all i and t, where N is the number of objects. This value is also the initialization point of $\{\lambda_{it}\}$ in our method. As shown in Table 5.3, the performance further drops compared to the no temporal control variant. The results indicate that optimizing the combination weights for a new text description is critical for generating images with high fidelity.

We further study the effects of our layout predictor, which includes the layout predictor trained with only one of the absolute and relative position objectives, the comparison with ground truth and user-provided pixel region, and a different strategy to construct the pixel region for an object.

5.5 Conclusion

In this chapter, we study the text-to-image synthesis task based on diffusion models. We find existing methods lack explicit control on cross-attention in diffusion models, which leads to the generation of low-fidelity images. We propose an algorithm that imposes control on cross-attention in spatial and temporal aspects. Experiments show our method outperforms baselines in generating high-fidelity images. Further ablation study verifies the effectiveness of our spatial and temporal attention control. One limitation of our method is the reliance on a time-consuming optimization scheme, which takes around 10 minutes for each text-to-image generation. Future work may consider striking a better balance between performance and efficiency.

Chapter 6

Measuring Visual and Textual Understanding in MLLMs

6.1 Introduction

The rapid advancement of large language models has driven considerable growth in their capabilities to produce fluent text in many domains, generating outputs exhibiting potential "reasoning" and "understanding" abilities [193–196]. Recently, multimodal large language models (MLLMs) have advanced on LLMs through training on native image inputs, to achieve impressive performance generating text describing and relating to input images [197–201], with applications in image captioning, visual question answering, and visual reasoning [202–205]. The swift evolution of MLLMs has enabled them to tackle increasingly sophisticated tasks that require multiple emerging abilities in complex scenarios. However, as model capabilities and deployment needs advance, the challenges in usefully evaluating them grow in kind.

Planning is a fundamental capability in intelligent systems that is particularly contested in LMs [206]. Visual spatial planning refers to the task of comprehending the

Table 6.1: Comparison with representative existing benchmarks. Note that we focus on spatial reasoning related capabilities. ✔ indicates that the measurement is covered by the corresponding benchmark, while ✘ indicates not covered.

Benchmark	Capabilities			Diagnosing Incapabilities
	Visual Recognition	Spatial Relation Understanding	Plan with Spatial Info	
MMMU [207]	✔	✘	✘	✘
MathVision [208]	✔	✘	✘	✘
MME [209]	✔	✔	✘	✘
SeedBench [210]	✔	✔	✘	✘
MM-Vet [211]	✔	✔	✘	✘
AlgoPuzzle [212]	✔	✔	✔	✘
MMSI [213]	✔	✔	✔	✘
VSP	✔	✔	✔	✔

spatial arrangement of objects in a scene and designing action plans to achieve a desired outcome. For example, the classical maze problem can be considered a visual planning task, where an agent is given an input image describing the maze environment and is asked to produce a viable path to navigate the player from the starting position to the goal. This task requires two capabilities: *image perception*, which enables the agent to understand the objects, environment and spatial relations present in the image, and *reasoning*, which enables the agent to perform strategic decision-making.

Visual spatial planning is an important capability in many potential applications for MLLMs, such as navigating in complex environments with autonomous driving [214,215] or manipulating objects with robotic hands [216,217]. However, although there have been increasingly more benchmarks to evaluate the vision processing capabilities of MLLMs, few systematically investigate the underlying factors affecting their (in)ability to perform visual spatial planning tasks. As shown in Table 6.1, while most of the existing benchmarks identify the capability to recognize objects as a key feature of MLLM, quite a few benchmarks [207, 208] do not evaluate their ability to understand spatial relations, and even fewer benchmarks [212, 213] examine the ability to use spatial information in complex tasks, *e.g.,* navigation on a road map based on the position towards goals. More

importantly, it is often unclear **why MLLMs perform so poorly on the spatial planning tasks that are generally considered easy for humans** [212]. As a result, two research questions emerged: ❶ [*Evaluation*] How performant are MLLMs in performing visual planning tasks? ❷ [*Diagnose*] What are the bottleneck capabilities, *e.g.*, perception or reasoning, that limit the performance of MLLMs in visual planning tasks?

To this end, we introduce Visual Spatial Planning (VSP), a benchmark specifically designed to evaluate and diagnose the spatial planning capabilities of MLLMs. As illustrated in Figure 6.1, the VSP benchmark consists of various environment settings designed to assess the capabilities of models in different scenarios. We focus on abstract scenarios to facilitate the design of diagnostic subtasks with consistent environments (Sec. 6.3.5). The first two scenarios, *Maze Navigation* and *Blocks World*, are basic environments developed from classical planning tasks. Additionally, we challenge the model's abilities in dynamic and realistic applications through the *Collision* and *Google Map* scenarios, respectively. All environments are fully observable through input images. The MLLMs are required to interpret the visual inputs, deduce the consequences of each action, and execute the designated tasks. To comprehensively evaluate the fine-grained capabilities needed for visual spatial planning, VSP includes 4.6K questions in 12 meticulously designed tasks that feature both simulated and photorealistic visual inputs. In addition to testing end-to-end spatial planning, these tasks test essential individual visual planning abilities, such as image perception and reasoning.

We apply the VSP benchmark to evaluate existing state-of-the-art MLLMs, including both open-source and private ones. Surprisingly, we find that even the most competitive MLLMs sometimes struggle in performing the simplest visual planning tasks, such as a 3x3 maze problem. Our fine-grained capability analysis further reveals that existing MLLMs have flaws in reasoning and bigger bottlenecks in perception. We believe the

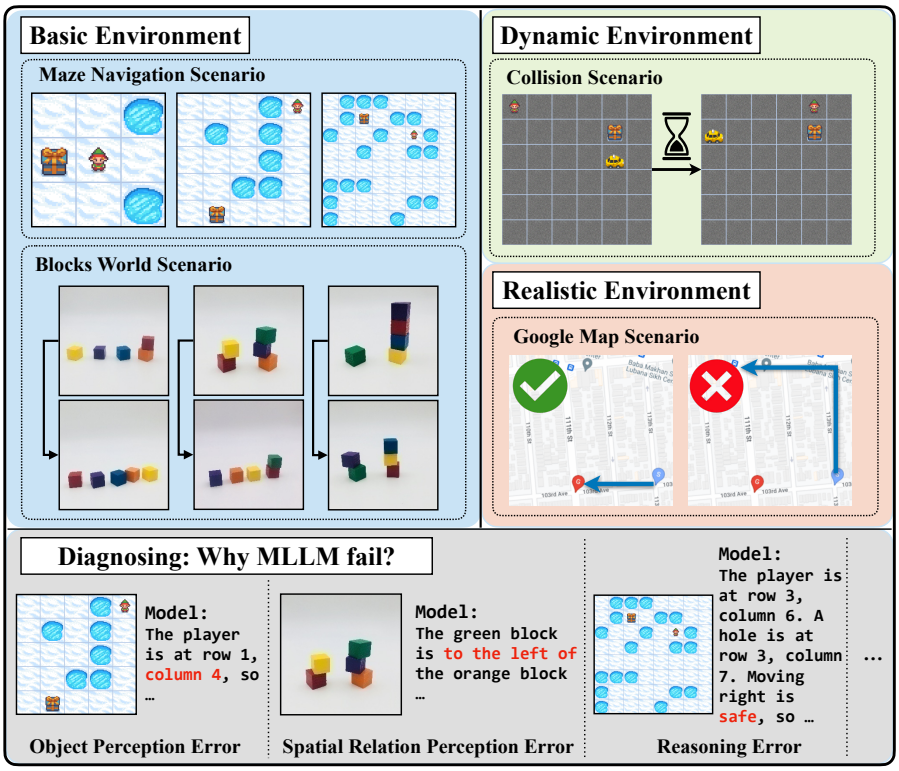


Figure 6.1: Overview of the VSP benchmark. The benchmark *evaluates* tasks that require MLLMs to perform complex reasoning by utilizing the spatial information they perceive, and *diagnoses* their failure modes using a series of carefully designed subtasks.

VSP benchmark highlights critical weaknesses in current MLLMs and sheds light on future directions for enhancing their spatial understanding and planning capabilities.

6.2 Background and Prior Work

6.2.1 General Planning in LMs

Planning has been a central focus of research in AI. Traditional work in AI planning includes using formal languages to represent and solve planning problems [218], and developing algorithms like dynamic programming and reinforcement learning to explore

environments and formulate viable plans [219, 220]. While these works mostly focus on predefined and restricted environments, recently, with the advancement of LMs, it has become intriguing to study whether LMs, with the potential to be general intelligent agents, can perform planning in different settings and environments [221–223]. Many works explore the best ways to activate the planning capabilities of LMs, including divide and conquer [224–227], grounding outputs in admissible actions [228, 229], retrospecting and refining [230, 231], and leveraging external tools [232, 233]. Meanwhile, with the increasing capabilities of LMs, growing research efforts are now dedicated to benchmark their planning capabilities in various complex environments [234–236].

6.2.2 Spatial and Visual Planning in LMs

Many planning tasks in LMs involve understanding visual environments and comprehending spatial information. In embodied agent studies, LMs play a crucial role in grounding visual entities with references in open-domain instructions and formulating plans based on spatial constraints. Consequently, they are increasingly used in physically grounded scenarios such as object rearrangement [216, 217, 237], cooking [238, 239], navigation [228, 229, 240], *etc.* LMs are also used in AIGC to propose spatial arrangements of entities following instructions [241]. While realistic planning tasks align with real needs, their complexity and expansive action spaces limit the analysis of LMs' detailed planning capabilities. Therefore, research also focuses on LMs' planning in abstract environments and games. For example, *mystery blocksworld* is a dynamically generated set of blocksworld tasks to test generalization in LMs [206]. Additionally, many text games have been introduced to test LMs' abilities in spatial understanding and imagination [234, 242–244]. While many of these studies transform visual information into text inputs, recent works also explore direct understanding and processing of visual inputs

leveraging MLLMs [212, 213].

6.2.3 Benchmarks for MLLMs

MLLMs have inherited and advanced many intriguing features from text-only LMs [243,245]. Benchmarks for MLLMs have rapidly emerged to evaluate performance in areas such as image content understanding [209,246], perception [247,248], knowledge [207,208, 249], and reasoning [207,209,250]. While these benchmarks quantify MLLMs' abilities in many fields, their capabilities in spatial understanding and reaction are relatively under-explored. Some benchmarks cover spatial relations understanding [210, 211], but often overlook the ability to devise complex spatial action plans based on visual environment constraints. Recently, several benchmarks explore the capability of planning utilizing spatial information, such QA and navigation in virtual or realistic scenarios [212, 213, 251,252]. While measurements on these benchmarks reveal some incapability of MLLMs, it is often unclear *why* they cannot perform well in these tasks. In this chapter, we focus on evaluating and diagnosing the ability to comprehend spatial arrangements of objects and devise action plans to achieve specific outcomes. Our benchmark highlights future directions for improving MLLMs towards models with general intelligence.

6.3 The Visual Spatial Planning Benchmark

6.3.1 Overview of the Benchmark

In this benchmark, our objectives are two-fold: ❶ quantify the visual spatial planning capabilities of MLLMs; and ❷ uncover current capability bottlenecks that limit the effectiveness of MLLMs in visual spatial planning tasks. The first objective requires *broader* task designs. Specifically, the tasks should range from classical planning tasks [236, 253]

to ones in those more dynamic and realistic environments. On top of that, the second objective requires *deeper* task designs. In particular, performing spatial planning in visual environments requires a series of cohesive steps. For example, to generate an accurate path to navigate a player to a goal, an agent needs to be able to correctly view and understand the visual map, reason to find which actions are safe or dangerous, and come up with a detailed plan to achieve the goal. Each of these steps could be challenging for a developing MLLM, and understanding which of these subtasks challenge them most will drive future improvement.

To this end, we propose the Visual Spatial Planning (VSP) benchmark, with the objective of measuring and diagnosing the capabilities of MLLMs in producing accurate spatial plans in visual environments. VSP consists of four scenarios: ❶ the simulated **Maze Navigation scenario**, whose main task is to move a game character through a maze; ❷ the photo-realistic **Blocks World scenario**, whose main task is to move blocks from a starting configuration to a goal configuration; ❸ the dynamic **Collision scenario**, whose main task is to determine if there is a danger of collision between two objects in the environment, and ❹ the realistic **Google Map scenario**, whose main task is to find a path in real streets of New York City. In addition to the main task, VSP introduces four sub-tasks that focus on the individual capabilities needed for the main task:

- **T1. Single Object Perception** – Determine the characteristics of a single object;
- **T2. Spatial Relation Perception** – Determine the relative positions of two objects;
- **T3. Environment Perception** – Find textual descriptions that describe the visual environment;
- **T4. Spatial Reasoning** – Determine the consequence of a series of actions or moves.

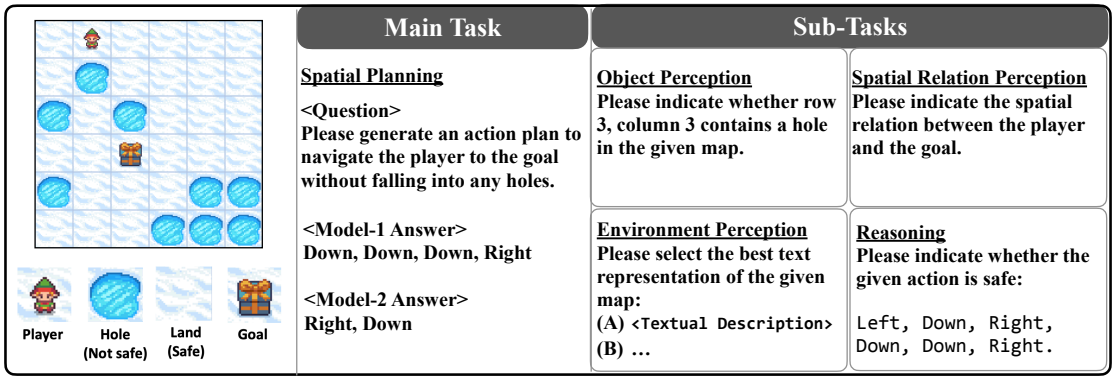


Figure 6.2: Overview of the *Maze Navigation* scenario.

