University of California
Santa Barbara

Controllable Visual Editing via Natural Language

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

Doctor of Philosophy
in
Computer Science

by

Tsu-Jui Fu

Committee in charge:

Professor William Yang Wang, Chair
Professor Miguel Eckstein
Professor Lei Li
Professor Xin Eric Wang

June 2024

The Dissertation of Tsu-Jui Fu is approved.

Professor Miguel Eckstein

Professor Lei Li

Professor Xin Eric Wang

Professor William Yang Wang, Committee Chair

March 2024

Controllable Visual Editing via Natural Language

by

Tsu-Jui Fu

To my family, whose unconditional love through this journey

Acknowledgements

I first want to express my deep gratitude to my wonderful advisor, William Yang Wang. Five years ago, when I entered UCSB, multimodal research was emerging. His unique perspective guided me into the realm of vision and language. He consistently provided endless research support and encouraged me to explore various topics. Without him, this dissertation would not have been possible.

Secondly, I thank my additional dissertation committee members Miguel Eckstein, Lei Li, and Xin Eric Wang for their insightful feedback. Miguel's comments in the aspect of psychology and brain science always brought forth novel ideas. Lei's focus on practicality led me towards more robust analyses and encouraged the creation of practical value in my research. Eric helped me a lot during my junior years, greatly aiding in the development of my skills in experimental design and paper writing. My Ph.D. journey was a pleasant experience because of them.

Besides my advisors on campus, I am grateful to my internship mentors—Zhe Gan and Yinfei Yang at Apple, Licheng Yu and Sean Bell at Meta, Linjie Li and Lijuan Wang at Azure, and Yale Song and Daniel McDuff at Microsoft. Their invaluable inspiration, support, and guidance have enriched and imbued meaning into my summers.

Throughout my graduate school, it was my fortune to have all my friends, colleagues, and collaborators. Here I would like to thank Arjun Akula, Sugato Basu, Wenhu Chen, Weixi Feng, Scott Grafton, Jing Gu, Xuehai He, Varun Jampani, Shih-Hung Liu, Siqi Liu, Yujie Lu, Pradyumna Narayana, Yi-Lin Tuan, Xinyi Wang, Yilin Wang, Wenhan Xiong, An Yan, Xiyou Zhou, and Wanrong Zhu.

Above all, my deepest thanks to my family for their unconditional love. My parents provided me with a carefree environment that cultivated my passion for learning and enabled me to pursue higher education. My brother brought immense joy and created

cherished memories during my childhood. Appreciate standing for me forever. Lastly, I consider myself incredibly blessed to have my wife, Yen-Hisu, by my side. She possesses a natural optimism that shines through even in the darkest of times. Whenever I feel stressed or frustrated, she can always uncover hope amidst the challenges. Her presence has been the driving force behind my ability to conquer obstacles over the years. I will keep making you proud!

Thesis Statement

I aim to develop text-guided visual editing systems, which bridge the gap between language understanding and visual editing in various forms such as images (step-by-step modification and artistic style transfer), videos (semantics manipulation and video completion), and natural scenarios (3D characters and realistic images). My ultimate goal is to build a generalist AI artist that can comprehend human instructions at any granularity and manipulate visual content with its creativity.

Abstract

Controllable Visual Editing via Natural Language

by

Tsu-Jui Fu

Nowadays, vision editing systems are widely used in daily life. Despite lots of demand, digital design tools such as Photoshop or Premiere require specific prior knowledge and complex operations, which makes it difficult for novices to kick off. In contrast, language is the most natural way of communication. If a system can utilize given instructions and automatically perform related editing actions, it will significantly facilitate accessibility and meet the considerable need. This dissertation presents our research trust in controllable visual editing via natural language, which connects text understanding and visual generation to benefit practical usage.

While data-driven learning has proven effective, gathering numerous pairs of input-result images is still laborious. Moreover, obtaining crucial instructions is equally challenging. To overcome this data scarcity issue, we integrate counterfactual thinking and mimic human iterative editing through self-supervised reasoning. In addition, we study how to perceive style patterns from visual attributes and human emotions, making artistic style transfer more attainable.

Different from static images, processing videos is more challenging due to their dynamic motion with smooth temporal coherence. We then investigate video editing, which should change only the semantics but preserve the scenario. We explore the multi-level conveyance of videos to modify their properties or moving actions. With arbitrary frames, we develop a unified video completion system that can follow the instruction to generate the full video from any time point.

Beyond images and videos, we take a step forward in natural visual manipulation. Specifically, we study two challenging tasks: 3D human generation and instruction-based editing for natural images. We propose an efficient fusion between textual descriptions and visual rendering to produce concrete 3D characters. We leverage latent visual knowledge from large language models to bridge the gap of instruction understanding for image editing. Our efforts shed light on generalizing visual editing to more diverse and practical scenarios. Finally, we summarize the contributions and implications of our work and discuss future directions toward this research field.

Contents

Chapter 1

Introduction

1.1 Overview

Vision is one of the most important ways for humans to experience, convey, and interact with. In our daily works, various visual editing tools are extensively utilized across various domains, from social media and graphic design to video production. These tools benefit our lives by providing us with better visual experiences. However, they all necessitate prior knowledge and proficiency in complex operations to accomplish editing tasks. For example, in the case of Photoshop[1], it may take 12 hours to learn those essential functions and 42 hours for specific topics, along with 84 hours of practice. This requirement forms a considerable barrier to entry, resulting in difficulty for novices to learn and get started. It limits access to a broader audience and further constrains the ability to create and manipulate visual content.

[1] https://www.photoshopbuzz.com/how-long-learn-photoshop

In contrast, language is the most direct form of human communication, offering an intuitive interface for expressing ideas, emotions, and commands. If a system can follow instructions to perform visual editing tasks automatically, it will significantly improve accessibility and controllability. By leveraging natural language, this system effectively eliminates the need for prior knowledge or technical skills. Such text-guided visual editing has the potential to transform the landscape of visual content creation. Thus everyone can unleash their creativity without any limitations.

Text-to-visual generation [1, 2] has presented impressive results nowadays. Instead of generation from scratch, visual editing requires the output to be modified/extended from the input visual content. This requires both visual perception and language understanding to perform the final visual synthesis. Early works [3, 4] can only change limited attributes (*e.g.*, colors of hair or petal) for specific domains (*e.g.*, human face or flower). With large-trained models [5, 6, 7], recent methods [8, 9] enable visual transformations from complete textual descriptions. However, it deviates from the conventional way humans instruct. For instance, rather than the exhaustive "*a girl is wearing a hat walking at the beach*" for the image depicting "*a girl is walking at the beach*", people usually opt for "*give her a hat*". This kind of instruction offers a more straightforward scenario for practical usage and can bring a more convenient interaction.

An image is composed of every single visual element yet presents an overall expression. Editing involves basic adding/removing objects as well as changing the properties of existing elements, such as the color, shape, or straight hair into curly. Apart from local alteration, optimizing the illumination and modifying the global pattern for diverse expressive feelings from the same content structure are essential functions of popular photo filters, which can lead to a more creative and attractive visual appearance. Built upon static images, a dynamic video can present richer temporal motion, including actions and scene transitions. Videos also share similar editing aspects to images, involving object

Figure 1.1: Controllable Image Editing.

replacement and existing property alteration. The addition of the time dimension brings more editing variations and possibilities. For example, changing the ending position of a moving object cannot be accomplished by editing each frame independently. It requires overall adjustments to the video, including gradual directional transitions between each frame while maintaining temporal consistency. Modifying the action contains the same challenge, which should consider yet keep the the remaining scene reasonable. Furthermore, temporal expansion to control future prediction or past rewind from partial frames is also featured to manipulate potential activities in videos. In this dissertation, we investigate various editing aspects and aim to develop visual editing systems via forthright human instruction.

1.2 Controllability via Natural Language

1.2.1 Image Editing

One of the primary challenges of text-guided editing is data scarcity, where collecting input-result image pairs with instructions is costly. Humans can associate possible alternatives to events that have happened already [10]. **Chapter 2** incorporates this counterfactual thinking to overcome data scarcity and follow iterative instructions to edit images step-by-step, as shown in Fig. 1.1. I introduce self-supervised counterfactual

reasoning (SSCR) [11] to imagine expected images under unseen instructions. As no ground-truth results to learn from, the proposed cross-task consistency (CTC) supplies detailed token-level loss without additional training data. I conduct evaluations on i-CLEVR [12] and CoDraw [13] for this iterative text-guided image editing. Experimental results support that SSCR can improve the correctness in both aspects of object identity and position.

In **Chapter 3**, I move forward to modify the overall visual patterns, also from textual descriptions, and introduce language-driven artistic style transfer (`LDAST`) [14]. `LDAST` must preserve the structure of the input image while rendering the style semantics in language, such as "*banded blue*" or "*chaotic and confused*" in Fig. 1.1. I present contrastive language visual artist (CLVA), which learns to extract latent style patterns via patch-wise style discrimination and composes as transferred results. CLVA further boosts by comparing contrastive pairs where relative content images or style instructions should present similar content structures or style patterns. I conduct evaluations on DTD^2 [15] and ArtEmis [16] for the challenging `LDAST`, considering texts of visual attributes and human style feelings. Extensive experiments and qualitative comparisons demonstrate that CLVA can carry out `LDAST` effectively and efficiently.

1.2.2 Video Editing

In contrast to images, videos are more challenging to process due to their dynamic motion and temporal coherence. **Chapter 4** initiates the first language-based video editing (`LBVE`) [17]. As shown in Fig. 1.2, `LBVE` allows changing only the semantics (gesture motion) but preserves the input scenario (the same person and background). Both video and language are multilevel conveyed, where videos are composed of a series of image frames and language is a set of word tokens with a specific order. I introduce multimodal

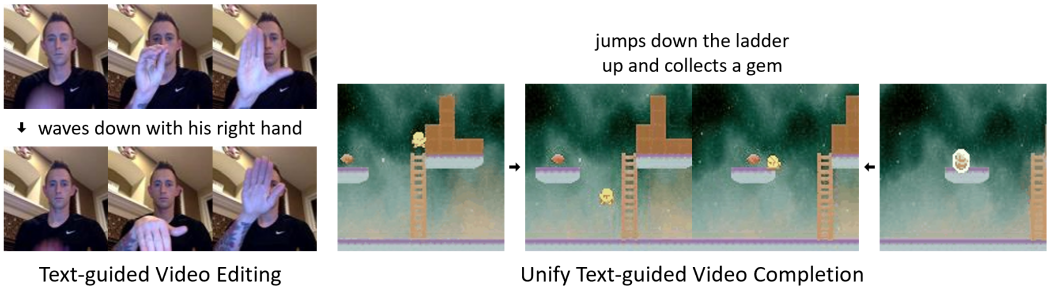


Figure 1.2: Controllable Video Editing.

multi-level transformer (M^3L), which contains the multi-level fusion (MLF) to fuse between the frame-word (local) and video-sentence (global) level. I build three benchmarks for evaluation, consisting of two diagnostic (E-MNIST [18] and E-CLEVR [19]) and one natural (E-JESTER [20]) datasets. The experiments support that M^3L can link video perception with language understanding for `LBVE` in both aspects of content replacement and semantic manipulation.

Video prediction [21, 22], on the other hand, anticipates the future by completing a video from the past frames. However, it may produce various outcomes, which are difficult to meet human expectations. Moreover, humans can imagine what has happened not only from the head but also from the tail, or even from both. **Chapter 5** introduces text-guided video completion (`TVC`) [23], where the partial frames and a given instruction jointly guide the video generation as Fig. 1.2. To tackle `TVC`, I present multimodal masked video generation (MMVG), where the mask-then-recover strategy is adopted for video completion with temporal-aware discrete visual representation. By varying the masking conditions, MMVG learns to generate the full video from frames at arbitrary time points, conditioned on the text. The single-trained model can handle all `TVC` tasks, including prediction, rewind, and infilling. I consider diverse video scenarios for evaluation, such as egocentric (Kitchen [24]), animation (Flintstones [25]), and gaming (MUGEN [26]). Extensive experiments indicate that MMVG can break the limitation of chronological

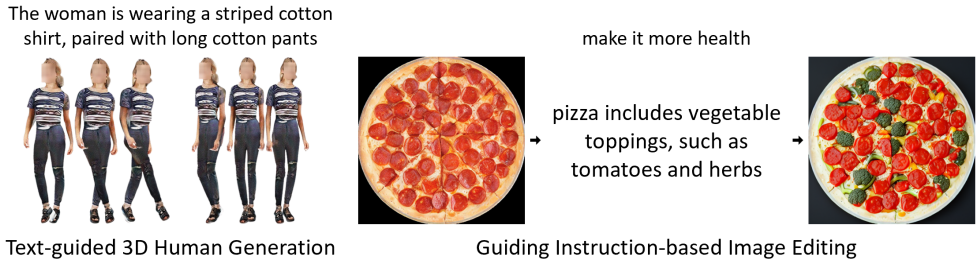


Figure 1.3: Natural Visual Manipulation.

guidance and achieve comprehensive performance boosts.

1.2.3 Natural Visual Manipulation

Beyond images and videos, I take a further step into natural visual manipulation. 3D human modeling has been widely used for engaging interaction in gaming, film, and animation. The customization of these characters is crucial for creativity and scalability, highlighting the importance of controllability. **Chapter 6** introduces text-guided 3D human generation (`T3H`) [27] to generate a 3D human with a customized outfit as Fig. 1.3. To tackle `T3H`, I propose cross-modal human (CCH) to learn 3D from 2D image collections, which divides the human body into different parts and employs individual volume rendering. CCH extracts the fashion semantics from the description and adopts cross-modal attention to fuse body volumes with textual features, enabling each part to learn perceiving its correlated fashion patterns. I perform evaluations on DeepFashion [28] and SHHQ [29], which consist of various types of shapes, fabrics, and colors. The experiments demonstrate the superiority of CCH in both 3D quality and textual relevance.