The sub-task details are designed specific to both scenarios. Furthermore, to demonstrate the model's performance under different levels of environmental complexity, we establish progressive difficulty settings for various tasks, which are measured by parameters such as map size, minimum required number of actions, *etc.* In what follows, we introduce each scenario in detail, as well as the data curation and the task creation processes.

6.3.2 The Maze Navigation Scenario

The *Maze Navigation* scenario is inspired by the popular implementation [253] of a fully observable path-finding problem. As depicted in Figure 6.2 left, it simulates a classical grid world environment with a designated start and goal position, where part of the grids contain obstacles (the "holes") and cannot be passed through.

The main spatial planning task and the four sub-tasks are defined as follows:

• **Main Task** (Spatial Planning) – Generate a safe path to navigate from the start grid to the goal;

• **T1** (Single Object Perception) – Determine if a specified grid is safe;

• **T2** (Spatial Relation Perception) – Find spatial relations between elements;

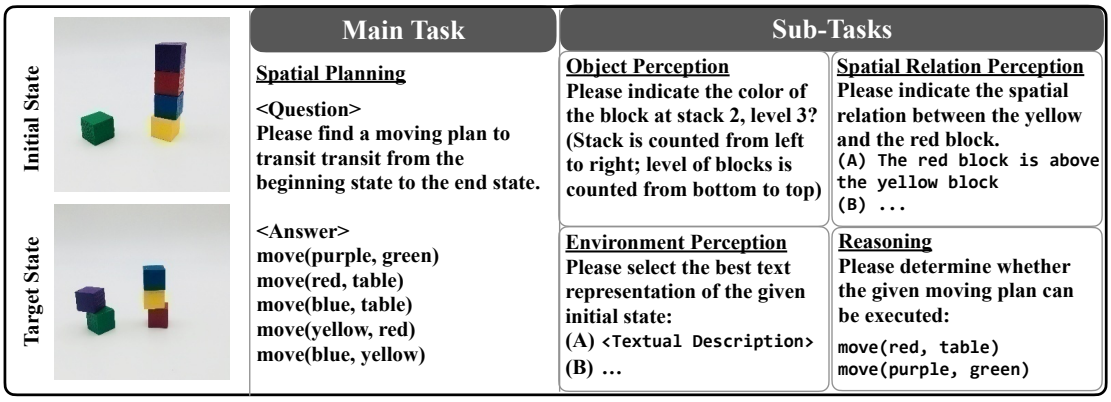


Figure 6.3: Overview of the *Blocks World* scenario.

- **T3** (Environment Perception) – Find the description that fits the environment;
- **T4** (Spatial Reasoning) – Determine the consequence of a given action series.

An example of input image and questions is demonstrated in Figure 6.2. Each task is equipped with progressive adjusted difficulty settings to comprehensively evaluate the model's capability. For **Main Task** and **T1-T3**, the difficulties are measured by the size of the map, ranging from 3x3 to 8x8, where a larger map introduces more challenges in correctly perceiving objects and planning accordingly. For task **T4**, since a longer path naturally introduces more challenges for reasoning, we adopt path length ranging from 1 to 9 as the difficulty measure.

6.3.3 The Blocks World Scenario

The *Blocks World* is a widely-adopted planning problem [236, 254, 255]. As depicted in Figure 6.3 left, in this scenario, the agent is given images containing sets of blocks in unique colors. These blocks are stacked vertically, forming multiple stacks on the table. The agent is asked to turn the blocks from initial state to target state through a series of moving actions. For each action, the agent can only move the top block of any stack, providing it is moved to either the table or the top of another stack.

Similarly, the main spatial planning task and the four sub-tasks are defined as follows:

• **Main Task** (Spatial Planning) – Form a moving plan to achieve the target state of block arrangement;

• **T1** (Single Object Perception) – Determine the color of the block at a specific position;

• **T2** (Spatial Relation Perception) – Determine the spatial relation between two blocks;

• **T3** (Environment Perception) – Find the text representation that fits the visual environment;

• **T4** (Spatial Reasoning) – Determine the consequence of a given moving plan.

An example of input image and questions is demonstrated in Figure 6.3. Similar to *Maze Navigation*, each task is equipped with progressive adjusted difficulty. Specifically, in **Main Task** and **T4**, the difficulties are measured by the number of actions involved, ranging from 1 to 7, which quantifies the complexity of the action plan. On the other hand, for tasks **T1**-**T3**, which focus on perception, the difficulty is measured by the number of blocks presented in the image, ranging from 3 to 5.

6.3.4 The Collision and Google Map Scenario

We further experiment in a more dynamic (the collision scenario) and realistic (the Google map scenario) settings to explore the capabilities of models in more challenging cases. The input of these two scenarios are shown in the right panels of Figure 6.1.

• **Collision Scenario** In this scenario, the input map is similar to *Maze Navigation* scenario. However, in this case, the player is moving in an environment where a car is also present and moving. Given the moving information (speed, direction) of the player

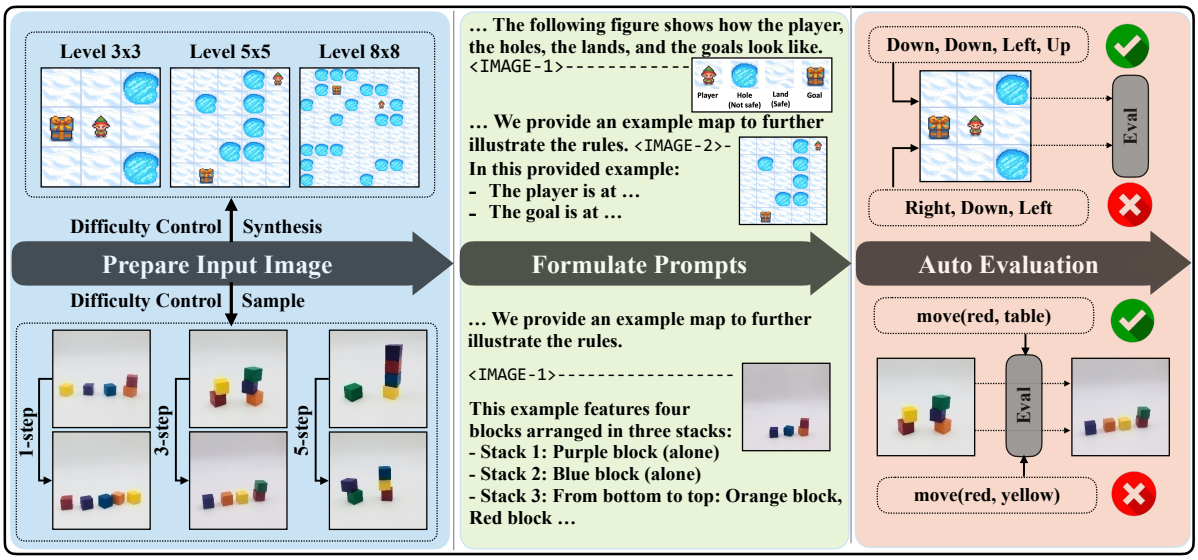


Figure 6.4: Benchmark creation process. **Left:** We prepare input images that fulfill the task requirements with different difficulties. **Mid:** We formulate input prompts consisting of interleaved texts and images. **Right:** We develop automatic evaluation process for each task.

and the car, the model needs to determine the time the player and the car reaches the goal, and then determine if there is a collision danger here.

- **Google Map Scenario** In this scenario, the input image is a real Google map depicting the streets and avenues in New York City. The starting location and the goal are randomly chosen crossroads in the map and are marked on the image. The goal of the model is to find a path from the starting location to the goal. The model needs to output a path described by directions (north, east, west, south) and the number of blocks to traverse (e.g., head north for 2 blocks, then head east for 3 blocks).

6.3.5 Benchmark Creation and Quality Control

In Figure 6.4, we demonstrate the general process for benchmark creation and quality control. We use the *Blocks World* and *Maze Navigation* scenarios as examples.

First, in the left panel of Figure 6.4, we prepare the input images used for each task

and scenario. In the *Maze Navigation* scenario, we generate input maps using the OpenAI Gym package [253], with modifications to ensure that the positions of the player, the goal, and the holes are all randomly generated. In the *Blocks World* scenario, we sample pairs of images from the BIRD dataset [255], ensuring there is at least one viable plan to move the blocks from the initial state to the target state. The images are prepared conditional on different levels of difficulty.

Second, as shown in the center of Figure 6.4, we formulate input prompts for each task. The prompt consists of interleaved text and images to provide sufficient information. For example, for *Maze Navigation*, we include images to show the appearance of elements in the map and provide example maps to better illustrate how the models should interpret the map. We invite native speakers to refine the prompts so that they accurately describe the task requirements.

Finally, in the right panel of Figure 6.4, we evaluate the performance of MLLMs. It is worth noting that the answer for each task is often not unique. For example, in the *Blocks World* scenario, there can be many ways to move the blocks to reach the target state. As such, we develop scripts to accurately evaluate each individual answer.

In addition to the steps above, some tasks require extra steps to ensure meaningful questions, candidates, and answers. For example, in T4 of the *Blocks World* scenario, the input actions must cover various valid/invalid movements.

6.4 Experiments

In this section, we present evaluation results of state-of-the-art MLLMs under our main tasks and sub-tasks. Our goal is to answer the following research questions: ❶ How well can state-of-the-art MLLMs perform in the visual spatial planning tasks? ❷ What are the bottleneck capabilities that limit the MLLMs in visual spatial planning tasks?

6.4.1 Baselines

We evaluate representative MLLMs including both private and open-source models.

For *private models*, we include their different versions to observe how their abilities involved and what is the state-of-the-art performance. We cover the following *private models*: ① Gemini [18, 199] has demonstrated remarkable capabilities in image understanding and reasoning. We adopt Gemini-1.0 (`Gemini-1.0-pro-vision`) and Gemini-2.0 (`Gemini-2.0-flash`) in our experiments. ② Claude [256, 257] is a family of MLLMs strong at advanced reasoning and vision analysis. We adopt Claude-3 (`claude-3-sonnet-20240229`) and Claude-3.7 (`claude-3-7-sonnet-20250219`) in the experiments. ③ GPT [197,258] inherent strong text understanding capabilities from text-only models and is equipped with vision capabilities. We use GPT-4V (`turbo-2024-04-09`) and GPT-4o (`gpt-4o-2024-05-13`) for evaluation.

Besides, we cover the following *open-source models*: ④ LLaVA-Next [198] demonstrates strong reasoning abilities comparable to private models. We adopt LLAVA-V1.6-VICUNA-7B for evaluation. ⑤ InternLM-XComposer2 [259] shows strong ability to understand free-form text-image composition. We adopt `internlm-xcomposer2-7b` and `internlm-xcomposer2-vl-7b`, with the former focusing on general text-image composition and the latter focusing on VL benchmarks. ⑥ SPHINX [260] unfreezes the LM during pre-training to enhance cross-model alignment. We adopt `SPHINX-v2-1k`. ⑦ LLaMA-3.2 [261] exceeds on several image understanding tasks. We adopt `llama-3.2-90B-Vision` in our experiments. ⑧ Pixtral [262] is a recent MLLM that demonstrates strong performance across a series of multimodal tasks. We adopt `pixtral-12b-2409`. ⑨ Qwen [55] demonstrates strong object localization and visual recognition in several vision-langauge tasks. We adopt `Qwen2.5-VL-7B-Instruct`. We use the public released checkpoints and codes.

Table 6.2: Zero-shot success rates for the spatial planning task at various difficulty levels. *Maze Navigation* difficulty levels represent the maze's square grid length. *Blocks World* difficulty levels correspond to minimum steps to a solution. Results better than 30% are **bolded**.

	Maze Navigation						Blocks World				Collision	Goo Map	Overall
Difficulty level	**3**	**4**	**5**	**6**	**7**	**8**	**1**	**3**	**5**	**7**			
Gemini-1.0 [199]	**0.31**	0.26	0.15	0.06	0.14	0.10	0.10	0.14	0.00	0.01	0.13	0.00	0.1167
Gemini-2.0 [18]	**0.93**	**0.81**	**0.62**	**0.41**	0.30	0.27	**0.85**	**0.49**	0.18	0.07	**0.44**	0.07	**0.4533**
Claude-3 [256]	**0.52**	**0.33**	0.16	0.15	0.16	0.09	0.12	0.03	0.00	0.00	0.18	0.02	0.1467
Claude-3.7 [257]	**0.71**	**0.56**	**0.56**	0.27	0.18	0.28	**0.89**	**0.57**	**0.33**	**0.44**	**0.49**	0.11	**0.4492**
GPT-4V [197]	**0.55**	**0.36**	0.27	0.13	0.17	0.10	**0.50**	0.17	0.03	0.00	0.24	0.02	0.2117
GPT-4o [258]	**0.68**	**0.58**	**0.35**	0.24	0.18	0.23	**0.71**	**0.33**	0.12	0.03	0.16	0.04	**0.3042**
LLaVA-Next [198]	0.03	0.03	0.02	0.08	0.09	0.04	0.04	0.01	0.00	0.00	0.02	0.00	0.0300
InternLM [259]	0.27	0.16	0.06	0.05	0.04	0.07	0.10	0.03	0.00	0.00	0.25	0.00	0.0858
InternLM-VL [259]	0.15	0.14	0.08	0.04	0.02	0.05	0.02	0.00	0.00	0.00	0.22	0.01	0.0600
SPHINX [260]	0.11	0.08	0.05	0.02	0.04	0.03	0.07	0.06	0.01	0.00	0.04	0.00	0.0425
LLaMA-3.2 [261]	0.23	0.18	0.16	0.08	0.08	0.10	0.05	0.00	0.00	0.00	0.14	0.00	0.0850
Pixtral [262]	**0.32**	0.20	0.17	0.10	0.06	0.06	0.21	0.11	0.01	0.00	0.22	0.03	0.1242
Qwen-2.5 [55]	**0.45**	**0.33**	0.20	0.09	0.15	0.08	0.06	0.03	0.00	0.01	**0.45**	0.06	0.1592

6.4.2 Main Task (Spatial Planning) Evaluation

First, we present the main task results for the four scenarios, which reflect the general spatial planning capabilities of MLLMs. All the evaluation in this section is conducted under zero-shot setting. Evaluations with in-context learning and fine-tuning are presented in Sections 6.4.5 and 6.4.6.