With promising large-scale training, recent methods can generate photorealistic images from the text prompts [5], which facilitates editing for natural images [30]. However, human instructions are sometimes too brief and insufficient to guide toward the intended goal. For example, in Fig. 1.3, it is difficult to capture what "*health*" means without ad-

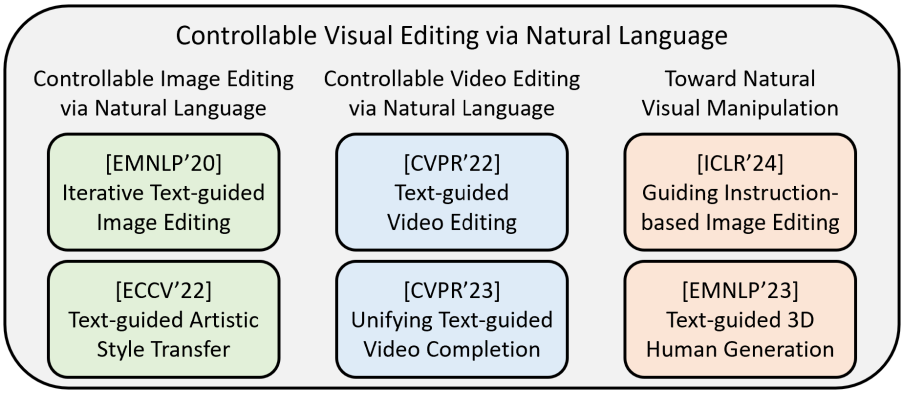


Figure 1.4: My contribution to controllable visual editing via natural language.

ditional context. In **Chapter 7**, I introduce MLLM-guided image editing (MGIE) [31], which leverages multimodal large language models to derive concise expressive instructions, offering explicit guidance. The diffusion model then jointly performs image editing via the latent imagination. In this way, MGIE can associate "*"vegetable toppings*" with the pizza and lead to the result that aligns with the human expectation. I consider various editing aspects in Photoshop-style modification (EVR [32] and GIER [33]), global photo optimization (MA5k [34]), and local object alteration (MagicBrush [35]). Experimental results support that MGIE significantly strengthens the editing performance, along with qualitative comparisons.

1.3 Contributions

Here I summarize my contribution to controllable visual editing via natural language in Fig. 1.4. Throughout my Ph.D., I have been studying these three areas:

- Understanding language as object properties, spatial relations, and visual patterns. I investigated how to perform step-by-step modification [11] and render style semantics from visual attributes or emotional effects [14];
- Perceiving dynamic motions and controlling video semantics by human commands.

I worked on the hierarchical relationship for video editing [17] and video completion from arbitrary guided frames [23];

- Extending to natural visual scenarios, including 3D modeling and natural images, for practical applications. I took the step of establishing 3D characters with fashion descriptions [27] and facilitating instruction-based editing via latent visual knowledge from large language models [31],

which can support the following editing aspects for both images and videos:

Image Aspect	Chapter	Instruction Example
Add/Remove	2 & 7	*add a purple cube behind it*
		edit out skiers on right
Global Pattern	3 & 7	*floating, colorful, white backdrop*
		add contrast to simulate more light
Local Alteration	7	*change the background to blue*
		make the face happy
3D Appearance	6	*she is dressed in a long-sleeved chiffon shirt with striped three-point shorts*

Video Aspect	Chapter	Instruction Example
Object Replacement	4	*change the number from 1 to 3*
Property Alteration	4	*change the red cube into the small purple*
Motion/Action	4 & 5	*rotate and swipe her right hand*
		put plate on counter
Temporal Expansion	5	*jumps down the stage. it runs from left to right and jumps on a worm*

With these efforts, I aim to bridge the gap between language understanding and visual generation, where a system can manipulate visual content with natural instructions from humans and allow everyone to unleash their creativity without limitations.

Part I

Controllable Image Editing via Natural Language

Chapter 2

Iterative Text-guided Image Editing

2.1 Introduction

Iterative language-based image editing (`ILBIE`) task follows iterative instructions to edit images step by step [12], as illustrated in Fig. 2.1. To accomplish `ILBIE`, models are required not only to modify images but also to understand the visual differences between the previous and resulting image, based on the given instructions. One of the primary limitations of `ILBIE` is data scarcity. Since collecting large-scale previous-resulting images with instructions is difficult, it makes learning the association between vision and language challenging.

Despite lacking prior experience with images or instructions, humans can still accomplish editing under unfamiliar image-instruction pairs. For example, for a given instruction "*add a purple cube in front of the blue cylinder*", humans can think about the resulting image if the instruction changed to "*adding a blue square*" or "*on the right of*". This process is known as counterfactual thinking [10], which allows humans to operate in data-scarce scenarios by considering alternative instructions from the seen examples.

In this chapter, we introduce self-supervised counterfactual reasoning (SSCR) that

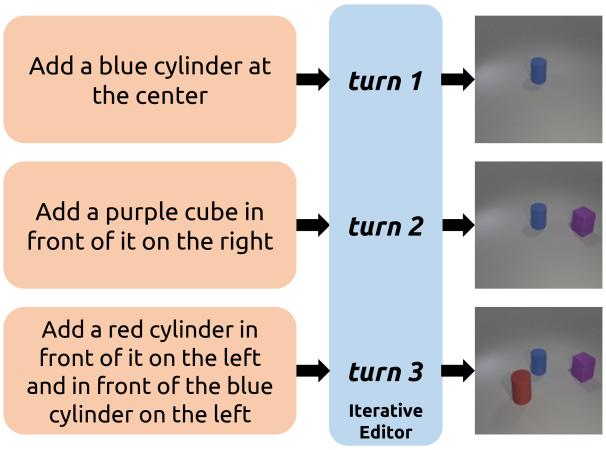


Figure 2.1: The iterative language-based image editing (ILBIE) task. For each turn, the model modifies the image from the previous turn based on the current instruction. Eventually, a desired image is accomplished after iterative editing.

incorporates counterfactual thinking to deal with the data scarcity issue. SSCR allows the model to think about expected resulting images under unseen instructions. Since there are no ground-truth resulting images, we propose cross-task consistency (CTC), which adopts an iterative explainer to reconstruct the instruction of each step. Through CTC, we can supply detailed token-level training loss (*e.g.*, wrong objects or incorrect positions). The experimental results on i-CLEVR [12] and CoDraw [13] demonstrate that our SSCR can improve the correctness of the ILBIE task in both aspects of object identity and position, even under data scarcity.

Dataset	Edit Aspect	Instruction Example
i-CLEVR	Color/Shape/Relation	*Add a purple cube behind it on the right*
CoDraw	Size/Object/Position	*Lower left corner is a small tree trunk*

2.2 Related Work

Text-to-Image (T2I) To generate an image that matches the given instruction, T2I is a challenging yet important task, which has vast potential in practical applications like

art generation or automatic design [36, 37, 38]. With the success of generative adversarial network [39] on the image generation task, several works [1, 40, 41] introduce different GAN-based models to synthesize an image from a text description. Unlike T2I, we focus on image editing, where a model needs to understand the visual difference between two images rather than generating an image from scratch.

Language-based Image Editing (LBIE) PixelTone [42] and Image Spirit [43] edit an image based on the rule-based text description, which only accepts pre-defined instructions and semantic labels. Some studies [3, 4] adopt the conditional GAN model to attend on the instruction and perform LBIE as image colorization. However, image colorization is not truly an editing task since it only supports fixed object templates, and the scene of the image remains the same after editing. In contrast, the editing processes of Photoshop or Illustrator are not accomplished in a single pass. GeNeVA [12] proposes an iterative GAN-based generator to accomplish iterative language-based image editing (`ILBIE`) but neglects the data scarcity issue.

Counterfactual Thinking The human propensity imagines possible alternatives to events that have happened already [10]. People can consider different outcomes from a wide range of conditions and engage in causal reasoning by asking questions like "*What if ...?*" or "*If I had only ...*". Previous works [44, 45] have shown how counterfactual fairness improves the robustness of the model and makes it more explainable. In addition, counterfactual thinking has also been applied to augment training targets [46, 47]. We incorporate counterfactual thinking into the `ILBIE` task that considers counterfactual instructions to deal with the data scarcity issue and improve generalizability.

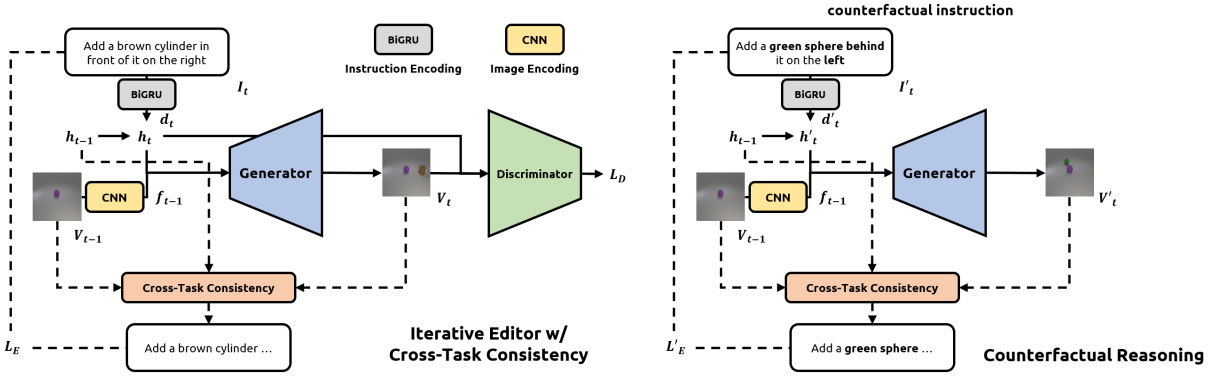


Figure 2.2: The overview of our self-supervised counterfactual reasoning (SSCR). The iterative editor modifies an image based on the current instruction and editing history. Counterfactual reasoning allows the model to think about various counterfactual instructions that can improve the generalizability and deal with data scarcity.

2.3 Self-Supervised Counterfactual Reasoning (SSCR)

2.3.1 Overview

Task Definition During each turn t, an editor edits the image from the previous turn V_{t-1} into the current turn V_t based on the instruction I_t. After a final turn T, we get the predicted final image V_T and evaluate the outcome with the ground truth resulting image O_T. Note that the editing process is at a pixel level where the model has to generate each pixel of the image:

$$\begin{aligned} V_t &= \text{Editor}(V_{t-1}, I_t), \\ \text{eval} &= \text{Compare}(V_T, O_T). \end{aligned} \tag{2.1}$$

To overcome data scarcity, we introduce self-supervised counterfactual reasoning (SSCR). The overall framework is illustrated in Fig. 2.2. The iterative editor is a conditional generator that modifies an image based on the current instruction and editing history. Counterfactual reasoning allows the model to think about the expected resulting images under various counterfactual instructions. Therefore, the editor can consider

more diverse instructions than the original data to improve the generalizability, even if under data scarcity.

2.3.2 Iterative Editor

Similar to GeNeVA [12], the iterative editor is a GAN-based architecture that contains a conditional generator G and a discriminator D. We first apply a bidirectional GRU [48] to encode the instruction I_t as d_t for each turn t. And another GRU is used to encode the history of instructions h_t as following:

$$h_t = \text{GRU}(d_t, h_{t-1}). \tag{2.2}$$

Then, to perform the editing for turn t, we adopt a convolutional neural network (CNN) [49] to extract image features f_{t-1} from the previous image V_{t-1}, concatenate with the instruction history h_t, and feed into G to predict the resulting image V_t:

$$V_t = G([f_{t-1}, h_t]). \tag{2.3}$$

After all iterations, there is the final image V_T after the final turn T. The encoded history h plays a crucial role in bridging contextual coherence (*e.g.*, the referential relationship of "*in front of it*"). Without h, a non-iterative editor cannot perceive which position to place the "*brown cylinder*".

For each turn, D provides a binary training signal by discriminating a resulting image that is generated from either G or the ground-truth data according to the instruction history h_t:

$$\mathcal{L}_G = \sum_{t=1}^{T} \mathbb{E}_{V_t \sim \mathcal{P}_{G_t}}[\log(D([V_t, h_t]))], \tag{2.4}$$

where $\mathcal{L}_G$ is the binary loss from D. For training D, similar to T2I [1], we add additional

[real image, wrong instruction] pairs as false examples:

$$\mathcal{L}_D = \sum_{t=1}^{T} \mathcal{L}_{D_{\text{real}_t}} + \frac{1}{2}(\mathcal{L}_{D_{\text{false}_t}} + \mathcal{L}_{D_{\text{wrong}_t}}), \tag{2.5}$$

where

$$\begin{aligned} \mathcal{L}_{D_{\text{real}_t}} &= \mathbb{E}_{O_t \sim \mathcal{P}_{\text{data}}}[\log(D([O_t, h_t]))], \\ \mathcal{L}_{D_{\text{false}_t}} &= \mathbb{E}_{V_t \sim \mathcal{P}_{G_t}}[\log(1 - D([V_t, h_t]))], \\ \mathcal{L}_{D_{\text{wrong}_t}} &= \mathbb{E}_{O_t \sim \mathcal{P}_{\text{data}}}[\log(1 - D([O_t, h'_t]))], \end{aligned} \tag{2.6}$$

with ground-truth data distribution $\mathcal{P}_{\text{data}}$ and h'_t being the wrong instruction history by randomly selecting another instruction. Then G and D are optimized through an alternating minmax game:

$$\max_G \min_D \mathcal{L}_G + \mathcal{L}_D. \tag{2.7}$$

2.3.3 Cross-Task Consistency (CTC)

Though we can train the iterative editor for `ILBIE`, D only supports a binary training loss, which is not explicit enough to express the complex association between the visual difference and the text description. To supply a more explicit training loss, we propose cross-task consistency (CTC). Despite being image generation, we consider instruction generation, which explains the visual difference between previous-resulting image pairs, to do reasoning for the editing process in a cross-task scenario. During CTC, an iterative explainer provides a token-level training signal that encourages the matching between the predicted image and the original instruction.