The performance is demonstrated in Table 6.2. We also present difficulty levels in the table, which is measured by the size of the map (3 represents 3x3 maps) in *Maze Navigation* and by the minimum number of steps in *Blocks World*. From the table, we summarize our findings as follows:

VSP highlights both the advancements and the challenges in MLLMs' spatial planning capabilities. Notably, private MLLMs demonstrate a clear trajectory of improvement, with earlier versions performing worse and later iterations showing advancements - though still exhibiting deficiencies. In fact, even some of the most capable models frequently make mistakes on tasks such as navigating a 7x7 map or executing 3-step block-moving tasks, which are often trivial for humans. Meanwhile, open-source

models face even greater difficulties and rarely succeed in these tasks. Overall, VSP demonstrates that while MLLMs have made progress in spatial planning, they still exhibit considerable room for improvement.

MLLMs face significant difficulties with spatial planning in dynamic and realistic environments. Based on the experimental results, most models perform worse when tested in dynamic (*Collision*) and realistic (*Google Map*) settings. We identify two major reasons: First, the tasks in *Collision* are complex and typically require multiple capabilities. For example, to assess collision danger, the model must locate both the player and the car on the map, calculate the time needed for the player to reach the goal, and determine if the car will hit the player during that time. This poses significant challenges for the models. Second, the inputs in the *Google Map* are intricate and contain a series of irrelevant symbols, making it difficult for the model to accurately interpret the map. The models' performance in these environments suggests that current models are not yet equipped to handle spatial reasoning in such complex environments. Consequently, we focus on the *Maze Navigation* and *Blocks World* scenarios in the following experiments to diagnose the models' hidden weaknesses in VSP tasks.

Quick performance decay as difficulty increases. We observe a significant drop in the success rates of MLLMs as task difficulty escalates. For example, GPT-4V may achieve a success rate of over 50% on 3x3 size maps, but this plummets to just 10% on 8x8 maps. Analyzing the impact of increased difficulty, we identify two major challenges for the models: First, increasing size of the map in *Maze Navigation* scenario could make it difficult for the model to accurately *perceive* the positions of elements within the map. Second, the increase in both map size and the number of steps required for moving blocks heightens the challenge for the model to *reason* deeply through the entire path and devise a complete, viable solution. In the following experiments, we focus on these two factors and provide in-depth diagnoses with subsequent tasks.

Table 6.3: Decomposed Capability Analysis. Results better than 70% are **bolded**. Each task consists of test with different difficulties.

	Maze Navigation				Blocks World			
Task	**T1**	**T2**	**T3**	**T4**	**T1**	**T2**	**T3**	**T4**
Random Guess	0.5	0.25	0.25	0.5	0.17	0.25	0.25	0.5
Gemini-1.0 [199]	0.58	0.56	0.33	0.49	**0.86**	0.51	0.54	0.55
Gemini-2.0 [18]	0.66	**0.91**	**0.96**	**0.80**	**0.95**	**0.96**	**0.99**	**0.76**
Claude-3 [256]	0.45	0.67	0.32	0.61	0.43	0.53	0.49	0.66
Claude-3.7 [257]	**0.72**	**0.88**	**0.97**	**0.79**	**0.94**	**0.84**	**0.99**	**0.94**
GPT-4V [197]	0.56	0.27	0.46	0.56	**0.73**	**0.80**	**0.70**	**0.71**
GPT-4o [258]	0.58	0.67	0.58	**0.74**	**0.95**	**0.90**	**0.90**	**0.76**
LLaVA-Next [198]	0.49	0.27	0.21	0.54	0.22	0.21	0.24	0.55
InternLM [259]	0.48	0.27	0.29	0.58	0.25	0.32	0.26	0.53
InternLM-VL [259]	0.41	0.20	0.17	0.47	0.22	0.20	0.20	0.53
SPHINX [260]	0.56	0.28	0.32	0.59	0.24	0.33	0.27	0.58
LLaMA-3.2 [261]	0.53	0.29	0.33	0.57	0.28	0.36	0.29	0.56
Pixtral [262]	0.44	0.35	0.33	0.51	0.49	**0.72**	0.63	0.57
Qwen-2.5 [55]	0.58	0.62	0.31	0.50	0.53	0.46	**0.79**	0.56

Challenges in open-source models. Finally, we note that open-source models often face difficulties on these tasks. As will be analyzed in Section 6.4.3, their perception and reasoning capabilities present greater challenges. Besides, we note two other factors: ① Context length: Open-source models typically have shorter context windows compared to private models. Thus, these models may not have enough capacity to understand the complete inputs. For example, llava-v1.6-vicuna-7b is trained with a context window of 2048 tokens, while each image consumes 576 tokens. As such, when fed with multiple images and long texts in our tasks, the total token length may surpass training, resulting in poor performance. ② Multiple image input: Our tasks require the model to understand multiple images interleaved with text inputs, whereas some open-source models are only trained with single-image inputs, with the image positioned at the start of the input. To further explore their potential, we assess their performance after fine-tuning in Section 6.4.6. Also, we note that recent open-source MLLMs often avoid these

two issues, which may contribute to the improved performance observed in Table 6.2.

6.4.3 Diagnose the Perception and Reasoning

From the previous observation, we identify that spatial *perception* and *reasoning* could be two important capabilities for an agent to successfully perform visual spatial planning. Next, we evaluate these two abilities through the remaining tasks T1-T4. Similar to previous setting, all the evaluation is conducted under zero-shot settings.

Table 6.3 shows the decomposed capability results. Strong private models like Gemini-2.0 and Claude-3.7 show better performance across these tasks, demonstrating solid perception and reasoning capabilities, which explain their superior performance in the main tasks. However, they still exhibit some deficiencies, such as perceiving the location of a single object in *Maze* T1, highlighting a critical area for improvement. Furthermore, the performance of many open-source models is mostly close to random guessing on these tasks, indicating significant gaps compared to private models. One caveat is that while T4 focuses on reasoning capabilities, it still relies on the perception capabilities because the input still contains images. We perform further analysis to disentangle these two abilities in Section 6.4.4.

6.4.4 The Effects of Visual Input

Previous analysis shows that even strong MLLMs have clear deficiencies in various aspects of visual spatial planning. In this study, we focus on disentangling the effects of perception and reasoning by exploring the performance gain assuming the model had perfect perception.

The key strategy here is to create a scenario where the model has acquired all necessary information that would typically be obtained through visual perception. To this

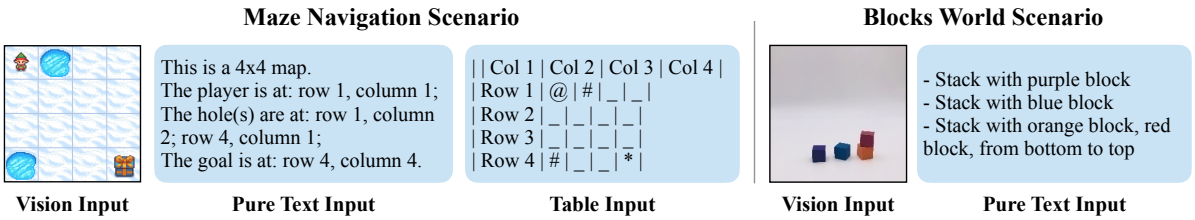


Figure 6.5: The visual and corresponding textual inputs.

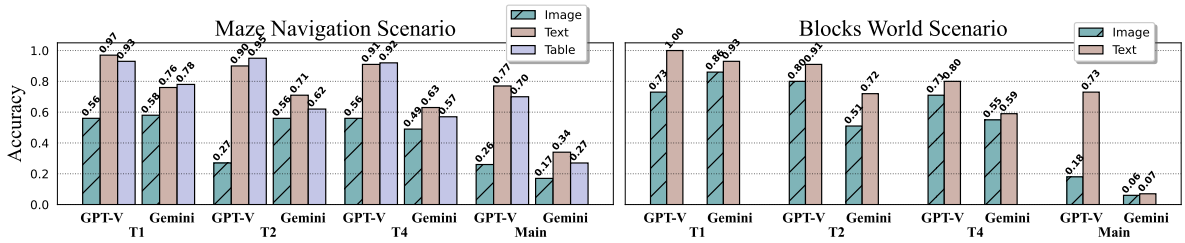


Figure 6.6: Result with the visual/textual input. When environment is described by text instead of image, the performance increases vastly.

end, for every input image, we produce the corresponding textual inputs and replace those images, as shown in Figure 6.5. For the *Maze Navigation* scenario, we use either pure text descriptions or tables to depict the image. For the *Blocks World* scenario, we use pure text descriptions. We do not use tables for the *Blocks World* scenario because the number of blocks in each horizontal stack is usually unequal, making it difficult to form a complete table.

The results are shown in Figure 6.6. We observe a clear performance improvement when using textual input across every task. This suggests image perception presents significant challenges for MLLMs, and poor perception ability is a key factor in the inferior performance observed in previous tasks. Meanwhile, note that even with textual input, Gemini still cannot achieve decent performance on tasks that require reasoning. This indicates its deficiencies in reasoning.

Table 6.4: Effects of providing in-context examples.

	Maze Navigation					Blocks World				
Task	**T1**	**T2**	**T3**	**T4**	**Main**	**T1**	**T2**	**T3**	**T4**	**Main**
Gemini, 0-shot	0.58	0.56	0.33	0.49	0.17	0.86	0.51	0.54	0.55	0.03
Gemini, 1-shot	0.50	0.66	0.31	0.48	0.20	0.91	0.68	0.71	0.59	0.03
Gemini, 2-shot	0.53	0.68	0.31	0.51	0.21	0.90	0.76	0.70	0.61	0.03
Gemini, 4-shot	0.53	0.67	0.35	0.53	0.19	0.91	0.64	0.69	0.62	0.06
GPT-4V, 0-shot	0.56	0.27	0.46	0.56	0.26	0.73	0.80	0.70	0.71	0.10
GPT-4V, 1-shot	0.55	0.50	0.47	0.57	0.28	0.89	0.84	0.94	0.73	0.11
GPT-4V, 2-shot	0.55	0.63	0.50	0.56	0.30	0.90	0.83	0.95	0.71	0.16
GPT-4V, 4-shot	0.54	0.69	0.54	0.56	0.29	0.90	0.79	0.96	0.73	-

6.4.5 In-context Learning in Visual Spatial Planning

In-context learning is a widely-adopted method to enhance LM's reasoning ability [196]. In this analysis, we study if it boosts visual spatial planning capabilities. We included varying numbers of examples for Gemini-1.0 and GPT-4V and show results in Table 6.4. There are two key observations: First, in-context examples make some potential contributions, but they are not significant. The examples only benefits in several sparse cases, such as **T2** in *Maze Navigation* and **T3** in *Blocks World*. Second, scaling examples generally does not help, as illustrated by the saturated performance in each task.

6.4.6 Fine-tuning in VSP Tasks

Finally, we evaluate the open-source model through dedicated training. We performed LoRA fine-tuning on `llava-v1.6-vicuna-7b` and `internlm-xcomposer2-7b`. The models are trained on 10k data points (image-text pairs) for each task and scenario. We use the default hyperparameters provided in the official repository. The results, shown in Table 6.5, demonstrate clear performance improvements for both models across a series of tasks, highlighting their potential in spatial planning. Additionally, we observe

Table 6.5: Fine-tuning results for open-source models.

		Maze Navigation					Blocks of World				
Model	Setting	**T1**	**T2**	**T3**	**T4**	**Main**	**T1**	**T2**	**T3**	**T4**	**Main**
LLaVA	zero-shot	0.49	0.27	0.21	0.54	0.05	0.22	0.21	0.24	0.55	0.01
	fine-tune	0.53	0.99	0.51	0.93	0.60	1.00	1.00	1.00	1.00	0.97
InternLM	zero-shot	0.48	0.27	0.29	0.58	0.11	0.25	0.32	0.26	0.53	0.00
	fine-tune	0.52	0.59	0.91	0.59	0.17	0.29	0.44	0.69	0.62	0.09

that LLaVA shows greater improvement compared to InternLM, suggesting that different model architectures may exhibit varying levels of efficacy in spatial planning capabilities.

6.5 Conclusion

This chapter presents VSP, a benchmark measuring and diagnosing the visual spatial planning capabilities in MLLMs. VSP quantifies the model's performance through a series of carefully designed tasks, with main tasks testing general spatial planning abilities and sub-tasks diagnosing individual capabilities needed for the main task. VSP shows that current MLLMs fail to generate effective plans for even simple spatial planning tasks, and further analyses expose their bottlenecks in spatial perception and reasoning abilities. This chapter illuminates future directions for improving MLLMs' abilities in spatial planning.

Chapter 7

Enhancing Visual Understanding in MLLMs

7.1 Introduction

Multimodal large language models (MLLMs) have shown exceptional reasoning capabilities on complex tasks that require multimodal reasoning. However, recent studies [263,264] reveal a reasoning imbalance: these models often rely heavily on textual cues while under-exploiting visual information when generating answers. This over-reliance on text leads to suboptimal results on tasks that require genuine visual reasoning. We refer to this phenomenon as the *modality gap*. As exemplified in Figure 7.1, when critical information present in the visual modality is removed from the text, MLLMs fail to answer questions that could have been correctly answered when the full text was provided, highlighting their insufficient visual reasoning.