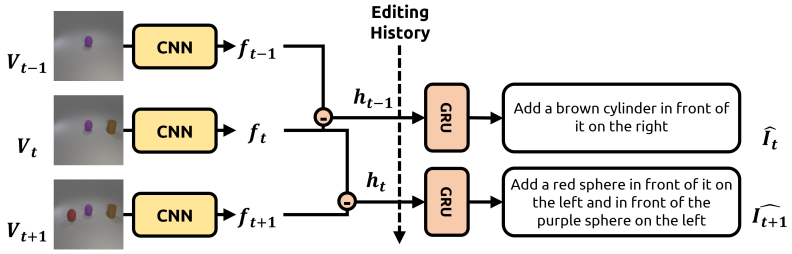


Figure 2.3: The architecture of our iterative explainer. We consider the previous-resulting image pair and the encoded instruction history as the input to reconstruct the editing instruction by an attention-based GRU decoder, which can provide an explicit token-level training signal.

Iterative Explainer Our iterative explainer E is an instruction decoder, which considers the previous-resulting image pair and the instruction history as the input, as shown in Fig. 2.3:

$$\hat{I}_t = E(V_t, V_{t-1}, h_{t-1}). \tag{2.8}$$

Similar to the iterative editor, we apply CNN to extract visual features f for both previous and predicted resulting images:

$$f_{t-1} = \text{CNN}(V_{t-1}), f_t = \text{CNN}(V_t). \tag{2.9}$$

Then, a GRU serves as an attention-based language decoder [50] which reconstructs the instruction $\hat{I}_t$ according to the features difference and instruction history h_{t-1} of the previous turn:

$$\begin{aligned} g_0 &= [f_d, h_{t-1}], \\ \hat{w}_i, g_i &= \text{GRU}(w_{i-1}, g_{i-1}), \\ \hat{I}_t &= \{\hat{w}_1, \hat{w}_2, ..., \hat{w}_L\}, \end{aligned} \tag{2.10}$$

where $f_d = f_t - f_{t-1}$ represents the visual difference by subtracting previous and result features, g_i is the decoding history, and $\hat{w}_i$ is the predicted word token of the instruction.

All w_i are combined as the reconstruction where L is the length of the instruction. The iterative explainer considers not only the visual difference but also the instruction history so that we can reconstruct the instruction, which explains the editing of the resulting image followed by the editing history.

Finally, we provide an explicit token-level training signal $\mathcal{L}_E$ by computing the teacher-forcing loss [51] between the original instruction I_t and the reconstructed one $\hat{I}_t$:

$$\mathcal{L}_E = \sum_{i=1}^{L} \text{CELoss}(\hat{w}_i, w_i), \tag{2.11}$$

where w_i is the ith token of I_t and CELoss means the cross-entropy loss. By minimizing $\mathcal{L}_E$, G learns to match the original instruction with this cross-task consistency. Different from $\mathcal{L}_G$, which only supplies binary but vague loss, $\mathcal{L}_E$ provides token-level loss about the information of the wrong object or wrong position (by comparing $\hat{w}_i$ with w_i) that can train G better for each editing turn. In the experiments, E is pre-trained by the ground-truth image pairs and is fixed during the following training.

2.3.4 Counterfactual Reasoning

We assume that $\mathcal{U}$ is the available training data. Because of the practical challenge of collecting large-scale previous-resulting images with instructions, $\mathcal{U}$ suffers from data scarcity. To deal with this issue, we propose counterfactual reasoning to allow the model to consider various instructions out of the distribution of $\mathcal{U}$. For instance, an instruction $I' \sim \mathcal{U}'$ from the intervention data $\mathcal{U}'$ replaces the original instruction, and we edit the image based on the counterfactual instruction I'.

Dataset	Token Type	Example
i-CLEVR	color	*blue*, *purple*
	object	*cylinder*, *cube*
	relation	*at center*, *in front*
CoDraw	size	*small*, *meidum*
	object	*sun*, *boy*
	relation	*in middle*, *on left*

Table 2.1: The overview of token type on i-CLEVR [12] and CoDraw [13].

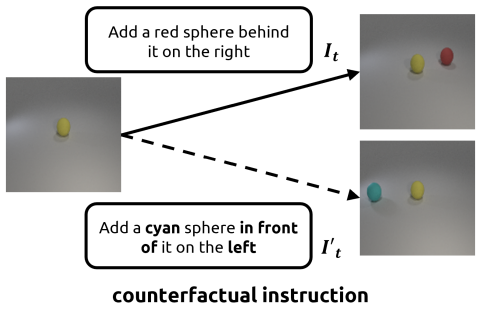


Figure 2.4: An example of instruction intervention for counterfactual reasoning.

Instruction Intervention To get the intervention data U' that provides diverse instructions, we do interventions on the original instructions I:

$$\begin{aligned} I, O &= \mathcal{U}, \\ I' &= \text{intervention}(I), \\ \mathcal{U}' &= \{I', O\}, \end{aligned} \tag{2.12}$$

where O is the image in the original $\mathcal{U}$. First, we apply NLTK [52] to parse out *tokens* in the original I, as summarized in Table 2.1. We then replace these *tokens* with randomly sampled *tokens* of the same type to get the counterfactual I'. Finally, I' combines with the original image G as the intervention data $\mathcal{U}'$. Our experiments show that this simple yet effective intervention makes the training data more diverse and deals with data scarcity during our counterfactual reasoning.

For each turn t, with I'_t from U', we predict the counterfactual resulting image V'_t:

$$V'_t = G([f_{t-1}, h'_t]), \tag{2.13}$$

where h'_t is the counterfactual instruction history encoded from I'. Since there is no ground-truth image for the counterfactual instruction I'_t, we adopt the iterative explainer

Algorithm 1 Iterative Editor with Cross-Task Consistency (CTC)

G, D: Generator / Discriminator
H: Instruction History Encoder
E: Iterative Explainer in Cross-Task Consistency (CTC)

Pre-train E
while TRAIN_EDITOR **do**
  for $t \leftarrow 1$ to T **do**
    I_t, $O_t \leftarrow$ sampled instruction / image
    $h_t = H(h_{t-1}, I_t)$
    $V_t = G(h_t, O_{t-1})$ ▷ teacher-forcing training
    $\hat{I}_t = E(V_t, O_{t-1}, h_{t-1})$

    $\mathcal{L}_G$, $\mathcal{L}_E \leftarrow$ binary / explicit loss ▷ Eq. 2.4 and 2.11
    Update G by maximizing $\mathcal{L}_G - \mathcal{L}_E$
    $\mathcal{L}_D \leftarrow$ discrimination loss ▷ Eq. 2.5
    Update D by minimizing $\mathcal{L}_D$
  end for
end while

E to provide counterfactual training loss L'_E in a self-supervised scenario:

$$\begin{aligned} \hat{I}'_t &= E(V'_t, V_{t-1}, h_{t-1}), \\ \mathcal{L}'_E &= \sum_{i=1}^{L} \text{CELoss}(\hat{w}'_i, w'_i), \end{aligned} \tag{2.14}$$

where $\hat{w}'_i$ and w'_i are the ith word token. By minimizing $\mathcal{L}'_E$, the model has an opportunity to access U', which is different from the original training data. With the help of our iterative explainer, SSCR improves the generalizability by reasoning diverse counterfactual instructions I' even if under data scarcity.

2.3.5 Learning of SSCR

Algo. 1 presents the learning process of training the iterative editor with CTC. Since `ILBIE` is also a sequential generation process, we apply the widely used teacher-forcing where we feed in the ground-truth resulting image (O_{t-1}) from the previous turn instead

of our predicted one (V_{t-1}) to make the training more robust. When training the iterative editor, for each turn t, we adopt G to perform image editing. We maximize the binary loss from D ($\mathcal{L}_G$) with minimizing the explicit token-level loss from E ($\mathcal{L}_E$) to train G. We also update D by minimize $\mathcal{L}_D$:

$$\max_G \min_D \mathcal{L}_G + \mathcal{L}_D - \mathcal{L}_E. \tag{2.15}$$

During counterfactual reasoning, we first perform an intervention on $\mathcal{U}$ to get the counterfactual instructions (I'). Then, we edit the image based on I'. Since there is no ground-truth resulting image for the counterfactual editing, we adopt CTC to compute the cycle-consistency loss (L'_E) self-supervisedly. Similar to the iterative editor, we also apply teacher-forcing training (feeding in O_{t-1} and h_{t-1}) to further update G. In this way, G can improve the generalizability by considering the counterfactual U', which is more diverse than U.

2.4 Experiments

2.4.1 Experimental Setup

Datasets We evaluate our counterfactual framework on two `ILBIE` datasets, i-CLEVR [12] and CoDraw [13]. Each example (6K in total) in i-CLEVR has a sequence of 5 (image, instruction) pairs. The instruction describes where the object should be placed relative to existing objects. CoDraw is a more difficult art-like dataset of children playing in a park. There are 58 objects and children with different poses from the 8K training examples.

Evaluation Metrics Following GeNeVA [12], we adopt F1 and RelSim to evaluate the editing result. The F1 score is based on a pre-trained object detector [53] (99%

	i-CLEVR				CoDraw			
Method	Precision↑	Recall↑	F1↑	RelSim↑	Precision↑	Recall↑	F1↑	RelSim↑
GeNeVA [12]	71.01	42.61	53.26	30.66	54.38	54.42	54.40	<u>38.93</u>
w/ CTC only	<u>72.24</u>	<u>45.51</u>	<u>55.84</u>	<u>33.67</u>	<u>57.69</u>	<u>55.60</u>	<u>56.62</u>	38.68
w/ SSCR	**73.75**	**46.39**	**56.96**	**34.54**	**58.17**	**56.61**	**57.38**	**39.11**

Table 2.2: The testing results of the baseline (GeNeVA), with only cross-task consistency (CTC only), and with self-supervised counterfactual reasoning (SSCR).

accuracy), which detects the objects in the predicted images that meet the ground-truth resulting images. To evaluate not only object type but also object position, we build the scene graph according to the object detector. The edges are given by the left-right and front-back relations between the vertices (objects). Then, RelSim determines how many of the ground-truth relations are in the predicted images:

$$\text{RelSim}(E_{\text{gt}}, E_{\text{pd}}) = \text{recall} \times \frac{|E_{\text{pd}} \cap E_{\text{gt}}|}{|E_{\text{gt}}|}, \tag{2.16}$$

where E_{gt} and E_{pd} are relational edges for ground-truth resulting images and predicted images. Note that we only evaluate the final predicted image of each example for both F1 and RelSim.

Baselines We use the SOTA model GeNeVA [12] as our baseline. It shares the same model architecture as our iterative editor and is trained with the GAN objective but without the cross-task consistency (CTC) and our counterfactual reasoning.

Implementation Detail We apply the ResBlocks [49] into G and D where the visual features size is 1024. For our E, we add self-attention [54] for the concatenation of the visual difference and the encoded instruction history. We adopt Adam [55] to optimize the iterative editor with the learning rate 1e-4 for $\mathcal{L}_G$ and $\mathcal{L}_E$, 4e-4 for $\mathcal{L}_D$. The learning

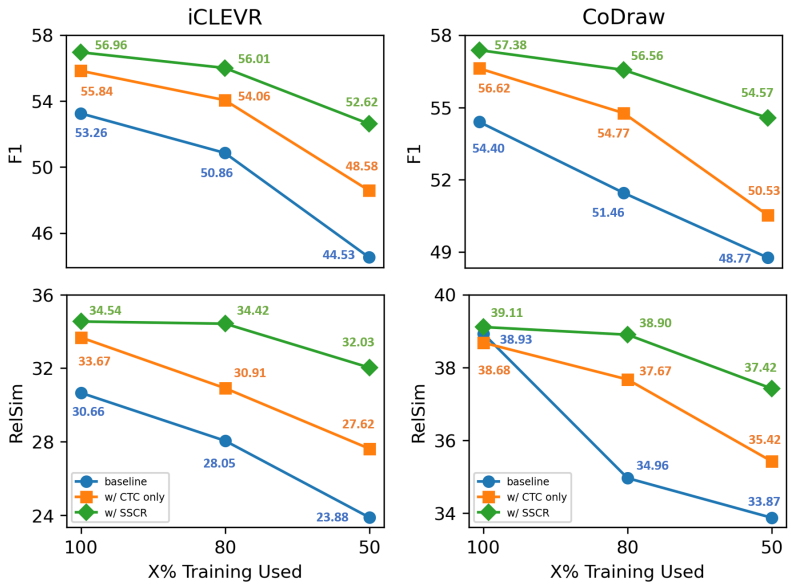


Figure 2.5: The result comparison under different ratios of training data. Note that the iterative explainer is also pre-trained from the same available data for each result.

rate of $\mathcal{L}'_E$ during the counterfactual reasoning is 5e-5.

2.4.2 Quantitative Results

Table 2.2 presents the testing results on both i-CLEVR and CoDraw. First, with our cross-task consistency (CTC only), which provides a more explicit training signal, we can improve the baseline in terms of all metrics. Additionally, for whole self-supervised counterfactual reasoning (SSCR), which allows the model to consider out-of-distribution instructions, it brings more improvements and achieves new SOTA results (*e.g.*, 56.9 F1 and 34.5 RelSim on i-CLEVR).

Similar trends can be found on CoDraw. Since the instructions under CoDraw are more complex, the improvement of relation correctness (RelSim) is not as high as i-CLEVR. But for object correctness, CTC still improves the baseline, and SSCR further achieves the new SOTA on all metrics (*e.g.*, 57.4 F1 and 39.1 RelSim).

X% Data	PPL↓	BLEU↓
100%	0.1073	50.236
80%	0.1295	48.873
50%	0.1163	48.763

Table 2.3: The PPL and BLEU of our iterative explainer with different ratios of training data used on i-CLEVR.

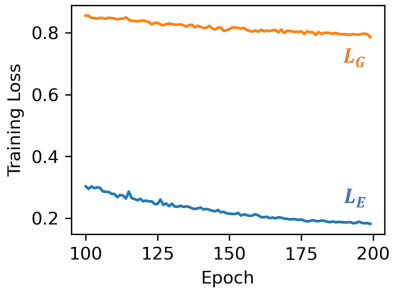


Figure 2.6: The learning curve of training loss provided from the discriminator ($\mathcal{L}_G$) and our iterative explainer ($\mathcal{L}_E$) on i-CLEVR.

Under Data Scarcity To examine the framework's effectiveness under the data scarcity scenario, we compare models trained using 100%, 80%, and 50% data. Note that our E is also pre-trained using the same amount of data. The results are shown in Fig. 2.5. We can observe that on both i-CLEVR and CoDraw datasets, the baseline performance drops drastically as the training data decreases, and our SSCR consistently outperforms the baseline. More importantly, the baseline severely suffers from the data scarcity issue, while SSCR is relatively resilient to data decrease and only drops 4.3 F1 score and 2.5 RelSim score (*vs.* 8.7 and 6.7 reduced by the baseline) on i-CLEVR when there is only 50% data. Similar results can be observed on CoDraw.

Table 2.3 presents the performance of our iterative explainer E with different ratios of training examples. Perplexity (PPL) and BLEU [56] are calculated between the reconstructed instructions and the original ones. We can see that the PPL and BLEU under 50% are similar to 100%. It shows that E still supplies meaningful training loss for SSCR even if only using 50% of data.

2.4.3 Ablation Study

Iterative Explainer versus Discriminator Fig. 2.6 shows the learning curve of the training loss of the discriminator D ($\mathcal{L}_G$) and our iterative explainer E ($\mathcal{L}_E$). The relative

	100%		50%	
Method	F1↑	RelSim↑	F1↑	RelSim↑
GeNeVA [12]	53.26	30.66	44.53	23.88
w/ SSCR (D)	54.05	30.87	43.31	22.99
w/ SSCR (E)	**56.95**	**34.54**	**52.62**	**32.03**

Table 2.4: The comparison between the discriminator D and the iterative explainer E used for self-supervised counterfactual reasoning (SSCR) on i-CLEVR.

decrease of $\mathcal{L}_G$ over time is very little, which means that D can barely provide an extra training signal after 100 epochs. In contrast, since E can supply explicit token-level loss instead of vague binary loss, $\mathcal{L}_E$ keeps decreasing to train the model.

Table 2.4 presents the comparison when using the discriminator D or our iterative explainer E to provide the training loss during the counterfactual reasoning. Since there are no ground-truth resulting images of counterfactual instructions, D can only provide training loss by discriminating them as false. Therefore, D for SSCR cannot improve the model effectively and may even hurt the generalizability under data scarcity (*e.g.*, 23.9 drops to 23.0 RelSim when using 50% of data).

Zero-shot Generalization There involves 3 shapes (*cube*, *sphere*, and *cylinder*) and 8 colors (*gray*, *red*, *blue*, *green*, *brown*, *purple*, *cyan*, and *yellow*), which lead to 24 different objects in the i-CLEVR dataset. We remove examples containing *gray cube*, *red cube*, *green sphere*, or *purple cylinder* in the training set but still evaluate the full testing set with all kinds of objects. The zero-shot results are shown in Table 2.7. Since there is no example like "*gray cube*" in the training set, CTC can only consider those seen objects and improve marginally. In contrast, the iterative explainer E can disentangle color and shape information from "*gray sphere*" and "*green cube*", and generalize to the unseen "*gray cude*". Hence SSCR can still bring out obvious improvements from the provided self-supervised loss.

Method	F1↑	RelSim↑
GeNeVA [12]	42.23	23.70
w/ CTC Only	<u>43.91</u>	<u>25.26</u>
w/ SSCR	**48.30**	**29.09**

Figure 2.7: The zero-shot results on i-CLEVR.

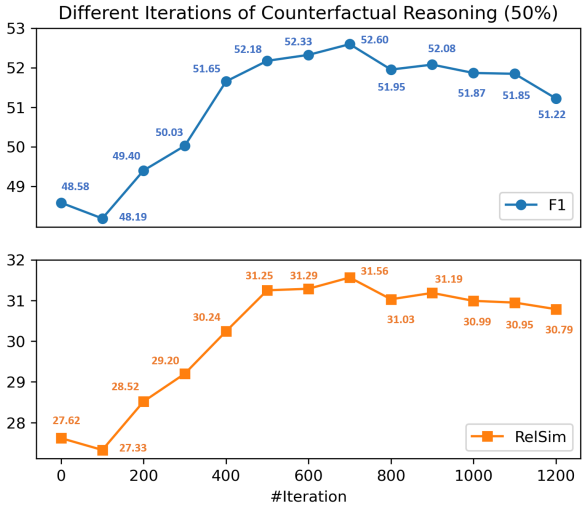


Figure 2.8: The testing results of different counterfactual reasoning iterations on i-CLEVR when using 50% of data.

Counterfactual Reasoning: the More the Better? Fig. 2.8 illustrates the performance of different counterfactual reasoning iterations (on 50% of training data). Despite allowing the model to explore various instructions, excessive iterations (possibly due to the imperfect function) may result in overfitting and degrade its effectiveness.

Limitation of Counterfactual Reasoning Since our counterfactual reasoning relies on instruction intervention to consider diverse unseen pairs, token-level substitution for complex instructions remains challenging and may even disrupt their grammatical structure. This will diminish the effectiveness of our SSCR and result in a limited performance boost. A more feasible approach to paraphrasing natural language is worth exploring to maximize the benefits of our framework.

2.4.4 Qualitative Results

Fig. 2.9 demonstrates an example of the iterative editing on CoDraw. Since GeNeVA only has a discriminator to provide vague loss, it makes the predicted images low quality,

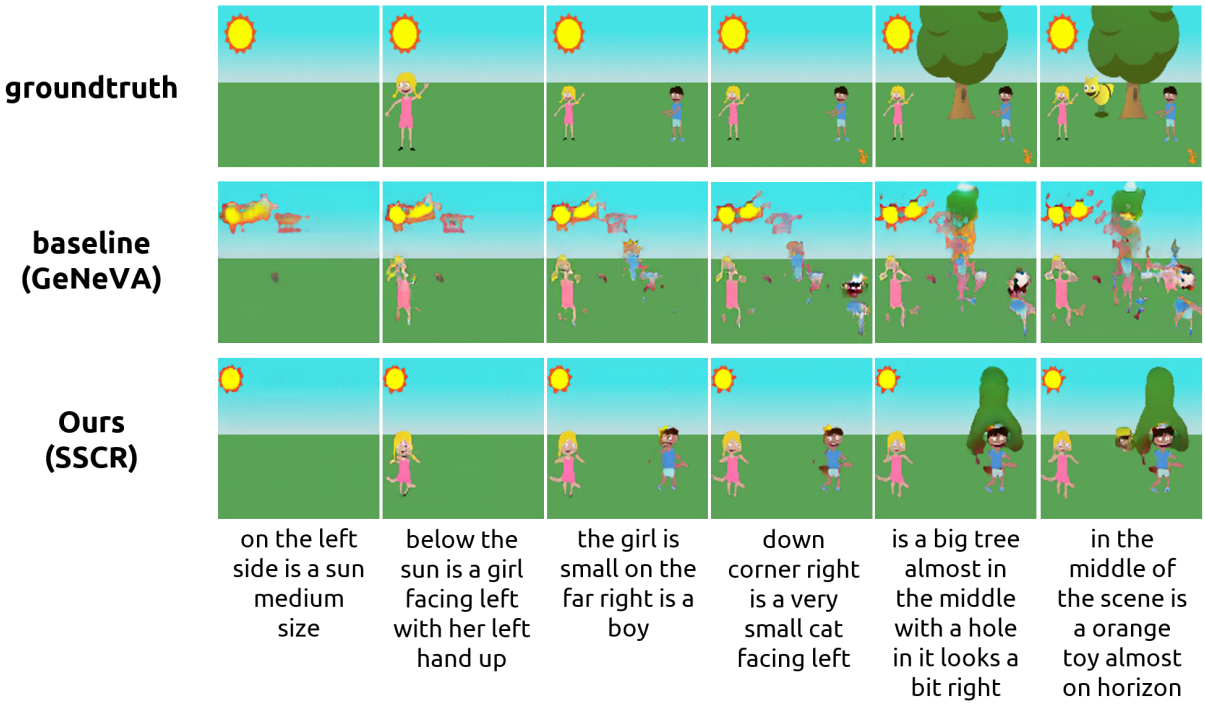


Figure 2.9: The qualitative comparison between the baseline and our SSCR on CoDraw.

where the pixels are almost broken. In contrast, for our SSCR, CTC can help train the generator better, which leads to defined objects. Furthermore, counterfactual reasoning also makes the predicted images more aligned with the instructions.

2.5 Summary

In this chapter, we investigate the data scarcity issue of iterative language-based image editing. We present self-supervised counterfactual reasoning, which leverages counterfactual thinking to allow considering unseen instruction-image pairs. We propose cross-task consistency to provide a more explicit training signal and train counterfactual examples in a self-supervised scenario. Experimental results support that our framework not only leads to a better image editor but also improves generalizability, even with less data.

Chapter 3

Text-guided Artistic Style Transfer

3.1 Introduction

Artistic style transfer [57, 58, 59, 60, 61] adopts appearances and visual patterns from another reference style images to manipulate a content image, which has a considerable application value for creative visual design. However, it requires preparing collections of style images in advance. It even needs to redraw new references first if there is no expected style image. In this chapter, we introduce language-driven artistic style transfer (`LDAST`). As illustrated in Fig. 3.1, `LDAST` treats a content image and a text as the input, and the style transferred result is manipulated based on the style description. It should preserve the structure of the content yet simultaneously modify the style pattern that corresponds to the instruction, which refers to characteristic visual attributes or aesthetic features. The textual description can contain specific visual attributes (*e.g.*, color and texture) and abstract emotional effects (*e.g.*, human feeling). For example, it has to connect "*water, sketching, and painting*" or "*peaceful, feel content*" with their exact visual concepts.

We present contrastive language visual artist (CLVA), including language visual artist (LVA) and contrastive reasoning (CR), to perform style transfer conditioning on guided

Figure 3.1: The language-driven artistic style transfer (LDAST) task, which performs style transfer for a content image, guided by a style instruction.

texts. LVA preserves content structures from content images and extracts visual semantics from style instructions. LVA learns the latent style pattern based on the distinguishment between patches of style images or transferred results from the patch-wise style discriminator. Furthermore, CR boosts LDAST by comparing contrastive pairs where relative content images or style instructions should present similar content structures or style patterns.

We conduct experiments on DTD^2 [15] and ArtEmis [16]. We treat their annotations as style instructions for the challenging LDAST concerning visual attributes (DTD^2) or human-style feelings (ArtEmis). The experiments support that our CLVA is effective for LDAST and achieves superb yet efficient transferred results on both automatic metrics and human evaluation.

Dataset	Style Aspect	Instruction Example
DTD^2	Visual Attribute	*floating, colorful, white backdrop*
		grayish bluish green smeared paint
ArtEmis	Emotional Effect	*bright soft colors reminds me of sunset*
		vibrant trees seems like they are alive

3.2 Related Work

Artistic Style Transfer In general, style transfer [57, 62] can be divided into two categories: *photorealistic* and *artistic*. Photorealistic style transfer [63, 64, 65] aims at applying reference styles on scenes without hurting details and satisfying contradictory objectives. By contrast, artistic style transfer [58, 59, 60] captures style concepts from reference and modifies color distributions and texture patterns of content images. However, it requires preparing numerous style images in advance, which limits the practicality of style transfer. To tackle this issue, `LDAST` allows following textual descriptions to perform *artistic* style transfer and improves the accessibility of visual effect (VFX) design.

CLIP-guided Optimization Based on the powerful visual-linguistic connection of CLIP [66], CLIP-guided image synthesis [67, 68] has shown promising results. StyleCLIP [69] and Style-NADA [70] tweak the latent code of a pre-trained StyleGAN [71] for image editing. Since heavily relying on a pre-trained generator, both are confined to the training domain, and the results can only present limited stylization. CLIPstyler [72] updates the style transfer network for target style patterns from the CLIP alignment but still requires hundreds of iterations and takes lots of time with considerable GPU memory, suffering from the efficiency and practicality overhead.

3.3 Contrastive Language Visual Artist (CLVA)

3.3.1 Overview

Task Definition For the training of `LDAST`, we have pairs of style images $\mathcal{S}$ with style instructions $\mathcal{X}$ to learn the mutual correlation. During testing, only $\mathcal{X}$ are provided for `LDAST` to carry out artistic style transfer purely relied on language.