To understand the origin of this imbalance, we focus on the training recipes of current MLLMs. An important observation is that many training samples contain overlapping information across the textual and visual modalities. In such cases, it may be easier for

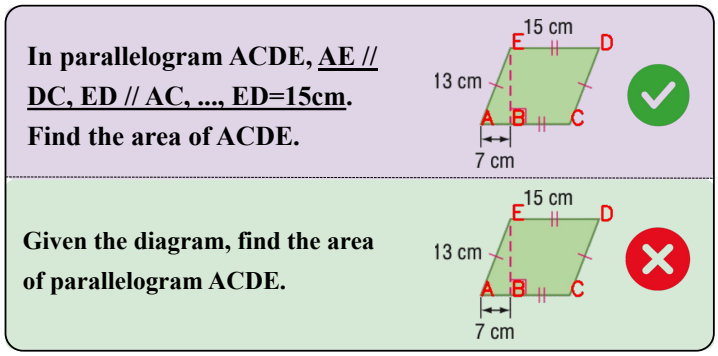


Figure 7.1: Current MLLMs exhibit an imbalance between visual and textual reasoning. When information present in the visual modality is removed from the text, the MLLM fails to answer the question.

MLLMs to rely on the already complete textual information rather than engage in visual reasoning. We hypothesize that this training process largely contributes to the modality gap. Our preliminary evidence supports this view: under standard training setups, the gap between text and vision centric performance widens over time, underscoring the need for more balanced training strategies.

Building on these insights, our goal is to identify improved training recipes for MLLMs that jointly ❶ ensure the effective use of visual information and ❷ maintain or enhance overall reasoning ability in the targeted domains. We approach this problem from two perspectives: *data* and *loss*. From the data perspective, we consider vision-centric and text-centric data in supporting balanced multimodal reasoning and explore how simple data mixing and carefully designed curriculum strategies can leverage the strengths of both modalities. From the loss perspective, we propose a KL-based self-distillation objective that transfers reasoning competence from full-text to partial-text inputs, preserving general reasoning performance while strengthening visual grounding. Our key contributions are as follows:

- We establish a diagnostic that reveals a consistent discrepancy between text-centric and vision-centric performance across a range of public and private MLLMs at

different scales. Furthermore, we show that current training setups *widen* this discrepancy, underscoring the need for targeted recipes beyond naive RL.

- We propose improved RL training recipes from the perspectives of *data* and *loss*, designed to preserve reasoning competence in the textual modality while reducing the reasoning gap across modalities.

7.2 Background and Prior Work

7.2.1 Multimodal Large Language Models

Multimodal Large Language Models (MLLMs) have emerged as powerful tools that integrate visual and textual information to perform a wide range of tasks, including image captioning, visual question answering, and geometric reasoning. Notable MLLMs include Qwen2.5-VL series [55, 265, 266], VL-Rethinker series [49], MiniCPM series [267], InternVL3.5 series [268], Kimi-VL series [269], Gemma series [270], GPT-5 [17] and Gemini [57]. These models typically employ a combination of pre-trained vision encoders and large language models, fine-tuned on multimodal datasets to enhance their understanding and generation capabilities across both modalities.

7.2.2 Visual Reasoning in MLLMs

Visual reasoning is a critical capability for MLLMs, enabling them to interpret and reason about visual content in conjunction with textual information. Recent studies like Mathverse [263] have highlighted the challenges MLLMs face in effectively utilizing visual information, often defaulting to text-based cues. This has led to the identification of the modality gap, where models perform significantly better on text-centric tasks compared to vision-centric ones.

Table 7.1: Base model performance.

Model	Dataset	Text	Vision	Avg	Gap
Qwen2.5-VL 3B	PGPS9K	*0.2397*	*0.1812*	*0.2105*	*0.0585*
	MathVerse	*0.3568*	*0.2866*	*0.3147*	*0.0702*
Qwen2.5-VL 7B	PGPS9K	*0.3775*	*0.2998*	*0.3387*	*0.0777*
	MathVerse	*0.5501*	*0.4576*	*0.5110*	*0.0925*
MiniCPM-V-4	PGPS9K	*0.3470*	*0.3020*	*0.3245*	*0.0450*
	MathVerse	*0.4435*	*0.3721*	*0.4007*	*0.0714*
Gemma-3-4b-it	PGPS9K	*0.4050*	*0.2612*	*0.3559*	*0.1438*
	MathVerse	*0.4236*	*0.3350*	*0.3705*	*0.0886*
Kimi-VL-A3B	PGPS9K	*0.4057*	*0.3132*	*0.3595*	*0.0925*
	MathVerse	*0.5791*	*0.4827*	*0.5123*	*0.0964*
VL-Rethinker 7B	PGPS9K	*0.4045*	*0.3605*	*0.3825*	*0.0440*
	MathVerse	*0.6542*	*0.5728*	*0.6053*	*0.0814*
InternVL3.5 8B	PGPS9K	*0.5218*	*0.3965*	*0.4592*	*0.1253*
	MathVerse	*0.6668*	*0.5419*	*0.5919*	*0.1249*
Qwen3-VL	PGPS9K	*0.6970*	*0.6699*	*0.6835*	*0.0271*
	MathVerse	*0.6406*	*0.6089*	*0.6167*	*0.0317*
GPT-5[2]	PGPS9K	*0.9400*	*0.8000*	*0.8700*	*0.1400*
	MathVerse	*0.7667*	*0.6333*	*0.7000*	*0.1334*
Gemini 2.5 Flash[2]	PGPS9K	*0.9200*	*0.7400*	*0.8300*	*0.1800*
	MathVerse	*0.8696*	*0.7778*	*0.8200*	*0.0918*

To address this issue, various methods have been proposed, including specialized training datasets [19, 271, 272], model architecture design [273, 274], and loss functions that encourage visual attention [264, 275, 276]. In this chapter, we build upon these foundations and explore RL-based methods to enhance visual reasoning, mitigating the modality gap.

7.3 Modality Gap in MLLMs

We begin by quantifying the modality gap across a range of open-source and commercial MLLMs. To illustrate this gap, we consider two kinds of data:

• $\mathcal{D}_1$: **Text-centric.** All necessary information is contained within the provided text, and the MLLM can solve the problem through textual reasoning.

[2]Due to API costs, the results are evaluated on a subset of 50 test samples.

Table 7.2: Standard RL training on PGPS9K Results.

Model	Text-centric	Vision-centric	Avg
Qwen2.5-VL 3B	*0.2397*	*0.1812*	*0.2105*
Qwen2.5-VL 3B $\mathcal{D}_1$	*0.6208*	*0.4480*	*0.5249*
Qwen2.5-VL 3B $\mathcal{D}_2$	*0.5118*	*0.5050*	*0.5084*
Qwen2.5-VL 7B	*0.3775*	*0.2998*	*0.3387*
Qwen2.5-VL 7B $\mathcal{D}_1$	*0.7422*	*0.5377*	*0.6400*
Qwen2.5-VL 7B $\mathcal{D}_2$	*0.6342*	*0.6040*	*0.6191*

• $\mathcal{D}_2$: **Vision-centric.** Some necessary information is present in the image but not in the text, requiring the MLLM to perform visual reasoning to successfully solve the problem.

To construct $\mathcal{D}_1$ and $\mathcal{D}_2$, we draw upon two challenging visual reasoning datasets: PGPS9K [277] and MathVerse [263]. In PGPS9K, each question consists of a textual condition and a question statement, accompanied by a fully annotated figure that specifies entities and their relations. Accordingly, we define $\mathcal{D}_1$ as the setting where both the image and text provide complete information, and $\mathcal{D}_2$ as the setting where information present in the image has been removed from the text. For MathVerse, following prior work, we define $\mathcal{D}_1$ and $\mathcal{D}_2$ subsets to focus respectively on text (*Text-Dominant* and *Text-Lite* subsets) and vision (*Vision-Intensive*, *Vision-Dominant*, and *Vision-Only* subsets) reasoning capabilities.

Metrics We report the **text-centric** and **vision-centric** performance measuring on $\mathcal{D}_1$ and $\mathcal{D}_2$. in addition, we report the **overall** performance as the average accuracy across $\mathcal{D}_1$ and $\mathcal{D}_2$.

Direct Inference Results We begin by evaluating a series of off-the-shelf MLLMs. The results are summarized in Table 7.1. Across both the PGPS9K and MathVerse datasets, we observe a consistent modality gap: text-centric performance is consistently higher than vision-centric performance across various open-source and commercial mod-

els of different sizes. Moreover, stronger MLLMs tend to exhibit a larger performance gap. This discrepancy underscores the need for targeted strategies to enhance the visual reasoning capabilities of MLLMs.

Effect of Standard RL Training Next, we explore how standard training influences the modality gap. In this experiment, we apply DAPO [278] to fine-tune Qwen2.5-VL (3B and 7B) under both $\mathcal{D}_1$ and $\mathcal{D}_2$ settings from the PGPS9K training set. Note that all figures in $\mathcal{D}_1$ and $\mathcal{D}_2$ have their entities, relations, and other geometric properties explicitly annotated. Thus, the model can always obtain complete information related to the question from the image. As shown in Table 7.2, training on $\mathcal{D}_1$ primarily improves text-centric performance but enlarges the modality gap as training progresses, whereas training on $\mathcal{D}_2$ strengthens vision-centric performance and narrows the gap, though at the expense of overall accuracy.

Moreover, as shown in Figure 7.2, during standard training on $\mathcal{D}_1$, the modality gap progressively increases with training steps. These observations indicate that the standard training recipe is insufficient to resolve the modality gap in MLLMs, highlighting the need for more nuanced training strategies.

7.4 Mitigating Modality Gap

In this section, we investigate an improved RL training recipe from two complementary perspectives to enhance the visual reasoning ability of MLLMs:

Data We explore two training strategies: ① *mixed training*, which combines $\mathcal{D}_1$ (full-text) and $\mathcal{D}_2$ (partial-text) samples to expose models to both text- and vision-centric inputs; and ② *curriculum training*, which first trains on $\mathcal{D}_1$ to consolidate reasoning

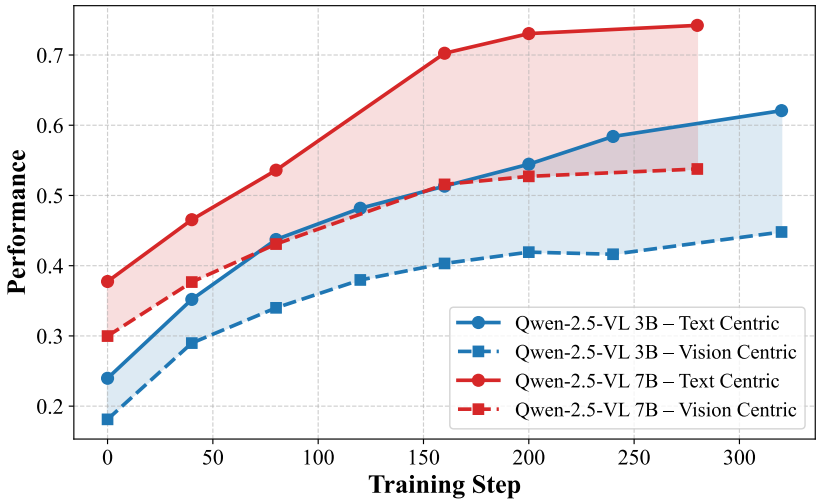


Figure 7.2: Standard training recipe widens modality gap.

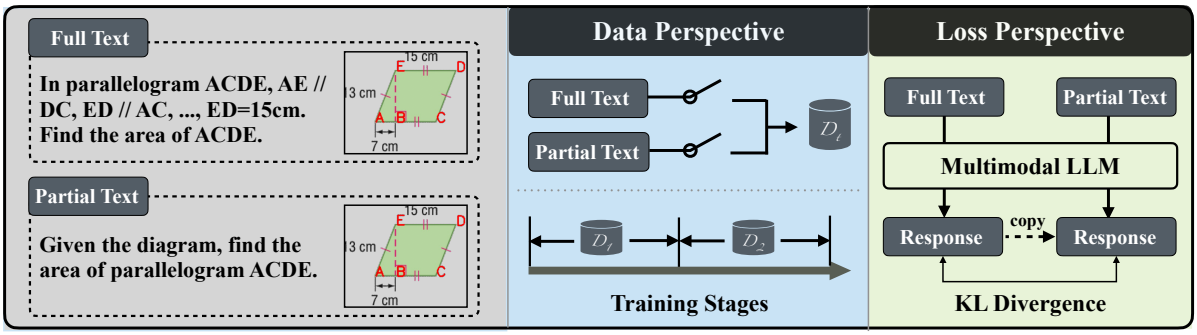


Figure 7.3: We consider two types of data: ❶ both text and image contain complete information, referred to as *full text*; and ❷ the text omits information already present in the image, referred to as *partial text*. We then analyze better training recipe from both data and loss perspectives.

under textual guidance, and then shifts to $\mathcal{D}_2$ to strengthen image-based reasoning and reduce shortcut reliance.

Loss We introduce a KL-based self-distillation loss to align the model's output distribution on $\mathcal{D}_2$ with that on $\mathcal{D}_1$, thereby preserving core reasoning ability while enhancing visual understanding.

Table 7.3: Data mixing and curriculum training results

	PGPS9K			MathVerse		
Training Strategy	**Text**	**Vision**	**Avg**	**Text**	**Vision**	**Avg**
Qwen2.5-VL 3B						
Mixed training	**0.6162**	**0.5683**	**0.5923**	0.4724	0.4202	0.4411
Curriculum Stage 1 ($\mathcal{D}_1$)	0.5840	0.4163	0.5002	**0.4937**	0.4317	0.4565
Curriculum Stage 2 ($\mathcal{D}_1 \rightarrow \mathcal{D}_2$)	0.5978	0.5400	0.5689	0.4902	**0.4369**	**0.4582**
Qwen2.5-VL 7B						
Mixed training	0.7010	0.6565	0.6788	0.5493	0.4833	0.5097
Curriculum Stage 1 ($\mathcal{D}_1$)	**0.7305**	0.5272	0.6289	0.4995	0.4442	0.4663
Curriculum Stage 2 ($\mathcal{D}_1 \rightarrow \mathcal{D}_2$)	0.7060	**0.6630**	**0.6845**	**0.5695**	**0.5060**	**0.5314**

7.4.1 Data Perspective

Implementation In this section, we compare the *mixed training* and *curriculum training* strategies. For mixed training settings, we stop training until the DAPO training parameter `num_gen_batches` reaches 10, for curriculum settings, we use the same total training steps but split evenly between the two stages, in which Stage 1 trains on $\mathcal{D}_1$ and Stage 2 trains on $\mathcal{D}_2$.