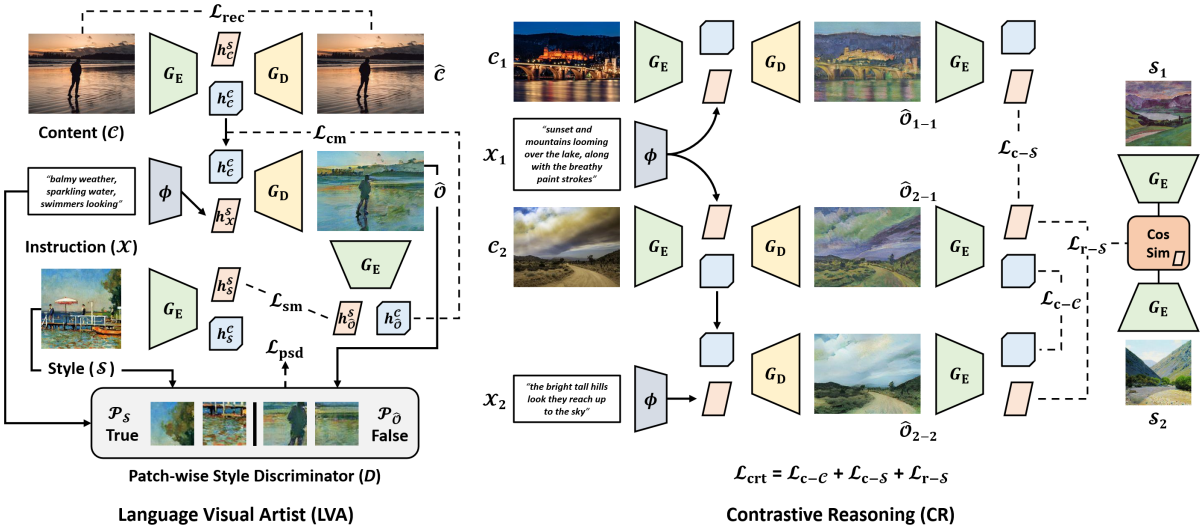


Figure 3.2: The overview of our contrastive language visual artist (CLVA). Language visual artist learns to jointly embed style images and style instructions by the patch–wise style discriminator and perform LDAST for content images. Contrastive reasoning compares contrastive pairs to improve the relativeness between transferred results.

We present contrastive language visual artist (CLVA) in Fig. 3.2. Language visual artist (LVA) extracts content structures from $\mathcal{C}$ and visual patterns from $\mathcal{X}$ to perform LDAST. Contrastive reasoning (CR) allows comparing contrastive pairs of content image and style instruction. In this way, it should present consistent content structures from the same content image or analogous style patterns from related style images, despite using different style instructions.

3.3.2 Language Visual Artist (LVA)

To tackle LDAST, language visual artist (LVA) first adopts visual encoder G_{E} to extract the content features $h^{\mathcal{C}}$ and the style features $h^{\mathcal{S}}$ for an image. Text encoder ϕ also extracts the style instruction features $h^{\mathcal{S}}_{\mathcal{X}}$ from an instruction. $h^{\mathcal{C}}$ is a spatial tensor containing the content structure features, and $h^{\mathcal{S}}$ represents the global style pattern. $\mathcal{S}^{S}_{\mathcal{X}}$ embeds into the same space of $h^{\mathcal{S}}$ to reflect the extracted visual semantic. Then, visual decoder G_{D} produces transferred results $\hat{\mathcal{O}}$ from $h^{\mathcal{C}}_{\mathcal{C}}$ and $h^{\mathcal{S}}_{\mathcal{X}}$, which performs style transfer

by style instructions:

$$\begin{aligned} h_{\mathcal{C}}^{\mathcal{C}}, h_{\mathcal{C}}^{\mathcal{S}} &= G_{\mathrm{E}}(\mathcal{C}), \quad h_{\mathcal{X}}^{\mathcal{S}} = \phi(\mathcal{X}), \\ \hat{\mathcal{O}} &= G_{\mathrm{D}}(h_{\mathcal{C}}^{\mathcal{C}}, h_{\mathcal{X}}^{\mathcal{S}}). \end{aligned} \tag{3.1}$$

There are two goals for LVA: (i) preserving *content structures* from content images; (ii) presenting *style patterns* correlated with visual semantics of style instructions.

Structure Reconstruction To preserve content structures, we consider that visual decoder G_{D} should be able to reconstruct input content images using extracted content features $h_{\mathcal{C}}^{\mathcal{C}}$ and style features $h_{\mathcal{C}}^{\mathcal{S}}$ from visual encoder G_{E}:

$$\begin{aligned} \hat{\mathcal{C}} &= G_{\mathrm{D}}(h_{\mathcal{C}}^{\mathcal{C}}, h_{\mathcal{C}}^{\mathcal{S}}), \\ \mathcal{L}_{\mathrm{rec}} &= ||\hat{\mathcal{C}} - \mathcal{C}||_2, \end{aligned} \tag{3.2}$$

where the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$ is computed as the mean L2 difference between reconstructed content images $\hat{\mathcal{C}}$ and input content images $\mathcal{C}$.

Patch-wise Style Discriminator (D) Regarding style patterns, results $\hat{\mathcal{O}}$ guided by style instructions $\mathcal{X}$ are expected to present analogously to reference style images $\mathcal{S}$. Inspired by texture synthesis [73, 74], images with analogous patch patterns should appear perceptually similar texture patterns. The patch-wise style discriminator D tries to recognize the correspondence between an image patch $\mathcal{P}$ and a style instruction $\mathcal{X}$:

$$\begin{aligned} \mathcal{P}_{\hat{\mathcal{O}}}, \mathcal{P}_{\mathcal{S}} &= \mathrm{Crop}(\hat{\mathcal{O}}), \mathrm{Crop}(\mathcal{S}), \\ \mathcal{L}_D &= \log(1 - D(\mathcal{P}_{\hat{\mathcal{O}}}, \mathcal{X})) + \log(D(\mathcal{P}_{\mathcal{S}}, \mathcal{X})), \\ \mathcal{L}_{\mathrm{psd}} &= \log(1 - D(\mathcal{P}_{\hat{\mathcal{O}}}, \mathcal{X})), \end{aligned} \tag{3.3}$$

where `Crop` is to randomly crop an image into patches. By the discriminator loss $\mathcal{L}_D$, D learns to distinguish that a patch $\mathcal{P}$ is from style images or transferred results. Contrarily, to address the connection between linguistic and visual semantics, the patch-wise style loss $\mathcal{L}_{\text{psd}}$ aims at generating transferred results that are correlated with $\mathcal{X}$.

Content Matching and Style Matching To further enhance the alignment with inputs, inspired by cycle consistency [75, 76, 77], we consider the content matching loss $\mathcal{L}_{\text{cm}}$ and the style matching loss $\mathcal{L}_{\text{sm}}$. We adopt G_{E} again to extract content features $h^{\mathcal{C}}_{\hat{\mathcal{O}}}$ and style features $h^{\mathcal{S}}_{\hat{\mathcal{O}}}$ for $\hat{\mathcal{O}}$, where $h^{\mathcal{C}}_{\hat{\mathcal{O}}}$ and $h^{\mathcal{S}}_{\hat{\mathcal{O}}}$ should correlate with $h^{\mathcal{C}}_{\mathcal{C}}$ from $\mathcal{C}$ and $h^{\mathcal{S}}_{\mathcal{S}}$ from $\mathcal{S}$:

$$\begin{aligned}(h^{\mathcal{C}}_{\hat{\mathcal{O}}}, h^{\mathcal{S}}_{\hat{\mathcal{O}}}), (h^{\mathcal{C}}_{\mathcal{S}}, h^{\mathcal{S}}_{\mathcal{S}}) &= G_{\text{E}}(\hat{\mathcal{O}}), G_{\text{E}}(S), \\ \mathcal{L}_{\text{cm}}, \mathcal{L}_{\text{sm}} &= ||h^{\mathcal{C}}_{\hat{\mathcal{O}}} - h^{\mathcal{C}}_{\mathcal{C}}||_2, ||h^{\mathcal{S}}_{\hat{\mathcal{O}}} - h^{\mathcal{S}}_{\mathcal{S}}||_2.\end{aligned} \tag{3.4}$$

Hence transferred results are required to align with content structures and style patterns from inputs, which meets the goal of `LDAST`.

3.3.3 Contrastive Reasoning (CR)

The content image should transfer to various styles while preserving the same structure. Related style instructions can apply analogous style patterns to arbitrary content images. As shown in Fig. 3.2, contrastive reasoning (CR) compares content structures ($\mathcal{C}_1$ and $\mathcal{C}_2$) or style patterns ($\{\mathcal{S}_1, \mathcal{X}_1\}$, and $\{\mathcal{S}_2, \mathcal{X}_2\}$) from transferred results of contrastive pairs. We follow the LVA inference to acquire cross results for pairs of content images

and style instructions:

$$
\begin{aligned}
(h^{\mathcal{C}}_{\mathcal{C}_1}, h^{\mathcal{S}}_{\mathcal{C}_1}), (h^{\mathcal{C}}_{\mathcal{C}_2}, h^{\mathcal{S}}_{\mathcal{C}_2}) &= G_{\mathrm{E}}(\mathcal{C}_1), G_{\mathrm{E}}(\mathcal{C}_2), \\
h^{\mathcal{S}}_{\mathcal{X}_1}, h^{\mathcal{S}}_{\mathcal{X}_2} &= \phi(\mathcal{X}_1), \phi(\mathcal{X}_2), \\
\hat{\mathcal{O}}_{\mathcal{C}_1\text{-}\mathcal{X}_1}, \hat{\mathcal{O}}_{\mathcal{C}_1\text{-}\mathcal{X}_2} &= G_{\mathrm{D}}(h^{\mathcal{C}}_{\mathcal{C}_1}, h^{\mathcal{S}}_{\mathcal{X}_1}), G_{\mathrm{D}}(h^{\mathcal{C}}_{\mathcal{C}_1}, h^{\mathcal{S}}_{\mathcal{X}_2}), \\
\hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_1}, \hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_2} &= G_{\mathrm{D}}(h^{\mathcal{C}}_{\mathcal{C}_2}, h^{\mathcal{S}}_{\mathcal{X}_1}), G_{\mathrm{D}}(h^{\mathcal{C}}_{\mathcal{C}_2}, h^{\mathcal{S}}_{\mathcal{X}_2}).
\end{aligned} \tag{3.5}
$$

Consistent Matching Transferred results should present similar content structures ($\hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_1}$ and $\hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_2}$) or analogous style patterns ($\hat{\mathcal{O}}_{\mathcal{C}_1\text{-}\mathcal{X}_1}$ and $\hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_1}$) if using the same content image ($\mathcal{C}_2$) or the same style instruction ($\mathcal{X}_1$):

$$
\begin{aligned}
h^{\mathcal{C}}_{\hat{\mathcal{O}}_{\mathcal{C}_i\text{-}\mathcal{X}_j}} &= G_{\mathrm{E}}(\hat{\mathcal{O}}_{\mathcal{C}_i\text{-}\mathcal{X}_j}), \\
\mathcal{L}_{\mathrm{c}-\mathcal{C}} &= ||h^{\mathcal{C}}_{\hat{\mathcal{O}}_{\mathcal{C}_1\text{-}\mathcal{X}_1}} - h^{\mathcal{C}}_{\hat{\mathcal{O}}_{\mathcal{C}_1\text{-}\mathcal{X}_2}}||_2 + ||h^{\mathcal{C}}_{\hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_1}} - h^{\mathcal{C}}_{\hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_2}}||_2, \\
\mathcal{L}_{\mathrm{c}-\mathcal{S}} &= ||h^{\mathcal{S}}_{\hat{\mathcal{O}}_{\mathcal{C}_1\text{-}\mathcal{X}_1}} - h^{\mathcal{S}}_{\hat{\mathcal{S}}_{2-1}}||_2 + ||h^{\mathcal{S}}_{\hat{\mathcal{O}}_{\mathcal{C}_1\text{-}\mathcal{X}_2}} - h^{\mathcal{S}}_{\hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_2}}||_2,
\end{aligned} \tag{3.6}
$$

where *consistent matching* of content structure $\mathcal{L}_{\mathrm{c}-\mathcal{C}}$ or style pattern $\mathcal{L}_{\mathrm{c}-\mathcal{S}}$ is aligned by content features or style features, extracted by G_{E}.