Result As summarized in Table 7.3, curriculum training generally matches or surpasses mixed-data training in both in-distribution (PGPS9K) and out-of-distribution (MathVerse) evaluations. Intuitively, Stage 1 on $\mathcal{D}_1$ consolidates general reasoning and solution formatting under rich textual guidance; Stage 2 on $\mathcal{D}_2$ then compels stronger visual grounding. This two-stage approach effectively improves both text-centric and vision-centric performance.

7.4.2 Loss Perspective

Implementation We introduce a *contrastive self-distillation KL loss* to transfer reasoning ability from inputs with *full text condition* to those with *partial text condition*. For each paired prompt that shares the same image and question, $(x^{(1)}, x^{(2)}) \in (\mathcal{D}_1, \mathcal{D}_2)$,

Table 7.4: Loss perspective results.

	PGPS9K			MathVerse		
Training Strategy	**Text**	**Vision**	**Avg**	**Text**	**Vision**	**Avg**
Qwen2.5-VL 3B						
Plain	0.5840	0.4163	0.5002	**0.4937**	**0.4317**	**0.4565**
w\ KL	0.5595	0.4647	0.5321	0.4819	0.4158	0.4423
w\ KL + Curriculum	**0.6122**	**0.5527**	**0.5825**	0.4708	0.4223	0.4417
Qwen2.5-VL 7B						
Plain	0.7305	0.5272	0.6289	0.4995	0.4442	0.4663
w\ KL	0.7328	0.5430	0.6379	**0.5700**	**0.4879**	**0.5207**
w\ KL + Curriculum	**0.7342**	**0.6787**	**0.7065**	0.5376	0.4855	0.5063

we first sample a sequence $\hat{\mathbf{y}}$ from $\pi_\theta(\cdot \mid x^{(1)})$. When this sequence is verified correct, we align next-token distributions under the same prefix $\hat{y}_{<t}$ by defining $p_t := \pi_\theta(\cdot \mid \hat{y}_{<t}, x^{(2)})$ and $q_t := \text{stopgrad}[\pi_\theta(\cdot \mid \hat{y}_{<t}, x^{(1)})]$. We then minimize a forward KL averaged over time:

$$\mathcal{L}_{\text{cKL}}(\theta) = \frac{1}{T} \sum_{t=1}^{T} \text{KL}\left(p_t \,\|\, q_t\right). \tag{7.1}$$

The forward KL encourages the model's response distribution under partial-text inputs to *cover* the high-confidence region of its own distribution under full-text inputs. In practice, this KL loss is computed for all rollouts (without DAPO roll out batch group filtering) and added to the RL objective with weight $\alpha = 0.01$, providing a dense learning signal and helping maintain the overall training loss optimization process stable. After the contrastive KL loss has stabilized, the model is further fine-tuned on $\mathcal{D}_2$ to enhance its visual reasoning ability.

Result We compare three training strategies in Table 7.4: ❶ Plain RL on $\mathcal{D}_1$, ❷ with KL, i.e., adding the contrastive KL loss, and ❸ with KL + Curriculum, where the KL-trained model is subsequently fine-tuned on $\mathcal{D}_2$. From the in-distribution results on PGPS9K, both KL and KL + Curriculum consistently outperform the plain baseline,

confirming that the KL term effectively transfers reasoning ability and stabilizes the optimization process. However, on the out-of-distribution dataset MathVerse, the improvements are less consistent, likely due to annotation and representation mismatches between the datasets. Specifically, PGPS9K provides explicit geometric cues, whereas MathVerse often omits such markings, weakening cross-domain transfer. Overall, the KL loss enhances general reasoning ability, while the subsequent curriculum fine-tuning slightly degrades out-of-distribution performance, reflecting the impact of differing annotation styles across datasets.

7.5 Conclusion

We present a systematic study on reducing the modality gap of MLLMs through reinforcement learning. Our experiments show that curriculum training effectively balances text-centric and vision-centric reasoning, and a KL-based self-distillation loss transfers reasoning competence from text-rich to vision-centric inputs. Together, these findings yield practical guidance: favor curriculum + contrastive KL to build MLLMs with stronger and more balanced visual reasoning capabilities.

Chapter 8

Conclusion and Future Directions

8.1 Conclusion

This dissertation studies the dual role of text in vision-language models. On the generation side, text serves as a powerful control interface that allows users to specify visual intent through natural language. On the understanding side, text can also become a shortcut that causes multimodal models to underuse visual inputs. Through these two perspectives, this dissertation examines how text enables controllability in visual generation and how excessive reliance on text reveals limitations in multimodal understanding.

The first part of this dissertation focuses on text-controlled visual generation and editing. I study instruction-based image retouching with professional editing tools, where an MLLM agent translates user requests into executable operations and learns from a generalist reward model. I then study text-guided camera control in video generation, where natural-language descriptions specify complex camera motions without requiring explicit camera trajectories. I further investigate diffusion-based image generation and editing, showing how text embeddings can be manipulated for visual attribute control and how region-aware attention guidance can improve prompt-image fidelity. Together,

these works demonstrate that text can provide an expressive and flexible interface for controlling visual content across editing tools, image diffusion models, and video generation systems.

The second part of this dissertation focuses on over-reliance on text in multimodal understanding. I study this issue through spatial reasoning, visual planning, and geometry reasoning tasks, where essential information must be inferred from images rather than textual descriptions. These studies show that strong language reasoning does not necessarily imply strong visual reasoning. Current MLLMs may perform well when textual cues are sufficient, yet struggle when they must rely on visual evidence. I further explore training strategies that reduce excessive textual hints and encourage models to reason more directly from images.

Overall, this dissertation contributes methods for building vision-language models that use text effectively while maintaining strong visual grounding. Text can make visual systems more controllable and accessible, but it should not replace visual perception and reasoning. The studies in this dissertation suggest that future vision-language models should be designed not only to follow language instructions, but also to faithfully ground those instructions in visual content.

8.2 Future Directions

The findings in this dissertation suggest several directions for future research.

More flexible text-based control in visual generation. A first direction is to develop visual generative models that support more flexible and fine-grained control through natural language. Existing models can often follow high-level prompts, but reliable control over specific visual aspects remains difficult. Future systems should allow

users to control various features such as object layout, attributes, lighting, temporal dynamics, and editing operations with greater precision. Such control should also be robust to abstract, incomplete, or subjective instructions, since real users often describe visual intent in natural language rather than through formal specifications. Achieving this goal requires stronger grounding between text and visual representations, better mechanisms for identifying which parts of the visual content should change, and methods that preserve irrelevant content while modifying only the intended aspects.

Unified and simultaneous control across visual aspects. A second direction is to move beyond separate models or separate mechanisms for different kinds of control. Many current approaches treat attribute editing, layout control, and style control as independent problems. However, real visual creation often requires these forms of control simultaneously. For example, a user may want to change the style of an image, preserve the identity of a subject, adjust local lighting, and maintain a specific spatial composition at the same time. Similarly, video generation may require coordinated control over scene content, object motion, camera movement, and temporal consistency. Future work should develop unified frameworks that can integrate multiple control signals within a single model or system. Such systems would reduce the need for task-specific control modules and make text-guided visual generation more practical for complex creative workflows.

Balanced visual and textual understanding in MLLMs. A third direction is to build multimodal models that integrate visual and textual information in a more balanced way. Current MLLMs often benefit from strong language priors, but these priors can also dominate model behavior when visual evidence is necessary. Future research should study this imbalance across a broader range of tasks, including spatial reasoning, visual question answering, embodied planning, and user-interface reasoning. These set-

tings require different forms of visual grounding, from low-level perception to high-level relational and causal reasoning. A more complete understanding of text-vision imbalance will require benchmarks that separate what can be inferred from language priors from what must be inferred from images. Beyond diagnosis, this imbalance also calls for training and evaluation pipelines that discourage models from relying on textual shortcuts. This may involve reducing excessive textual descriptions during training, constructing examples where language priors conflict with visual evidence, and evaluating models under visual-only or visually critical settings. It may also require new reward models and diagnostic tools that explicitly measure whether model predictions are grounded in the image. These efforts can help move MLLMs toward stronger visual grounding and more reliable multimodal reasoning.

Together, these future directions point toward vision-language models that are more controllable in generation and more visually grounded in understanding. Such models should use text as a powerful interface, while preserving the ability to perceive, reason about, and act upon visual information directly.

Bibliography

[1] Google, "Introducing nano banana pro." `https://blog.google/innovation-and-ai/products/nano-banana-pro/`, 2025. Google product announcement.

[2] OpenAI, "Introducing chatgpt images 2.0." `https://openai.com/index/introducing-chatgpt-images-2-0/`, 2026. OpenAI product announcement.

[3] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, *Zero-shot text-to-image generation*, in *ICML*, 2021.

[4] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, *et. al.*, *Cogvideox: Text-to-video diffusion models with an expert transformer*, *arXiv preprint arXiv:2408.06072* (2024).

[5] Y. Guo, C. Yang, A. Rao, Z. Liang, Y. Wang, Y. Qiao, M. Agrawala, D. Lin, and B. Dai, *Animatediff: Animate your personalized text-to-image diffusion models without specific tuning*, *arXiv preprint arXiv:2307.04725* (2023).

[6] OpenAI, "Sora." https://openai.com/sora/, 2024.

[7] T. Seedance, D. Chen, L. Chen, X. Chen, Y. Chen, Z. Chen, Z. Chen, F. Cheng, T. Cheng, Y. Cheng, *et. al.*, *Seedance 2.0: Advancing video generation for world complexity*, *arXiv preprint arXiv:2604.14148* (2026).

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-resolution image synthesis with latent diffusion models*, in *CVPR*, 2022.

[9] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, *et. al.*, *Photorealistic text-to-image diffusion models with deep language understanding*, *arXiv preprint arXiv:2205.11487* (2022).

[10] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, *et. al.*, *Imagen video: High definition video generation with diffusion models*, *arXiv preprint arXiv:2210.02303* (2022).

[11] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, *Sdedit: Guided image synthesis and editing with stochastic differential equations*, in *ICLR*, 2021.

[12] T. Brooks, A. Holynski, and A. A. Efros, *Instructpix2pix: Learning to follow image editing instructions*, in *CVPR*, 2023.

[13] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, *Prompt-to-prompt image editing with cross attention control*, *arXiv preprint arXiv:2208.01626* (2022).

[14] L. Zhang and M. Agrawala, *Adding conditional control to text-to-image diffusion models*, *arXiv preprint arXiv:2302.05543* (2023).

[15] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee, *Gligen: Open-set grounded text-to-image generation*, *arXiv preprint arXiv:2301.07093* (2023).

[16] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or, *Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models*, *ACM transactions on Graphics (TOG)* (2023).

[17] OpenAI, "Introducing gpt-5." `https://openai.com/index/introducing-gpt-5`, 2025. Accessed: 2025-10-04.

[18] G. Deepmind, "Introducing gemini 2.0: our new ai model for the agentic era." https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/.

[19] F. Li, R. Zhang, H. Zhang, Y. Zhang, B. Li, W. Li, Z. Ma, and C. Li, *Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models*, *arXiv preprint arXiv:2407.07895* (2024).

[20] Q. Team, *Qwen3. 5-omni technical report*, *arXiv preprint arXiv:2604.15804* (2026).

[21] Q. Wang, R. Ding, Y. Zeng, Z. Chen, L. Chen, S. Wang, P. Xie, F. Huang, and F. Zhao, *Vrag-rl: Empower vision-perception-based rag for visually rich information understanding via iterative reasoning with reinforcement learning*, *arXiv preprint arXiv:2505.22019* (2025).

[22] OpenAI, "Introducing deep research." https://openai.com/index/introducing-deep-research/.

[23] Z. Wang, A. Li, Z. Li, and X. Liu, *Genartist: Multimodal llm as an agent for unified image generation and editing*, *Neurips* (2024).

[24] L. Yang, Z. Yu, C. Meng, M. Xu, S. Ermon, and B. Cui, *Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms*, in *ICML*, 2024.

[25] K. Huang, Y. Huang, X. Ning, Z. Lin, Y. Wang, and X. Liu, *Genmac: compositional text-to-video generation with multi-agent collaboration*, *arXiv preprint arXiv:2412.04440* (2024).

[26] Z. Lu, Y. Chai, Y. Guo, X. Yin, L. Liu, H. Wang, H. Xiao, S. Ren, G. Xiong, and H. Li, *Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning*, *arXiv preprint arXiv:2503.21620* (2025).

[27] R. Luo, L. Wang, W. He, L. Chen, J. Li, and X. Xia, *Gui-r1: A generalist r1-style vision-language action model for gui agents*, *arXiv preprint arXiv:2504.10458* (2025).

[28] OpenAI, "Introducing chatgpt agent: bridging research and action." https://openai.com/index/introducing-chatgpt-agent/.

[29] Adobe, "Adobe lightroom." https://lightroom.adobe.com.

[30] Picsart, "Picsart." https://picsart.com.

[31] N. S. Dutt, D. Ceylan, and N. J. Mitra, *Monetgpt: Solving puzzles enhances mllms' image retouching skills*, *ACM Transactions on Graphics (TOG)* (2025).

[32] Y. Lin, Z. Lin, K. Lin, J. Bai, P. Pan, C. Li, H. Chen, Z. Wang, X. Ding, W. Li, *et. al.*, *Jarvisart: Liberating human artistic creativity via an intelligent photo retouching agent*, *arXiv preprint arXiv:2506.17612* (2025).

[33] H. Chen, W. Li, J. Gu, J. Ren, S. Chen, T. Ye, R. Pei, K. Zhou, F. Song, and L. Zhu, *Restoreagent: Autonomous image restoration agent via multimodal large language models*, *Neurips* (2024).

[34] M. V. Conde, Z. Lu, and R. Timofte, *Pixtalk: Controlling photorealistic image processing and editing with language*, in *ICCV*, 2025.

[35] S. Kosugi and T. Yamasaki, *Unpaired image enhancement featuring reinforcement-learning-controlled image editing software*, in *AAAI*, 2020.

[36] Z. Ke, C. Sun, L. Zhu, K. Xu, and R. W. Lau, *Harmonizer: Learning to perform white-box image and video harmonization*, in *ECCV*, 2022.

[37] Y. Hu, H. He, C. Xu, B. Wang, and S. Lin, *Exposure: A white-box photo post-processing framework*, *ACM Transactions on Graphics (TOG)* **37** (2018), no. 2 1–17.

[38] W. Ouyang, Y. Dong, X. Kang, P. Ren, X. Xu, and X. Xie, *Rsfnet: A white-box image retouching approach using region-specific color filters*, in *ICCV*, 2023.

[39] J. Shi, N. Xu, Y. Xu, T. Bui, F. Dernoncourt, and C. Xu, *Learning by planning: Language-guided global image editing*, in *CVPR*, 2021.

[40] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, *Neurips* (2020).

[41] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-based generative modeling through stochastic differential equations*, in *ICLR*, 2021.

[42] Z.-P. Duan, J. Zhang, Z. Lin, X. Jin, X. Wang, D. Zou, C.-L. Guo, and C. Li, *Diffretouch: Using diffusion to retouch on the shoulder of experts*, in *AAAI*, 2025.

[43] Q. Wu, Y. Liu, H. Zhao, A. Kale, T. Bui, T. Yu, Z. Lin, Y. Zhang, and S. Chang, *Uncovering the disentanglement capability in text-to-image diffusion models*, in *CVPR*, 2023.

[44] J. Liu, Q. Ying, Z. Qian, S. Li, R. Zhang, J. Liu, and X. Zhang, *Mofrr: Mixture of diffusion models for face retouching restoration*, in *ICCV*, 2025.

[45] M. Ye-Bin, R. Miles, T.-H. Oh, I. Elezi, and J. Deng, *Retouchllm: Training-free code-based image retouching with vision language models*, *arXiv e-prints* (2025).

[46] X. He, L. Li, Y. Wang, H. Zheng, K. Cao, K. Yan, R. Li, C. Xie, J. Zhang, and M. Zhou, *Training-free large model priors for multiple-in-one image restoration*, *arXiv preprint arXiv:2407.13181* (2024).

[47] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, *et. al.*, *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*, *arXiv preprint arXiv:2501.12948* (2025).

[48] W. Hong, W. Yu, X. Gu, G. Wang, G. Gan, H. Tang, J. Cheng, J. Qi, J. Ji, L. Pan, *et. al.*, *Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning*, *arXiv e-prints* (2025) arXiv–2507.

[49] H. Wang, C. Qu, Z. Huang, W. Chu, F. Lin, and W. Chen, *Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning*, *arXiv preprint arXiv:2504.08837* (2025).

[50] W. Shen, J. Pei, Y. Peng, X. Song, Y. Liu, J. Peng, H. Sun, Y. Hao, P. Wang, J. Zhang, *et. al.*, *Skywork-r1v3 technical report*, *arXiv preprint arXiv:2507.06167* (2025).

[51] R. Qiao, Q. Tan, P. Yang, Y. Wang, X. Wang, E. Wan, S. Zhou, G. Dong, Y. Zeng, Y. Xu, *et. al.*, *We-math 2.0: A versatile mathbook system for incentivizing visual mathematical reasoning*, *arXiv preprint arXiv:2508.10433* (2025).

[52] K. Fan, K. Feng, H. Lyu, D. Zhou, and X. Yue, *Sophiavl-r1: Reinforcing mllms reasoning with thinking reward*, *arXiv preprint arXiv:2505.17018* (2025).

[53] Z. Liu, P. Wang, R. Xu, S. Ma, C. Ruan, P. Li, Y. Liu, and Y. Wu, *Inference-time scaling for generalist reward modeling*, *arXiv preprint arXiv:2504.02495* (2025).

[54] J. Zhou, J. Ji, B. Chen, J. Sun, W. Chen, D. Hong, S. Han, Y. Guo, and Y. Yang, *Generative rlhf-v: Learning principles from multi-modal human preference*, *arXiv preprint arXiv:2505.18531* (2025).

[55] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, *Qwen2.5-vl technical report*, *arXiv preprint arXiv:2502.13923* (2025).

[56] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, *Learning photographic global tonal adjustment with a database of input / output image pairs*, in *CVPR*, 2011.

[57] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen, *et. al.*, *Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities*, *arXiv preprint arXiv:2507.06261* (2025).

[58] B. F. Labs, "Flux 1.1 pro." https://bfl.ai/models/flux-pro.

[59] X. Liu, Y.-W. Tai, and C.-K. Tang, *Chatcam: Empowering camera control through conversational ai*, *arXiv preprint arXiv:2409.17331* (2024).

[60] ArtistT2V., "Artist text-to-video model." https://artlist.io/image-to-video-ai.

[61] Google, "Google veo 3." https://gemini.google/overview/video-generation/, 2025.

[62] H. He, Y. Xu, Y. Guo, G. Wetzstein, B. Dai, H. Li, and C. Yang, *Cameractrl: Enabling camera control for text-to-video generation*, *arXiv preprint arXiv:2404.02101* (2024).

[63] S. Bahmani, I. Skorokhodov, G. Qian, A. Siarohin, W. Menapace, A. Tagliasacchi, D. B. Lindell, and S. Tulyakov, *Ac3d: Analyzing and improving 3d camera control in video diffusion transformers*, in *CVPR*, 2025.

[64] Q. Wang, Y. Luo, X. Shi, X. Jia, H. Lu, T. Xue, X. Wang, P. Wan, D. Zhang, and K. Gai, *Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation*, in *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025.

[65] X. Shuai, H. Ding, Z. Qin, H. Luo, X. Ma, and D. Tao, *Free-form motion control: A synthetic video generation dataset with controllable camera and object motions*, *arXiv preprint arXiv:2501.01425* (2025).

[66] M. Saito, E. Matsumoto, and S. Saito, *Temporal generative adversarial nets with singular value clipping*, in *ICCV*, 2017.

[67] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, *Mocogan: Decomposing motion and content for video generation*, in *CVPR*, 2018.

[68] W. Peebles and S. Xie, *Scalable diffusion models with transformers*, in *ICCV*, 2023.

[69] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden, and S. Xie, *Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers*, *arXiv preprint arXiv:2401.08740* (2024).

[70] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis, *Align your latents: High-resolution video synthesis with latent diffusion models*, in *CVPR*, 2023.

[71] J. Gu, S. Wang, H. Zhao, T. Lu, X. Zhang, Z. Wu, S. Xu, W. Zhang, Y.-G. Jiang, and H. Xu, *Reuse and diffuse: Iterative denoising for text-to-video generation*, *arXiv preprint arXiv:2309.03549* (2023).

[72] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, *et. al.*, *Make-a-video: Text-to-video generation without text-video data*, *arXiv preprint arXiv:2209.14792* (2022).

[73] R. Wu, L. Chen, T. Yang, C. Guo, C. Li, and X. Zhang, *Lamp: Learn a motion pattern for few-shot-based video generation*, *arXiv preprint arXiv:2310.10769* (2023).

[74] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, *et. al.*, *Stable video diffusion: Scaling latent video diffusion models to large datasets*, *arXiv preprint arXiv:2311.15127* (2023).

[75] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang, *Cogvideo: Large-scale pretraining for text-to-video generation via transformers*, *arXiv preprint arXiv:2205.15868* (2022).

[76] X. Ma, Y. Wang, G. Jia, X. Chen, Z. Liu, Y.-F. Li, C. Chen, and Y. Qiao, *Latte: Latent diffusion transformer for video generation*, *arXiv preprint arXiv:2401.03048* (2024).

[77] W. Wang, J. Liu, Z. Lin, J. Yan, S. Chen, C. Low, T. Hoang, J. Wu, J. H. Liew, H. Yan, *et. al.*, *Magicvideo-v2: Multi-stage high-aesthetic video generation*, *arXiv preprint arXiv:2401.04468* (2024).

[78] Kuaishou, "Kling." https://kling.kuaishou.com/en, 2025.

[79] W. Sun, S. Chen, F. Liu, Z. Chen, Y. Duan, J. Zhang, and Y. Wang, *Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion*, *arXiv preprint arXiv:2411.04928* (2024).

[80] H. Zhou, C. Wang, R. Nie, J. Liu, D. Yu, Q. Yu, and C. Wang, *Trackgo: A flexible and efficient method for controllable video generation*, in *AAAI*, 2025.

[81] B. Peng, J. Wang, Y. Zhang, W. Li, M.-C. Yang, and J. Jia, *Controlnext: Powerful and efficient control for image and video generation*, *arXiv preprint arXiv:2408.06070* (2024).

[82] D. J. Zhang, R. Paiss, S. Zada, N. Karnad, D. E. Jacobs, Y. Pritch, I. Mosseri, M. Z. Shou, N. Wadhwa, and N. Ruiz, *Recapture: Generative video camera controls for user-provided videos using masked video fine-tuning*, *arXiv preprint arXiv:2411.05003* (2024).

[83] D. Xu, Y. Jiang, C. Huang, L. Song, T. Gernoth, L. Cao, Z. Wang, and H. Tang, *Cavia: Camera-controllable multi-view video diffusion with view-integrated attention*, *arXiv preprint arXiv:2410.10774* (2024).

[84] C. Hou, G. Wei, Y. Zeng, and Z. Chen, *Training-free camera control for video generation*, *arXiv preprint arXiv:2406.10126* (2024).

[85] P. Ling, J. Bu, P. Zhang, X. Dong, Y. Zang, T. Wu, H. Chen, J. Wang, and Y. Jin, *Motionclone: Training-free motion cloning for controllable video generation*, *arXiv preprint arXiv:2406.05338* (2024).

[86] D. Xu, W. Nie, C. Liu, S. Liu, J. Kautz, Z. Wang, and A. Vahdat, *Camco: Camera-controllable 3d-consistent image-to-video generation*, *arXiv preprint arXiv:2406.02509* (2024).

[87] S. Yang, L. Hou, H. Huang, C. Ma, P. Wan, D. Zhang, X. Chen, and J. Liao, *Direct-a-video: Customized video generation with user-directed camera movement and object motion*, in *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–12, 2024.

[88] Z. Wang, Z. Yuan, X. Wang, Y. Li, T. Chen, M. Xia, P. Luo, and Y. Shan, *Motionctrl: A unified and flexible motion controller for video generation*, in *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

[89] H. He, C. Yang, S. Lin, Y. Xu, M. Wei, L. Gui, Q. Zhao, G. Wetzstein, L. Jiang, and H. Li, *Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models*, *arXiv preprint arXiv:2503.10592* (2025).

[90] M. Yu, W. Hu, J. Xing, and Y. Shan, *Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models*, *ICCV* (2025).

[91] J. Bai, M. Xia, X. Wang, Z. Yuan, X. Fu, Z. Liu, H. Hu, P. Wan, and D. Zhang, *Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints*, *arXiv preprint arXiv:2412.07760* (2024).

[92] M. J. Black, P. Patel, J. Tesch, and J. Yang, *Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion*, in *CVPR*, 2023.

[93] Z. Wang, Y. Li, Y. Zeng, Y. Fang, Y. Guo, W. Liu, J. Tan, K. Chen, T. Xue, B. Dai, *et. al.*, *Humanvid: Demystifying training data for camera-controllable human image animation*, *arXiv preprint arXiv:2407.17438* (2024).

[94] Z. Yang, Z. Cai, H. Mei, S. Liu, Z. Chen, W. Xiao, Y. Wei, Z. Qing, C. Wei, B. Dai, *et. al.*, *Synbody: Synthetic dataset with layered human models for 3d human perception and modeling*, in *ICCV*, 2023.

[95] J. Bai, M. Xia, X. Fu, X. Wang, L. Mu, J. Cao, Z. Liu, H. Hu, X. Bai, P. Wan, *et. al.*, *Recammaster: Camera-controlled generative rendering from a single video*, *arXiv preprint arXiv:2503.11647* (2025).

[96] X. Fu, X. Liu, X. Wang, S. Peng, M. Xia, X. Shi, Z. Yuan, P. Wan, D. Zhang, and D. Lin, *3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation*, *arXiv preprint arXiv:2412.07759* (2024).

[97] S. Y. Cheong, D. Ceylan, A. Mustafa, A. Gilbert, and C.-H. P. Huang, *Boosting camera motion control for video diffusion transformers*, *arXiv preprint arXiv:2410.10802* (2024).

[98] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, *Fvd: A new metric for video generation*, .

[99] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, *Stereo magnification: Learning view synthesis using multiplane images*, *arXiv preprint arXiv:1805.09817* (2018).

[100] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, in *Neurips* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), 2014.

[101] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein gan*, 2017.

[102] T. Karras, T. Aila, S. Laine, and J. Lehtinen, *Progressive growing of gans for improved quality, stability, and variation*, 2017.

[103] A. Brock, J. Donahue, and K. Simonyan, *Large scale gan training for high fidelity natural image synthesis*, 2018.

[104] T. Karras, S. Laine, and T. Aila, *A style-based generator architecture for generative adversarial networks*, in *CVPR*, 2019.

[105] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, *arXiv preprint arXiv:1312.6114* (2013).

[106] D. J. Rezende, S. Mohamed, and D. Wierstra, *Stochastic backpropagation and approximate inference in deep generative models*, in *ICML*, 2014.

[107] D. Rezende and S. Mohamed, *Variational inference with normalizing flows*, in *ICML*, 2015.

[108] A. Razavi, A. van den Oord, and O. Vinyals, *Generating diverse high-fidelity images with vq-vae-2*, in *Neurips*, 2019.

[109] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, *Deep unsupervised learning using nonequilibrium thermodynamics*, in *ICML*, 2015.