Relative Matching Apart from consistent matching, distinct style instructions, which imply corresponding visual semantics, should still present relative patterns. For example, we can only discover "*reach up to the sky*" literally from $\mathcal{X}_2$. If comparing reference style images $\mathcal{S}_1$ and $\mathcal{S}_2$, we can perceive the sharing of a similar style pattern and link the visual concept of "*bright tall hills*" in $\mathcal{X}_2$ to "*mountains looming over the lake*" in $\mathcal{X}_1$. We define *relative matching* $\mathcal{L}_{\mathrm{r}-\mathcal{S}}$ with the cosine similarity (CosSim) between reference

Algorithm 2 Language Visual Artist (LVA)

G_{E}, G_{D}, ϕ: Visual Encoder / Visual Decoder / Text Encoder
D: Patch-wise Style Discriminator

while TRAIN_LVA **do**
  $\mathcal{C}$, $\{\mathcal{S}, \mathcal{X}\} \leftarrow$ sampled content / style
  $h_{\mathcal{C}}^{\mathcal{C}}$, $h_{\mathcal{C}}^{\mathcal{S}} = G_{\mathrm{E}}(\mathcal{C})$ $\hat{\mathcal{C}} = G_{\mathrm{D}}(h_{\mathcal{C}}^{\mathcal{C}}, h_{\mathcal{C}}^{\mathcal{S}})$
  $\mathcal{L}_{\mathrm{rec}} \leftarrow$ reconstruction loss ▷ Eq. 3.2
  $h_{\mathcal{X}}^{\mathcal{S}} = \phi(\mathcal{X})$ $\hat{\mathcal{O}} = G_{\mathrm{D}}(h_{\mathcal{C}}^{\mathcal{C}}, h_{\mathcal{X}}^{\mathcal{S}})$
  $\mathcal{P}_{\mathcal{S}}$, $\mathcal{P}_{\hat{\mathcal{O}}} = \mathrm{Crop}(\mathcal{S})$, $\mathrm{Crop}(\hat{\mathcal{O}})$
  $\mathcal{L}_{\mathrm{psd}} \leftarrow$ patch-wise style loss ▷ Eq. 3.3
  $(h_{\hat{\mathcal{O}}}^{\mathcal{C}}, h_{\hat{\mathcal{O}}}^{\mathcal{S}})$, $(h_{\mathcal{S}}^{\mathcal{C}}, h_{\mathcal{S}}^{\mathcal{S}}) = G_{\mathrm{E}}(\hat{\mathcal{O}})$, $G_{\mathrm{E}}(\mathcal{S})$
  $\mathcal{L}_{\mathrm{cm}}$ / $\mathcal{L}_{\mathrm{sm}} \leftarrow$ content / style matching loss ▷ Eq. 3.4

  $\mathcal{L}_G = \mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{psd}} + \mathcal{L}_{\mathrm{cm}} + \mathcal{L}_{\mathrm{sm}}$
  Update G_{E}, G_{D}, ϕ by minimizing $\mathcal{L}_G$
  $\mathcal{L}_D \leftarrow$ discrimination loss ▷ Eq. 3.3
  Update D by maximizing $\mathcal{L}_D$
end while

style images:

$$
\begin{aligned}
(h_{\mathcal{S}_i}^{\mathcal{C}}, h_{\mathcal{S}_i}^{\mathcal{S}}) &= G_{\mathrm{E}}(\mathcal{S}_i), \\
r &= \mathrm{CosSim}(h_{\mathcal{S}_1}^{\mathcal{S}}, h_{\mathcal{S}_2}^{\mathcal{S}}), \\
\mathcal{L}_{\mathrm{r}-\mathcal{S}} &= (||h_{\hat{\mathcal{O}}_{\mathcal{C}_1\text{-}\mathcal{X}_1}}^{\mathcal{S}} - h_{\hat{\mathcal{O}}_{\mathcal{C}_1\text{-}\mathcal{X}_2}}^{\mathcal{S}}||_2 + \\
&\quad ||h_{\hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_1}}^{\mathcal{S}} - h_{\hat{\mathcal{O}}_{\mathcal{C}_2\text{-}\mathcal{X}_2}}^{\mathcal{S}}||_2) \cdot r.
\end{aligned}
\tag{3.7}
$$

When style images are related, it has to align style features to a certain extent even if paired style instructions are different. Otherwise, $\mathcal{L}_{\mathrm{r}-\mathcal{S}}$ will be close to 0 and ignore this unrelated style pair. The overall contrastive reasoning loss $\mathcal{L}_{\mathrm{ctr}}$ considers both *consistent matching* and *relative matching*:

$$
\mathcal{L}_{\mathrm{ctr}} = \mathcal{L}_{\mathrm{c}-\mathcal{C}} + \mathcal{L}_{\mathrm{c}-\mathcal{S}} + \mathcal{L}_{\mathrm{r}-\mathcal{S}}. \tag{3.8}
$$

3.3.4 Learning of CLVA

For each epoch of CLVA training, we first train with the LVA process and then CR. As Algo. 2, we consider reconstruction loss $\mathcal{L}_{\text{rec}}$ to preserve content structure and patch-wise style loss $\mathcal{L}_{\text{psd}}$ between style instruction and visual pattern of transferred results. Both content matching loss $\mathcal{L}_{\text{cm}}$ and style matching loss $\mathcal{L}_{\text{sm}}$ enhance the matching with the inputs. Simultaneously, we update D by maximizing discriminator loss $\mathcal{L}_D$ to distinguish between true patches $\mathcal{P}_{\mathcal{S}}$ or false patches $\mathcal{P}_{\hat{\mathcal{O}}}$, concerning style instructions. During CR, contrastive pairs of content images and style instructions are randomly sampled, and the transferred results are across-produced. We further update it by minimizing contrastive reasoning loss $\mathcal{L}_{\text{ctr}}$ to allow considering content consistency and mutual style relativeness. The overall optimization of CLVA is summarized as:

$$\begin{aligned} \mathcal{L}_G =& \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{psd}} + \mathcal{L}_{\text{cm}} + \mathcal{L}_{\text{sm}}, \\ \min_{G,\phi} \max_{D} \; & \mathcal{L}_G + \mathcal{L}_D + \mathcal{L}_{\text{ctr}}. \end{aligned} \tag{3.9}$$

3.4 Experiments

3.4.1 Experimental Setup

Datasets We consider DTD^2 [15] and ArtEmis [16] as reference style instructions. DTD^2 contains 5K texture images with its natural descriptions for visual attributes such as colors and texture patterns. ArtEmis provides 80K artworks from WikiArt [1] with annotations of visual contents and emotional effects as human-style feelings. We also collect 15K wallpapers from WallpapersCraft[2], which presents diverse scenes as content images (resized into 256x192).

[1]WikiArt: `https://www.wikiart.org`
[2]WallpapersCraft: `https://wallpaperscraft.com`

Evaluation Metrics To support large-scale evaluation, we treat transferred results directly from style images as semi-ground truth (Semi-GT) by the SOTA style transfer AdaAttn [78]. We apply the following metrics:

- SSIM [79] compares images in the luminance, contrast, and structure aspects. A higher SSIM has a higher structural similarity;
- Percept [80] computes from the gram matrix of visual features. A lower Percept loss shows that two images share a similar style pattern;
- FAD [81] is computed by the mean L2 distance of the activations from the InceptionV3 [53] features. As a distance metric, a lower FAD indicates that `LDAST` results and Semi-GT are more relevant;
- VLS [82] calculates the cosine similarity (from CLIP [66]) between style instructions and `LDAST` results.

We consider SSIM and FAD to compare with Semi-GT and calculate Percept loss directly with reference style images. Since each metric has different deficiencies, we also conduct a comprehensive human evaluation (75 samples for each method) from aspects of content, instruction, and style matching.

Baselines We conduct baselines for `LDAST` from various aspects:

- Style Transfer: We consider previous artistic style transfer methods SANet [60] and LST [61] that support arbitrary contetn images. We use the same style (instruction and image) encoding from our CLVA as style features and follow their own training process to perform `LDAST` upon them;
- Language-based Image Editing: We adopt ManiGAN [83] with affine combination module (ACM) as the general language-based editing baseline, where it modifies the content image by the style instruction. We treat normal style transferred results as ground truth for ManiGAN to learn from;

Method	Automatic Metrics				Human Evaluation			
	SSIM↑	Percept↓	FAD↓	VLS↑	Content↑	Instruction↑	Style↑	Semi-GT↑
SANet [60]	35.50	0.2129	0.1627	23.57	2.701	2.477	2.738	2.630
LST [61]	34.84	0.2129	0.1533	23.16	2.743	2.831	2.651	2.528
ManiGAN [83]	32.70	0.2401	0.1663	23.25	2.757	2.562	2.937	2.922
CLIPstyler [72]	25.24	0.2598	0.1818	**24.62**	2.948	3.388	3.073	3.265
CLVA	**36.65**	**0.2033**	**0.1493**	24.00	**3.852**	**3.742**	**3.603**	**3.655**

Table 3.1: The testing results of `LDAST` with visual attribute instructions on DTD^2.

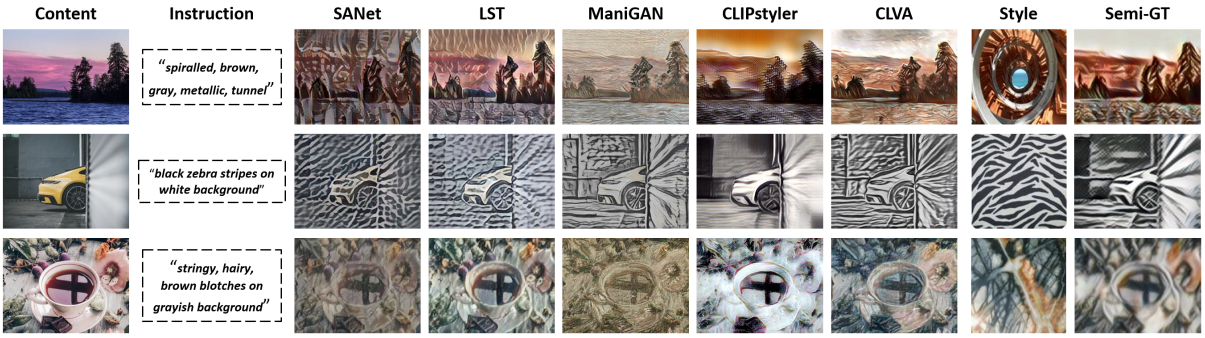


Figure 3.3: The qualitative comparison with visual attribute instructions on DTD^2.

- CLIP-based Optimization: CLIPstyler [72] manipulates the content image based on the CLIP alignment of the guided instruction, which can carry out arbitrary content images for `LDAST`.

Implementation Detail We adopt VGG-19 [84] as our visual encoder G_{E} and visual decoder G_{D}. Text encoder ϕ first adopts RoBERTa [85] for a general linguistic and then expands its spatial dimension to jointly embed with style features. We follow SANet [60] to fuse between content and style features in G_{D}. The patch-wise style discriminator D contains a similar architecture with a dense layer to determine the correlation between instructions and image patches. Both G_{E} and G_{D} are initialized from SANet and further updated during the CLVA training process. We adopt Adam [55] to optimize CLVA with the learning rate 3e-4 for $\mathcal{L}_G$, 1e-4 for $\mathcal{L}_D$, and 3e-5 for $\mathcal{L}_{\mathrm{ctr}}$.

3.4.2 Main Results

Instruction with Visual Attributes Table 3.1 illustrates the comparison of `LDAST` with baselines on DTD[2]. As regards automatic metrics, CLVA preserves content structures (the highest 36.6 SSIM) and stylizes with related visual attributes to style images (the lowest 0.203 Percept loss). Furthermore, CLVA brings out the highest overall similarity as Semi-GT (the lowest 0.149 FAD). Since CLIPstyler directly optimizes by CLIP [66], it makes the highest VLS. Through the patch-wise discriminator, our CLVA can still produce style patterns correlated to given instructions even without the pre-trained CLIP.

The human evaluation investigates the matching between transferred results with content images (Content), style instructions (Instruction), style images (Style), and Semi-GT (Semi-GT). The results are calculated by the mean ranking score (from 1 to 5, the higher is better) of each method. In general, our CLVA has an apparent advantage in preserving content structures (the highest 3.85 Content) and presenting aligned style patterns (the highest 3.74 Instruction). Despite the aid of CLIP, CLIPstyler is still behind CLVA, with an even higher gap in style image matching.

From the aspect of the qualitative comparison in Fig. 3.3, previous SANet and LST only produce repetitive and disordered textures in their transferred results. ManiGAN modifies the style directly over pixels, suffering from blurring objects. CLIPstyler is sometimes misguided by CLIP, making irrelevant patterns, such as the bright white background in the third case. In contrast, CLVA extracts a more detailed style from different kinds of guidance (*e.g.*, "*brown metallic*" and "*stringy hairy*"), leading to superior `LDAST` results that correspond to style instructions.

Instruction with Emotional Effects Unlike visual attributes, emotional effect instructions are more challenging in connecting the visual semantics of described objects or style patterns from human feelings. We consider this on ArtEmis [16], where the model

Method	Automatic Metrics				Human Evaluation			
	SSIM↑	Percept↓	FAD↓	VLS↑	Content↑	Instruction↑	Style↑	Semi-GT↑
SANet [60]	38.36	**0.0352**	0.1548	19.30	3.170	2.978	2.980	2.890
LST [61]	**42.13**	0.0386	0.1595	19.92	2.967	2.714	2.614	2.757
ManiGAN [83]	38.46	0.0500	0.1554	19.69	2.729	2.583	2.879	3.192
CLIPstyler [72]	24.17	0.0659	0.1759	**21.04**	2.777	3.140	2.998	2.952
CLVA	40.32	0.0357	**0.1418**	20.11	**3.357**	**3.586**	**3.530**	**3.208**

Table 3.2: The testing results of LDAST with emotional effect instructions on ArtEmis.

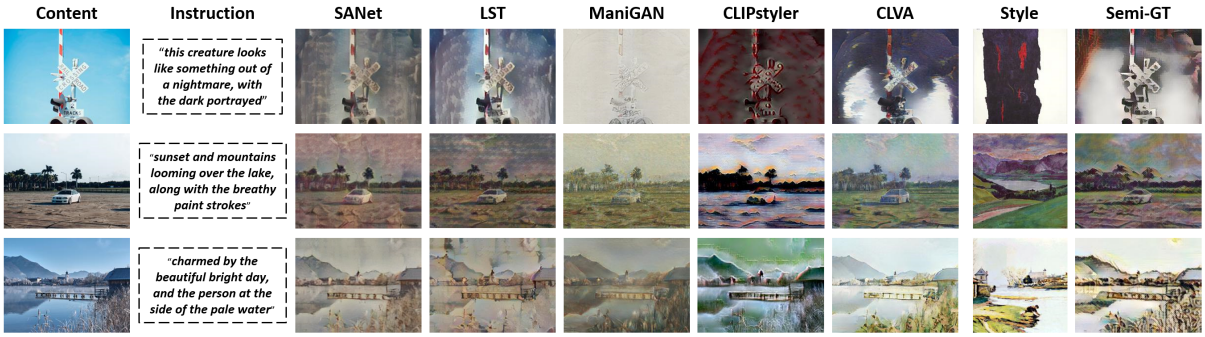


Figure 3.4: The qualitative comparison with emotional effect instructions on ArtEmis.

has to express the latent visual concepts of emotional effect instructions. CLVA performs with more balance (both the second-highest SSIM and the second-lowest Percept) from Table 3.2, especially the lowest 0.141 FAD, making the most similar transferred results to Semi-GT. From human aspects, CLVA can preserve more concrete contents and present more correlated style patterns than CLIPstyler.