[110] J. Song, C. Meng, and S. Ermon, *Denoising diffusion implicit models*, *arXiv preprint arXiv:2010.02502* (2020).

[111] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, *Ganspace: Discovering interpretable gan controls*, in *Neurips*, 2020.

[112] A. Achille and S. Soatto, *Emergence of invariance and disentanglement in deep representations*, *The Journal of Machine Learning Research* (2018).

[113] C. Eastwood and C. K. Williams, *A framework for the quantitative evaluation of disentangled representations*, in *ICLR*, 2018.

[114] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, *Analyzing and improving the image quality of stylegan*, in *CVPR*, 2020.

[115] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, *Alias-free generative adversarial networks*, in *Neurips*, 2021.

[116] Y. Shen, C. Yang, X. Tang, and B. Zhou, *Interfacegan: Interpreting the disentangled face representation learned by gans*, *IEEE transactions on pattern analysis and machine intelligence* (2020).

[117] Y. Shen, J. Gu, X. Tang, and B. Zhou, *Interpreting the latent space of gans for semantic face editing*, in *CVPR*, 2020.

[118] Y. Shen and B. Zhou, *Closed-form factorization of latent semantics in gans*, in *CVPR*, 2021.

[119] R. Abdal, Y. Qin, and P. Wonka, *Image2stylegan: How to embed images into the stylegan latent space?*, in *ICCV*, 2019.

[120] R. Abdal, Y. Qin, and P. Wonka, *Image2stylegan++: How to edit the embedded images?*, in *CVPR*, 2020.

[121] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, *Styleclip: Text-driven manipulation of stylegan imagery*, in *ICCV*, 2021.

[122] X. Ren, T. Yang, Y. Wang, and W. Zeng, *Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view*, in *ICLR*, 2021.

[123] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, *Infogan: Interpretable representation learning by information maximizing generative adversarial nets*, in *Neurips*, 2016.

[124] H. Kim and A. Mnih, *Disentangling by factorising*, in *ICML*, 2018.

[125] B. Paige, J.-W. van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, P. Torr, *et. al.*, *Learning disentangled representations with semi-supervised deep generative models*, in *Neurips*, 2017.

[126] M. Kwon, J. Jeong, and Y. Uh, *Diffusion models already have a semantic latent space*, *arXiv preprint arXiv:2210.10960* (2022).

[127] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241, Springer, 2015.

[128] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn, *Diffusion autoencoders: Toward a meaningful and decodable representation*, in *CVPR*, 2022.

[129] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, *Cascaded diffusion models for high fidelity image generation.*, *J. Mach. Learn. Res.* (2022).

[130] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, *Image super-resolution via iterative refinement*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[131] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, *Palette: Image-to-image diffusion models*, in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022.

[132] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, *Repaint: Inpainting using denoising diffusion probabilistic models*, in *CVPR*, 2022.

[133] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, *Variational diffusion models*, *ArXiv* (2021).

[134] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, *Video diffusion models*, *arXiv preprint arXiv:2204.03458* (2022).

[135] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, *Diffwave: A versatile diffusion model for audio synthesis*, *arXiv preprint arXiv:2009.09761* (2020).

[136] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, *Wavegrad: Estimating gradients for waveform generation*, *arXiv preprint arXiv:2009.00713* (2020).

[137] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. v. d. Berg, *Structured denoising diffusion models in discrete state-spaces*, 2021.

[138] Y. Tashiro, J. Song, Y. Song, and S. Ermon, *Csdi: Conditional score-based diffusion models for probabilistic time series imputation*, 2021.

[139] B. Jing, G. Corso, J. Chang, R. Barzilay, and T. Jaakkola, *Torsional diffusion for molecular conformer generation*, 2022.

[140] P. Dhariwal and A. Nichol, *Diffusion models beat gans on image synthesis*, in *Neurips*, 2021.

[141] X. Liu, D. H. Park, S. Azadi, G. Zhang, A. Chopikyan, Y. Hu, H. Shi, A. Rohrbach, and T. Darrell, *More control for free! image synthesis with semantic diffusion guidance*, *arXiv preprint arXiv:2112.05744* (2021).

[142] J. Ho and T. Salimans, *Classifier-free diffusion guidance*, 2022.

[143] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, *Glide: Towards photorealistic image generation and editing with text-guided diffusion models*, 2021.

[144] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, *Hierarchical text-conditional image generation with clip latents*, *arXiv preprint arXiv:2204.06125* (2022).

[145] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, in *ICCV*, 2017.

[146] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, *Image analogies*, in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.

[147] D. Bau, H. Strobelt, W. Peebles, J. Wulff, B. Zhou, J.-Y. Zhu, and A. Torralba, *Semantic photo manipulation with a generative image prior*, *arXiv preprint arXiv:2005.07727* (2020).

[148] Y. Jo and J. Park, *Sc-fegan: Face editing generative adversarial network with user's sketch and color*, in *ICCV*, 2019.

[149] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, *Manigan: Text-guided image manipulation*, in *CVPR*, 2020.

[150] Y. Liu, M. De Nadai, D. Cai, H. Li, X. Alameda-Pineda, N. Sebe, and B. Lepri, *Describe what to change: A text-guided unsupervised image-to-image translation approach*, in *ACM MM*, 2020.

[151] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum, *Compositional visual generation with composable diffusion models*, in *ECCV*, 2022.

[152] G. Couairon, J. Verbeek, H. Schwenk, and M. Cord, *Diffedit: Diffusion-based semantic image editing with mask guidance*, *arXiv preprint arXiv:2210.11427* (2022).

[153] M. Li, J. Lin, C. Meng, S. Ermon, S. Han, and J.-Y. Zhu, *Efficient spatially sparse inference for conditional gans and diffusion models*, *arXiv preprint arXiv:2211.02048* (2022).

[154] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et. al.*, *Learning transferable visual models from natural language supervision*, in *ICML*, 2021.

[155] O. Avrahami, D. Lischinski, and O. Fried, *Blended diffusion for text-driven editing of natural images*, in *CVPR*, 2022.

[156] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani, *Imagic: Text-based real image editing with diffusion models*, *arXiv preprint arXiv:2210.09276* (2022).

[157] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, *Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation*, *arXiv preprint arXiv:2208.12242* (2022).

[158] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, *An image is worth one word: Personalizing text-to-image generation using textual inversion*, *arXiv preprint arXiv:2208.01618* (2022).

[159] G. Kim, T. Kwon, and J. C. Ye, *Diffusionclip: Text-guided diffusion models for robust image manipulation*, in *CVPR*, 2022.

[160] J. Ackermann and M. Li, *High-resolution image editing via multi-stage blended diffusion*, *arXiv preprint arXiv:2210.12965* (2022).

[161] O. Avrahami, O. Fried, and D. Lischinski, *Blended latent diffusion*, *arXiv preprint arXiv:2206.02779* (2022).

[162] R. Gal, O. Patashnik, H. Maron, A. H. Bermano, G. Chechik, and D. Cohen-Or, *Stylegan-nada: Clip-guided domain adaptation of image generators*, *ACM Transactions on Graphics (TOG)* (2022).

[163] J. Johnson, A. Alahi, and L. Fei-Fei, *Perceptual losses for real-time style transfer and super-resolution*, in *ECCV*, 2016.

[164] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, *Laion-400m: Open dataset of clip-filtered 400 million image-text pairs*, *arXiv* (2021).

[165] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, *Pseudo numerical methods for diffusion models on manifolds*, in *ICLR*, 2022.

[166] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, in *ICLR*, 2015.

[167] Z. Liu, P. Luo, X. Wang, and X. Tang, *Deep learning face attributes in the wild*, in *ICCV*, 2015.

[168] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, *Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop*, *arXiv preprint arXiv:1506.03365* (2015).

[169] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, *Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework*, in *ICML*, 2022.

[170] T. Karras, M. Aittala, T. Aila, and S. Laine, *Elucidating the design space of diffusion-based generative models*, 2022.

[171] Y. Balaji, S. Nah, X. Huang, A. Vahdat, J. Song, K. Kreis, M. Aittala, T. Aila, S. Laine, B. Catanzaro, *et. al.*, *ediffi: Text-to-image diffusion models with an ensemble of expert denoisers*, *arXiv* (2022).

[172] W. Feng, X. He, T.-J. Fu, V. Jampani, A. Akula, P. Narayana, S. Basu, X. E. Wang, and W. Y. Wang, *Training-free structured diffusion guidance for compositional text-to-image synthesis*, *arXiv* (2022).

[173] G. Marcus, E. Davis, and S. Aaronson, *A very preliminary analysis of dall-e 2*, 2022.

[174] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick, *Microsoft coco captions: Data collection and evaluation server*, 2015.

[175] F. Liu, G. Emerson, and N. Collier, *Visual spatial reasoning*, *arXiv preprint arXiv:2205.00363* (2022).

[176] S. Xie, Z. Zhang, Z. Lin, T. Hinz, and K. Zhang, *Smartbrush: Text and shape guided object inpainting with diffusion model*, *arXiv preprint arXiv:2212.05034* (2022).

[177] S. Yu, K. Sohn, S. Kim, and J. Shin, *Video probabilistic diffusion models in projected latent space*, *arXiv preprint arXiv:2302.07685* (2023).

[178] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto, *Diffusion-lm improves controllable text generation*, *arXiv preprint arXiv:2205.14217* (2022).

[179] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, and J. Tang, *Geodiff: a geometric diffusion model for molecular conformation generation*, 2022.

[180] Y. Zhou, R. Zhang, C. Chen, C. Li, C. Tensmeyer, T. Yu, J. Gu, J. Xu, and T. Sun, *Lafite: Towards language-free training for text-to-image generation*, *arXiv preprint arXiv:2111.13792* (2021).

[181] W.-C. Fan, Y.-C. Chen, D. Chen, Y. Cheng, L. Yuan, and Y.-C. F. Wang, *Frido: Feature pyramid diffusion for complex scene image synthesis*, *arXiv preprint arXiv:2208.13753* (2022).

[182] Z. Yang, D. Liu, C. Wang, J. Yang, and D. Tao, *Modeling image composition for complex scene generation*, in *CVPR*, 2022.

[183] N. Huang, F. Tang, W. Dong, T.-Y. Lee, and C. Xu, *Region-aware diffusion for zero-shot text-driven image editing*, *arXiv preprint arXiv:2302.11797* (2023).

[184] D. H. Park, G. Luo, C. Toste, S. Azadi, X. Liu, M. Karalashvili, A. Rohrbach, and T. Darrell, *Shape-guided diffusion with inside-outside attention*, *arXiv preprint arXiv:2212.00210* (2022).

[185] J. Singh, S. Gould, and L. Zheng, *High-fidelity guided image synthesis with latent diffusion models*, *arXiv preprint arXiv:2211.17084* (2022).

[186] Y. Zeng, Z. Lin, J. Zhang, Q. Liu, J. Collomosse, J. Kuen, and V. M. Patel, *Scenecomposer: Any-level semantic image synthesis*, *arXiv preprint arXiv:2211.11742* (2022).

[187] C.-F. Yang, W.-C. Fan, F.-E. Yang, and Y.-C. F. Wang, *Layouttransformer: Scene layout generation with conceptual and spatial diversity*, in *CVPR*, 2021.

[188] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, *Language models are few-shot learners*, 2020.

[189] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, *Neurips* (2017).

[190] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, *arXiv* (2019).

[191] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context*, in *ECCV*, 2014.

[192] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, *End-to-end object detection with transformers*, 2020.

[193] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et. al.*, *Llama: Open and efficient foundation language models*, *arXiv preprint arXiv:2302.13971* (2023).

[194] X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu, *et. al.*, *Deepseek llm: Scaling open-source language models with longtermism*, *arXiv preprint arXiv:2401.02954* (2024).

[195] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, *et. al.*, *Mixtral of experts*, *arXiv preprint arXiv:2401.04088* (2024).

[196] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et. al.*, *Language models are few-shot learners*, *Neurips* (2020).

[197] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et. al.*, *Gpt-4 technical report*, *arXiv preprint arXiv:2303.08774* (2023).

[198] H. Liu, C. Li, Q. Wu, and Y. J. Lee, *Visual instruction tuning*, *Neurips* (2023).

[199] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et. al.*, *Gemini: a family of highly capable multimodal models*, *arXiv preprint arXiv:2312.11805* (2023).

[200] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa, *et. al.*, *Openflamingo: An open-source framework for training large autoregressive vision-language models*, *arXiv preprint arXiv:2308.01390* (2023).

[201] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et. al.*, *Flamingo: a visual language model for few-shot learning*, *Neurips* (2022).

[202] K. Ying, F. Meng, J. Wang, Z. Li, H. Lin, Y. Yang, H. Zhang, W. Zhang, Y. Lin, S. Liu, *et. al.*, *Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi*, *arXiv preprint arXiv:2404.16006* (2024).

[203] X. Yang, Y. Wu, M. Yang, H. Chen, and X. Geng, *Exploring diverse in-context configurations for image captioning*, *Neurips* (2024).

[204] Z. Shao, Z. Yu, M. Wang, and J. Yu, *Prompting large language models with answer heuristics for knowledge-based visual question answering*, in *CVPR*, 2023.

[205] G. Zheng, B. Yang, J. Tang, H.-Y. Zhou, and S. Yang, *Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models*, *Neurips* (2023).

[206] K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati, *On the planning abilities of large language models - a critical investigation*, in *Neurips*, 2023.

[207] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, C. Wei, B. Yu, R. Yuan, R. Sun, M. Yin, B. Zheng, Z. Yang, Y. Liu, W. Huang, H. Sun, Y. Su, and W. Chen, *Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi*, in *CVPR*, 2024.

[208] K. Wang, J. Pan, W. Shi, Z. Lu, M. Zhan, and H. Li, *Measuring multimodal mathematical reasoning with math-vision dataset*, *arXiv preprint arXiv:2402.14804* (2024).

[209] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, J. Yang, X. Zheng, K. Li, X. Sun, Y. Wu, and R. Ji, *Mme: A comprehensive evaluation benchmark for multimodal large language models*, *arXiv preprint arXiv:2306.13394* (2023).

[210] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan, *Seed-bench: Benchmarking multimodal llms with generative comprehension*, *arXiv preprint arXiv:2307.16125* (2023).

[211] W. Yu, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, X. Wang, and L. Wang, *Mm-vet: Evaluating large multimodal models for integrated capabilities*, *arXiv preprint arXiv:2308.02490* (2023).

[212] D. Ghosal, V. T. Y. Han, C. Y. Ken, and S. Poria, *Are language models puzzle prodigies? algorithmic puzzles unveil serious challenges in multimodal reasoning*, *arXiv preprint arXiv:2403.03864* (2024).

[213] W. Zhang, Z. Cheng, Y. He, M. Wang, Y. Shen, Z. Tan, G. Hou, M. He, Y. Ma, W. Lu, *et. al.*, *Multimodal self-instruct: Synthetic abstract image and visual reasoning instruction using language model*, *arXiv preprint arXiv:2407.07053* (2024).

[214] X. Tian, J. Gu, B. Li, Y. Liu, C. Hu, Y. Wang, K. Zhan, P. Jia, X. Lang, and H. Zhao, *Drivevlm: The convergence of autonomous driving and large vision-language models*, *arXiv preprint arXiv:2402.12289* (2024).

[215] Y. Ma, Y. Cao, J. Sun, M. Pavone, and C. Xiao, *Dolphins: Multimodal language model for driving*, *arXiv preprint arXiv:2312.00438* (2023).

[216] H. Chang, K. Gao, K. Boyalakuntla, A. Lee, B. Huang, H. U. Kumar, J. Yu, and A. Boularias, *Lgmcts: Language-guided monte-carlo tree search for executable semantic object rearrangement*, *arXiv preprint arXiv:2309.15821* (2023).

[217] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao, *Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning*, *arXiv preprint arXiv:2311.17842* (2023).

[218] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. Sri, A. Barrett, D. Christianson, *et. al.*, *Pddl— the planning domain definition language*, *Technical Report, Tech. Rep.* (1998).

[219] X. Guo, S. Singh, H. Lee, R. L. Lewis, and X. Wang, *Deep learning for real-time atari game play using offline monte-carlo tree search planning*, *Neurips* (2014).

[220] R. S. Sutton, *Planning by incremental dynamic programming*, in *Machine learning proceedings*. 1991.

[221] S. Kambhampati, *Can large language models reason and plan?*, *Annals of the New York Academy of Sciences* (2024).

[222] S. Kambhampati, K. Valmeekam, L. Guan, K. Stechly, M. Verma, S. Bhambri, L. Saldyt, and A. Murthy, *Llms can't plan, but can help planning in llm-modulo frameworks*, *arXiv preprint arXiv:2402.01817* (2024).

[223] K. Stechly, K. Valmeekam, and S. Kambhampati, *On the self-verification limitations of large language models on reasoning and planning tasks*, *arXiv preprint arXiv:2402.08115* (2024).

[224] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et. al.*, *Chain-of-thought prompting elicits reasoning in large language models*, *Neurips* (2022).

[225] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, *Tree of thoughts: Deliberate problem solving with large language models*, *Neurips* (2024).

[226] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, *Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face*, *Neurips* (2024).

[227] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, *React: Synergizing reasoning and acting in language models*, *arXiv preprint arXiv:2210.03629* (2022).

[228] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et. al.*, *Do as i can, not as i say: Grounding language in robotic affordances*, *arXiv preprint arXiv:2204.01691* (2022).

[229] R. Hazra, P. Z. Dos Martires, and L. De Raedt, *Saycanpay: Heuristic planning with large language models using learnable domain knowledge*, in *AAAI*, 2024.

[230] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, *Reflexion: Language agents with verbal reinforcement learning*, *Neurips* (2024).

[231] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, *et. al.*, *Self-refine: Iterative refinement with self-feedback*, *Neurips* (2024).

[232] L. Guan, K. Valmeekam, S. Sreedharan, and S. Kambhampati, *Leveraging pre-trained large language models to construct and utilize world models for model-based task planning*, *Neurips* (2023).

[233] J. Ruan, Y. Chen, B. Zhang, Z. Xu, T. Bao, G. Du, S. Shi, H. Mao, X. Zeng, and R. Zhao, *Tptu: Task planning and tool usage of large language model-based ai agents*, *arXiv preprint arXiv:2308.03427* (2023).

[234] Y. Wu, X. Tang, T. M. Mitchell, and Y. Li, *Smartplay: A benchmark for llms as intelligent agents*, *arXiv preprint arXiv:2310.01557* (2023).

[235] J. Xie, K. Zhang, J. Chen, T. Zhu, R. Lou, Y. Tian, Y. Xiao, and Y. Su, *Travelplanner: A benchmark for real-world planning with language agents*, *arXiv preprint arXiv:2402.01622* (2024).

[236] K. Valmeekam, A. Olmo, S. Sreedharan, and S. Kambhampati, *Large language models still can't plan (a benchmark for llms on planning and reasoning about change)*, *arXiv preprint arXiv:2206.10498* (2022).

[237] X. Liu, T. Zhang, Y. Gu, I. L. Iong, Y. Xu, X. Song, S. Zhang, H. Lai, X. Liu, H. Zhao, *et. al.*, *Visualagentbench: Towards large multimodal models as visual foundation agents*, *arXiv preprint arXiv:2408.06327* (2024).

[238] F. Joublin, A. Ceravola, P. Smirnov, F. Ocker, J. Deigmoeller, A. Belardinelli, C. Wang, S. Hasler, D. Tanneberg, and M. Gienger, *Copal: Corrective planning of robot actions with large language models*, *arXiv preprint arXiv:2310.07263* (2023).

[239] M. S. Sakib and Y. Sun, *From cooking recipes to robot task trees–improving planning correctness and task efficiency by leveraging llms with a knowledge network*, *arXiv preprint arXiv:2309.09181* (2023).

[240] A. Majumdar, A. Ajay, X. Zhang, P. Putta, S. Yenamandra, M. Henaff, S. Silwal, P. Mcvay, O. Maksymets, S. Arnaud, *et. al.*, *Openeqa: Embodied question answering in the era of foundation models*, in *CVPR*, 2024.

[241] W. Feng, W. Zhu, T.-j. Fu, V. Jampani, A. Akula, X. He, S. Basu, X. E. Wang, and W. Y. Wang, *Layoutgpt: Compositional visual planning and generation with large language models*, *Neurips* (2024).

[242] M. Shridhar, X. Yuan, M.-A. Côté, Y. Bisk, A. Trischler, and M. Hausknecht, *Alfworld: Aligning text and embodied environments for interactive learning*, *arXiv preprint arXiv:2010.03768* (2020).

[243] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang, *The dawn of lmms: Preliminary explorations with gpt-4v (ision)*, *arXiv preprint arXiv:2309.17421* **9** (2023), no. 1 1.

[244] M. Aghzal, E. Plaku, and Z. Yao, *Can large language models be good path planners? a benchmark and investigation on spatial-temporal reasoning*, *arXiv preprint arXiv:2310.03249* (2023).

[245] Z. Qi, Y. Fang, M. Zhang, Z. Sun, T. Wu, Z. Liu, D. Lin, J. Wang, and H. Zhao, *Gemini vs gpt-4v: A preliminary comparison and combination of vision-language models through qualitative cases*, *arXiv preprint arXiv:2312.15011* (2023).

[246] S. Cha, J. Lee, Y. Lee, and C. Yang, *Visually dehallucinative instruction generation: Know what you don't know*, *arXiv preprint arXiv:2402.09717* (2024).

[247] W. Ge, S. Chen, G. Chen, J. Chen, Z. Chen, S. Yan, C. Zhu, Z. Lin, W. Xie, X. Wang, *et. al.*, *Mllm-bench, evaluating multi-modal llms using gpt-4v*, *arXiv preprint arXiv:2311.13951* (2023).

[248] S. Tong, Z. Liu, Y. Zhai, Y. Ma, Y. LeCun, and S. Xie, *Eyes wide shut? exploring the visual shortcomings of multimodal llms*, *arXiv preprint arXiv:2401.06209* (2024).

[249] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao, *Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models*, *arXiv e-prints* (2023) arXiv–2310.

[250] F. Liu, T. Guan, Z. Li, L. Chen, Y. Yacoob, D. Manocha, and T. Zhou, *Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models*, *arXiv preprint arXiv:2310.14566* (2023).

[251] J. Nie, G. Zhang, W. An, Y.-P. Tan, A. C. Kot, and S. Lu, *Mmrel: A relation understanding dataset and benchmark in the mllm era*, *arXiv preprint arXiv:2406.09121* (2024).

[252] A.-C. Cheng, H. Yin, Y. Fu, Q. Guo, R. Yang, J. Kautz, X. Wang, and S. Liu, *Spatialrgpt: Grounded spatial reasoning in vision language model*, *arXiv preprint arXiv:2406.01584* (2024).

[253] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, *Openai gym*, *arXiv preprint arXiv:1606.01540* (2016).

[254] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu, *Reasoning with language model is planning with world model*, *arXiv preprint arXiv:2305.14992* (2023).

[255] T. Gokhale, S. Sampat, Z. Fang, Y. Yang, and C. Baral, *Blocksworld revisited: Learning and reasoning to generate event-sequences from image pairs*, *arXiv preprint arXiv:1905.12042* (2019).

[256] Anthropic, *The claude 3 model family: Opus, sonnet, haiku*, *Claude-3 Model Card* (2024).

[257] Antrhopic, "Claude 3.7 sonnet and claude code." `https://www.anthropic.com/news/claude-3-7-sonnet`, 2025.

[258] OpenAI, "Hello gpt-4o." https://openai.com/index/hello-gpt-4o/, 2024. OpenAI announcement.

[259] X. Dong, P. Zhang, Y. Zang, Y. Cao, B. Wang, L. Ouyang, X. Wei, S. Zhang, H. Duan, M. Cao, *et. al.*, *Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model*, *arXiv preprint arXiv:2401.16420* (2024).

[260] Z. Lin, C. Liu, R. Zhang, P. Gao, L. Qiu, H. Xiao, H. Qiu, C. Lin, W. Shao, K. Chen, *et. al.*, *Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models*, *arXiv preprint arXiv:2311.07575* (2023).

[261] Meta, "Llama 3.2: Revolutionizing edge ai and vision with open, customizable models." https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/, 2024. Meta AI announcement.

[262] Mistral AI Team, "Announcing pixtral 12b." https://mistral.ai/news/pixtral-12b/, 2024. Mistral AI announcement.

[263] R. Zhang, D. Jiang, Y. Zhang, H. Lin, Z. Guo, P. Qiu, A. Zhou, P. Lu, K.-W. Chang, Y. Qiao, *et. al.*, *Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems?*, in *ECCV*, 2024.

[264] M. Li, N. Su, F. Qu, Z. Zhong, Z. Chen, Y. Li, Z. Tu, and X. Li, *Vista: Enhancing vision-text alignment in mllms via cross-modal mutual information maximization*, *arXiv preprint arXiv:2505.10917* (2025).

[265] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, *Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution*, *arXiv preprint arXiv:2409.12191* (2024).

[266] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, *Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond*, *arXiv preprint arXiv:2308.12966* (2023).

[267] Y. Yao, T. Yu, A. Zhang, C. Wang, J. Cui, H. Zhu, T. Cai, H. Li, W. Zhao, Z. He, *et. al.*, *Minicpm-v: A gpt-4v level mllm on your phone*, *arXiv preprint arXiv:2408.01800* (2024).

[268] W. Wang, Z. Gao, L. Gu, H. Pu, L. Cui, X. Wei, Z. Liu, L. Jing, S. Ye, J. Shao, *et. al.*, *Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency*, *arXiv preprint arXiv:2508.18265* (2025).

[269] K. Team, A. Du, B. Yin, B. Xing, B. Qu, B. Wang, C. Chen, C. Zhang, C. Du, C. Wei, *et. al.*, *Kimi-vl technical report*, *arXiv preprint arXiv:2504.07491* (2025).

[270] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, *et. al.*, *Gemma 3 technical report*, *arXiv preprint arXiv:2503.19786* (2025).

[271] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, *Llava-next: Improved reasoning, ocr, and world knowledge*, 2024.

[272] J. Gao, R. Pi, J. Zhang, J. Ye, W. Zhong, Y. Wang, L. Hong, J. Han, H. Xu, Z. Li, *et. al.*, *G-llava: Solving geometric problem with multi-modal large language model*, *arXiv preprint arXiv:2312.11370* (2023).

[273] S. Lu, Y. Li, Q.-G. Chen, Z. Xu, W. Luo, K. Zhang, and H.-J. Ye, *Ovis: Structural embedding alignment for multimodal large language model*, *arXiv preprint arXiv:2405.20797* (2024).

[274] M. Bigverdi, Z. Luo, C.-Y. Hsieh, E. Shen, D. Chen, L. G. Shapiro, and R. Krishna, *Perception tokens enhance visual reasoning in multimodal language models*, in *CVPR*, 2025.

[275] R. Luo, Y. Li, L. Chen, W. He, T.-E. Lin, Z. Liu, L. Zhang, Z. Song, X. Xia, T. Liu, *et. al.*, *Deem: Diffusion models serve as the eyes of large language models for image perception*, *arXiv preprint arXiv:2405.15232* (2024).

[276] Z. Wang, X. Guo, S. Stoica, H. Xu, H. Wang, H. Ha, X. Chen, Y. Chen, M. Yan, F. Huang, *et. al.*, *Perception-aware policy optimization for multimodal reasoning*, *arXiv preprint arXiv:2507.06448* (2025).

[277] M.-L. Zhang, F. Yin, and C.-L. Liu, *A multi-modal neural geometric solver with textual clauses parsed from diagram*, *arXiv preprint arXiv:2302.11097* (2023).

[278] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, W. Dai, T. Fan, G. Liu, L. Liu, *et. al.*, *Dapo: An open-source llm reinforcement learning system at scale*, *arXiv preprint arXiv:2503.14476* (2025).