The qualitative comparison in Fig. 3.4 illustrates that previous SANet [60] and LST [61] contain unsmooth and fragmentary patterns with blurring contents. Without a style transformation process, ManiGAN [83] modifies with only monotonous colors. CLIPstyler fails to capture human-style feelings well, suffering from weird and unpleasant results. Different from them, our CLVA leads to a more colorful and corresponding stylization as human emotion (*e.g.*, reveals the latent yet correlated "*grassland*" from "*side of the water*").

	Ablation Settings				Automatic Metrics			
	$\mathcal{L}_{\text{rec}}$+$\mathcal{L}_{\text{psd}}$	$\mathcal{L}_{\text{cm}}$	$\mathcal{L}_{\text{sm}}$	$\mathcal{L}_{\text{ctr}}$	SSIM↑	Percept↓	FAD↓	VLS↑
(a)	✓	✗	✗	✗	34.73	0.2290	0.1568	23.29
(b)	✓	✓	✗	✗	<u>36.05</u>	0.2304	0.1512	23.27
(c)	✓	✗	✓	✗	35.73	<u>0.2049</u>	<u>0.1508</u>	<u>23.69</u>
(d)	✓	✓	✓	✗	35.86	0.2100	0.1499	23.54
(e)	✓	✓	✓	✓	**36.65**	**0.2033**	**0.1493**	**24.00**

Table 3.3: The ablation study of CLVA with visual attribute instructions on DTD^2.

	DTD^2		ArtEmis	
Method	R@1	R@5	R@1	R@5
CLIP [66]	13.9	30.7	9.8	20.7
CLVA	**19.3**	**45.1**	**13.9**	**30.7**

Table 3.4: The instruction-to-style retrieval results.

	Human Evaluation			
Method	Content↑	Instruction↑	Style↑	Semi-GT↑
CLIPstyler*	1.208	1.347	1.292	1.333
CLVA	**1.792**	**1.653**	**1.708**	**1.667**

Table 3.5: The human comparison with fine-tuned CLIPstyler on DTD^2.

3.4.3 Ablation Study

We conduct an ablation study of each component effect on DTD^2 in Table 3.3. At row (a), with the reconstruction $\mathcal{L}_{\text{rec}}$ and the patch-wise style $\mathcal{L}_{\text{psd}}$, CLVA achieves feasible `LDAST` results by concrete structures and extracted style semantics. Row (b)-(d) shows the strength of content matching $\mathcal{L}_{\text{cm}}$ and style matching $\mathcal{L}_{\text{sm}}$. If considered altogether, it can benefit and strike a balance between both. Finally, contrastive reasoning $\mathcal{L}_{\text{ctr}}$ enables CLVA to consider contrastive pairs, making a comprehensive improvement at row (e).

Why CLVA is better than CLIP-based? Despite no CLIP optimized, CLVA demonstrates superior results on `LDAST` with all aspects of automatic metrics and human evaluation. To investigate it, we conduct instruction-to-style retrieval based on the similarity between features of style instructions and style images. Table 3.4 shows that our learned CLVA performs higher Recall@k on both DTD^2 and ArtEmis, leading to a better

Method	Time (sec)			GPU (MB)		
	BS=1	32	50	BS=1	32	50
ManiGAN [83]	0.079	0.533	1.148	3312	6572	8129
CLIPstyler [72]	99.98	*	*	5429	*	*
CLVA	**0.029**	**0.246**	**0.405**	**1525**	**3207**	**4441**

Table 3.6: The time and GPU cost when performing `LDAST` on TITAN X. * means that this method can only run one input at a single time.

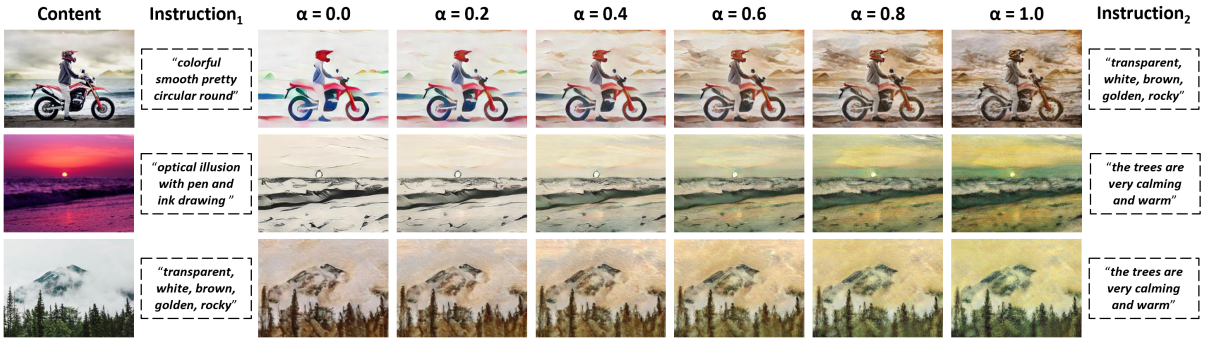


Figure 3.5: The style interpolation results over instructions.

instruction-style alignment than the used CLIP. From Table 3.5, even though the CLIP in CLIPstyler has been fine-tuned ahead, our CLVA still produces preferable `LDAST` results from all human aspects of content, instruction, and style matching.

Apart from transfer quality, CLVA also holds a higher efficiency than CLIP-based methods. Table 3.6 illustrates the time and GPU cost on a single TITAN X (12GB) with the content image size 256x192. CLIStyler takes more than 30 seconds for only one input pair. Instead of numerous iterations to align with CLIP, we carry out `LDAST` in one shot, taking merely 0.03 seconds. Without updating the model during inference, our CLVA supports parallelization and can accomplish 50 pairs in half a second. Besides, as a lightweight style transfer network, CLVA requires the least GPU memory for `LDAST`.

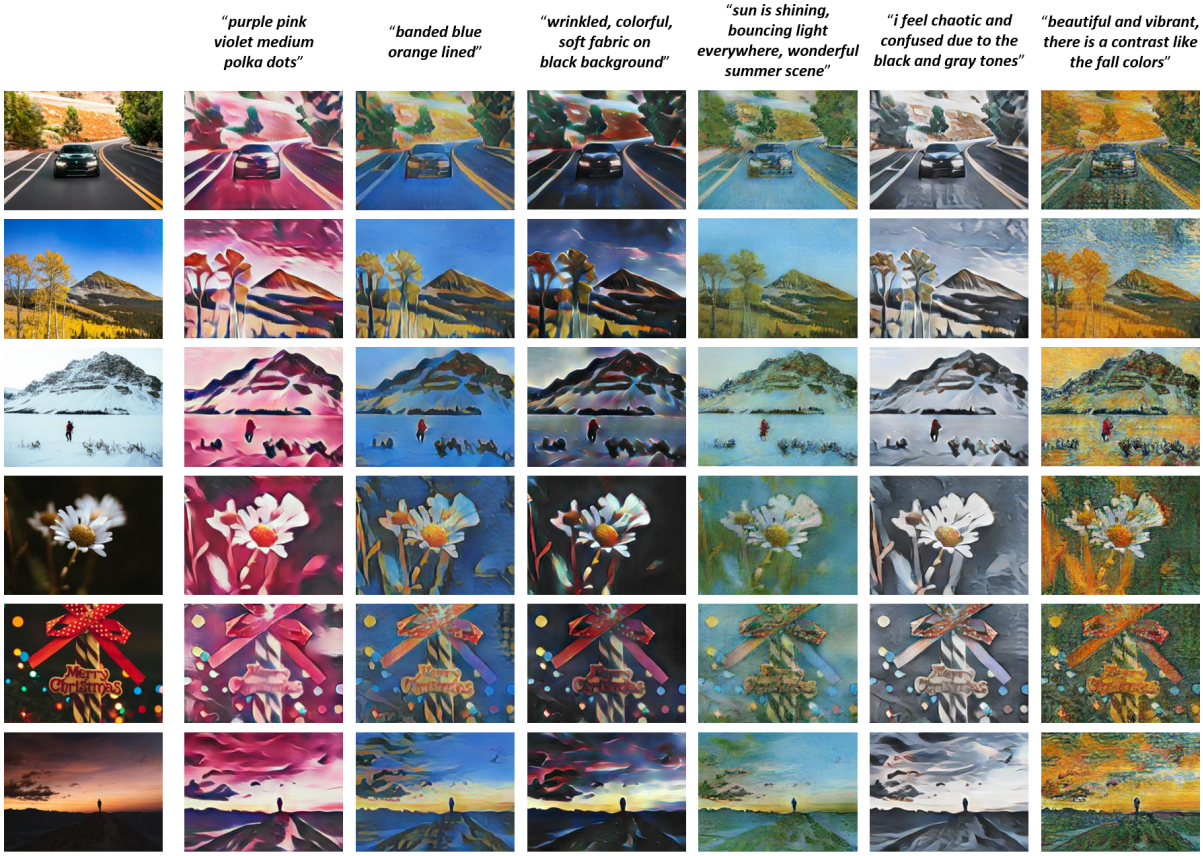


Figure 3.6: The qualitative examples on diverse content images and style instructions.

3.4.4 Qualitative Results

As illustrated in Fig. 3.5, we investigate the linear interpolation of extracted style patterns by CLVA. Considering style features $h^{\mathcal{S}}_{\mathcal{X}_1}$ and $h^{\mathcal{S}}_{\mathcal{X}_2}$ of instructions $\mathcal{X}_1$ and $\mathcal{X}_2$, the interpolated $h^{\mathcal{S}}_{\mathrm{p}}$ should be:

$$h^{\mathcal{S}}_{\mathrm{p}} = (1-\alpha)h^{\mathcal{S}}_{\mathcal{X}_1} + \alpha h^{\mathcal{S}}_{\mathcal{X}_2}, \tag{3.10}$$

where α is the style ratio between the two. By training on DTD[2] and ArtEmis altogether, CLVA even performs interpolated stylization by both visual attribute and emotional effect instructions in the third row. Fig. 3.6 demonstrates diverse `LDAST` pairs by our CLVA.

3.5 Summary

In this chapter, we introduce the language-driven artistic style transfer task to investigate text-guided style transfer. We present contrastive language visual artist, which adopts patch-wise style discrimination and contrastive reasoning to jointly learn between style images and style instructions. We showcase that our framework can express various style patterns on input contents, including visual attributes and emotional effects. Apart from effectiveness, we achieve higher time/memory efficiency, leading to practical usage.

Part II

Controllable Video Editing via Natural Language

Chapter 4

Text-guided Video Editing

4.1 Introduction

Video editing tools are widely used nowadays for digital design. Although the demand for these tools is high, the prior knowledge required makes it difficult for novices to get started. In this chapter, we introduce language-based video editing (`LBVE`), a general video-to-video (V2V) task, where the target video is controllable directly by language instruction. `LBVE` treats a video and an instruction as the input, and the target video is edited from the textual description. As illustrated in Fig. 4.1, the same person performs different hand gestures guided by the instruction. Different from text-to-video (T2V) [86, 87, 2], video editing enjoys two following features: 1) the scenario (*e.g.*, scene or human) of the source video is preserved instead of generating all content from scratch; 2) the semantic (*e.g.*, property of the object or its moving action) is presented differently in the target video.

To tackle the `LBVE` task, we propose multimodal multi-level transformer (M^3L) to perform video editing conditioning on the guided text. In M^3L, the encoder models the moving motion to understand the entire video, and the decoder serves as a global planner

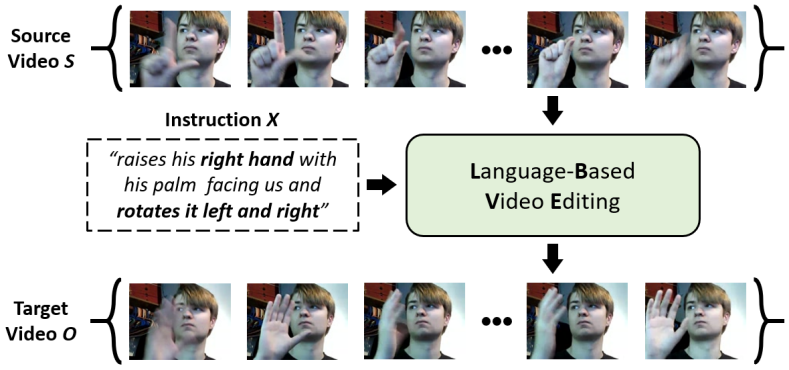


Figure 4.1: The language-based video editing (LBVE) task, which requires editing a source video into the target video guided by the instruction.

to generate each frame of the target video. For better video perception to link with the given instruction, the incorporated multi-level fusion fuses between these two modalities. Specifically, the local-level fusion is applied with the text tokens for fine-grained visual understanding, and the global-level fusion extracts the key features of the moving motion.

For evaluation, we collect three datasets for LBVE. E-MNIST and E-CLEVR are built from hand-written number recognition MNIST [88] and compositional VQA CLEVR [19], which are prepared for content replacement and semantic manipulation. E-JESTER is built upon the same person performing different hand gestures with human instruction for natural video evaluation.

Dataset	Edit Aspect	Instruction Example
E-MNIST	Object/Motion	*change the direction from upper right to lower left and the number 1 to 3*
E-CLEVR	Property/Position	*move to the right behind and change the red cube into the small purple*
E-JESTER	Hand Gesture	*rotate and swipe her right hand from left to right*

4.2 Related Work

Language-based Video Generation Generative video modeling [89, 90, 91, 92, 93] is a widely-discussed research topic that looks into the capability of a model to generate a video purely in pixel space. Built upon video generation, text-to-video (T2V) [86, 87, 2] synthesizes a video by the guided text description, which makes the video output controllable by the natural language. Different from generating video from scratch, we investigate the video editing task, which replaces the specific object with different properties or changes the moving motion in the input video.

Video-to-video Synthesis Video super-resolution [94, 95], segmentation video reconstruction [96, 97], video style transfer [98, 99], and video inpainting [100, 101] can be considered as the particular case of video-to-video synthesis (V2V). Among them, video prediction [102, 103, 104], which predicts future frames conditioning on the given video, is one of the most related to our `LBVE` task. While, for video prediction, there are many possibilities of future events, which makes it not deterministic for real-world usage [87]. In contrast, with the guided text description, `LBVE` can perform V2V with content editing and lead to a predictable target video.

4.3 Multimodal Multi-Level Transformer ($\mathbf{M^3L}$)

4.3.1 Overview

Task Definition We study the `LBVE` task to edit a source video $\mathcal{S}$ into a target video $\mathcal{O}$ by a given instruction $\mathcal{X}$. Specifically, the source video $\mathcal{S}$ contains N frames as $\{s_1, s_2, ..., s_N\}$, and the instruction $\mathcal{X} = \{w_1, w_2, ..., w_L\}$ where L is the number of word token in $\mathcal{X}$. The target video $\mathcal{O}$ also includes N frames as $\{o_1, o_2, ..., o_N\}$. For `LBVE`, the

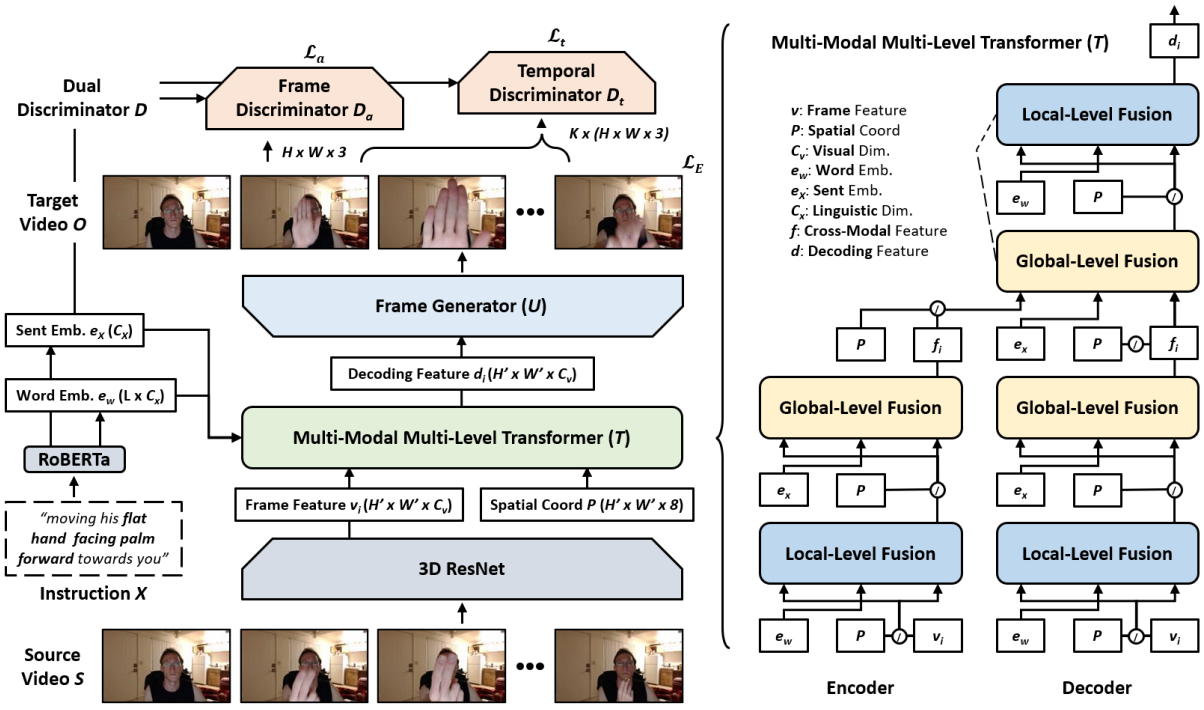


Figure 4.2: The overview of our multimodal multi-level transformer (M^{3}L). M^{3}L contains the transformer to encode the source video and decode for the target video frame by the multi-level fusion (MLF).

model should preserve the scenario from $\mathcal{S}$ but change the related semantics in $\mathcal{O}$ guided by $\mathcal{X}$. Note that the editing process is at a pixel level where the model has to generate each pixel of each frame and then assemble them as the target video.

The proposed multimodal multi-level transformer (M^{3}L) is illustrated in Fig. 4.2. M^{3}L first extracts the frame features for each frame in the source video, the sentence embedding, and each word embedding for the instruction. Then, the transformer is proposed to model the sequential information of the source and the target video as the decoding features. Finally, the generator generates the frame in the target video.

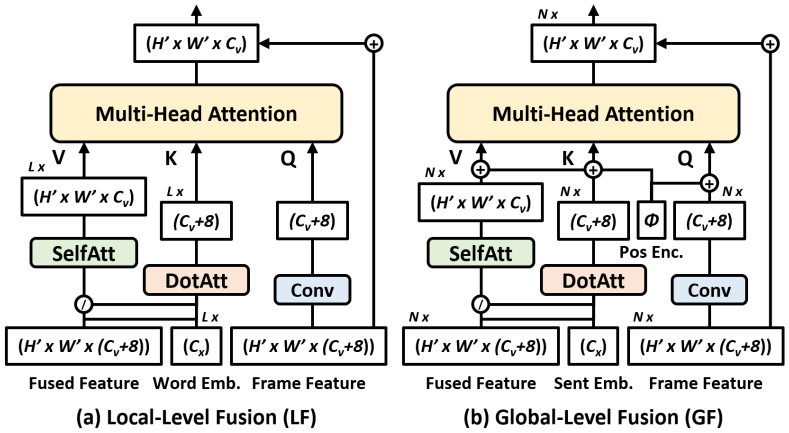


Figure 4.3: The computing flow of multi-level fusion (MLF), including the local-level fusion (LF) and the global-level fusion (GF).

4.3.2 Multimodal Encoder-Decoder

Frame and Linguistic Features Extraction We first apply 3D-ResNet and RoBERTa [85] to extract the frame features v and linguistic features $\{e_{\mathcal{X}}, e_w\}$ for the two modalities:

$$\begin{aligned} \{v_1, v_2, ..., v_N\} &= \text{3D-ResNet}(\{s_1, s_2, ..., s_N\}), \\ e_{\mathcal{X}}, \{e_{w_1}, e_{w_2}, ..., e_{w_L}\} &= \text{RoBERTa}(\mathcal{X}), \end{aligned} \tag{4.1}$$

where e_{w_i} is the word embedding of each token w_i, $e_{\mathcal{X}}$ is the entire sentence embedding, and L represents the length of the instruction $\mathcal{X}$.

Multi-Level Fusion Both video and language are multi-level conveyed, where video is composed of a series of image frames and language is a set of word tokens with a specific order. The multi-level fusion (MLF), as illustrated in Fig. 4.3, consists of the local-level fusion (LF) to fuse between a single frame and each word token, and the global-level fusion (GF) models the entire video sequence with the whole instruction. Both LF and GF are computed with the multi-head attention (MHA) [105]. MHA acquires the weighted sum of the value features (V) by considering the correlation between the query

features (Q) and the key features (K):

$$\mathrm{MHA}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{softmax}(\frac{\mathrm{Q}\cdot\mathrm{K}^T}{\sqrt{C_\mathrm{K}}})\mathrm{V}. \tag{4.2}$$

LF investigates which portion should be focused by each word e_w in a single frame v_i. We provide the relative spatial information by concatenating an 8-D spatial coordinate features P [106] with v_i as p^L. To fuse between vision and language, we apply the self-attention mechanism (SelfAtt) [54, 107] upon the concatenated features q^L to capture the correlation between word expression and visual context into s^L. We adopt a 1-layer convolutional net (Conv) to extract the context-only visual features c^L along the channel of v_i; and the widely-used dot-product attention (DotAtt) [50, 108] for the word-focused visual features d^L_l with each word e_{w_l}. We treat the context-only visual features c^L as K, the word-focused visual features d^L_l as Q, and the cross-modal features s^L as V in LF:

$$\mathrm{LF}(v_i^\mathrm{L}) = v_i^\mathrm{L} \oplus \mathrm{MHA}(\mathrm{c}^\mathrm{L},\mathrm{d}^\mathrm{L},\mathrm{s}^\mathrm{L}), \tag{4.3}$$

where

$$\begin{aligned}
&\mathrm{p}^\mathrm{L} = [v_i^\mathrm{L}, P], \quad \mathrm{q}^\mathrm{L} = \{[v_i^\mathrm{L}, P, e_{w_1}], ..., [v_i^\mathrm{L}, P, e_{w_L}]\},\\
&\mathrm{c}^\mathrm{L} = \mathrm{Conv}^\mathrm{L}(\mathrm{p}^\mathrm{L}),\\
&\mathrm{d}^\mathrm{L}_l = \mathrm{DotAtt}(\mathrm{p}^\mathrm{L}, e_{w_l}) = \sum_{(h,w)} \mathrm{softmax}(\mathrm{p}^\mathrm{L}\cdot W^L_\mathrm{d}\cdot e^T_{w_l})_{(h,w)}\cdot \mathrm{p}^\mathrm{L}_{(h,w)},\\
&\mathrm{s}^\mathrm{L}_l = \mathrm{SelfAtt}(\mathrm{q}^\mathrm{L}_l), \mathrm{s}^\mathrm{L}_{l(h,w)} = \sum_{(x,y)} \mathrm{softmax}(\mathrm{q}^\mathrm{L}_l \cdot {\mathrm{q}^\mathrm{L}_{l(h,w)}}^T)_{(x,y)}\cdot \mathrm{q}^\mathrm{L}_{l(x,y)},
\end{aligned} \tag{4.4}$$

and W^L_d is the learnable attention matrix between p^L and e_w. In this way, our LF fuses between visual context and word expression from SelfAtt and takes the important portion of each token from DotAtt into consideration.

GF views the entire frame sequence $\{v_1, ..., v_N\}$ with the whole instruction $e_{\mathcal{X}}$ to extract the global motion of the video. Similar to LF, we acquire the fused cross-modal features $\mathrm{s}_n^{\mathrm{G}}$ from SelfAtt, the context-only visual features $\mathrm{c}_n^{\mathrm{G}}$ from $\mathrm{Conv}^{\mathrm{G}}$, and the sentence-focused visual features $\mathrm{d}_n^{\mathrm{G}}$ from DotAtt for v_n^{G}. We follow [105], where the video-level features of v_i can be represented as the relative weighted-sum over all frame-level v, and add on the positional encoding ϕ to incorporate the sequential order. We treat $\{\mathrm{s}_n^{\mathrm{G}}\}$ as V, $\{\mathrm{c}_n^{\mathrm{G}}\}$ as Q, and $\{\mathrm{d}_n^{\mathrm{G}}\}$ as K for the correlation between a frame pair:

$$\mathrm{GF}(v^{\mathrm{G}}) = v^{\mathrm{G}} \oplus \mathrm{MHA}(\mathrm{c}^{\mathrm{G}} \oplus \phi, \mathrm{d}^{\mathrm{G}} \oplus \phi, \mathrm{s}^{\mathrm{G}} \oplus \phi), \tag{4.5}$$

where

$$\begin{aligned} &\mathrm{p}^{\mathrm{G}} = \{[v_1^{\mathrm{G}}, P], ..., [v_N^{\mathrm{G}}, P]\}, \quad \mathrm{q}^{\mathrm{L}} = \{[v_1^{\mathrm{G}}, P, e_{\mathcal{X}}], ..., [v_N^{\mathrm{G}}, P, e_{\mathcal{X}}]\}, \\ &\mathrm{c}_n^{\mathrm{G}} = \mathrm{Conv}^{\mathrm{G}}(\mathrm{p}^{\mathrm{G}})_n, \ \mathrm{d}_n^{\mathrm{G}} = \mathrm{DotAtt}(\mathrm{p}_n^{\mathrm{G}}, e_{\mathcal{X}}), \ \mathrm{s}_n^{\mathrm{G}} = \mathrm{SelfAtt}(\mathrm{q}_n^{\mathrm{G}}). \end{aligned} \tag{4.6}$$

GF models the video sequence as fused cross-modal features from SelfAtt.

Encoder and Decoder The encoder (Enc) first adopts the local-level fusion (LF) to extract important portions from each single frame v^s with each word embedding e_w; then the global-level fusion (GF) extracts the entire video motion with the sentence embedding $e_{\mathcal{X}}$ as the cross-modal features f_i^s:

$$f_i^s = \mathrm{GF}(\mathrm{LF}(v^s, e_w), e_{\mathcal{X}})_i. \tag{4.7}$$

During decoding, the decoder (Dec) also extracts the cross-modal features f_i^o in the same way from the previously generated frames $\{o_1, ..., o_{i-1}\}$. To acquire the decoding features d_i, GF is first adopted to give the high-level concept of moving motion by the

interaction between the cross-modal features. LF is applied for the detailed specific property provided from word tokens e_w:

$$f_i^o = \mathrm{LF}(\mathrm{GF}(\{v_1^o, ..., v_{i-1}^o\}, e_{\mathcal{X}} | f^s)_i, e_w). \tag{4.8}$$

In summary, the transformer T models the source video frame v^s and the given instruction $\{e_{\mathcal{X}}, e_w\}$, and considers previous generated target frames $\{o_1, ..., o_{i-1}\}$ to acquire the decoding features d_i:

$$d_i = T(\{o_1, ..., o_{i-1}\} \mid v^s, \{e_{\mathcal{X}}, e_w\}). \tag{4.9}$$

4.3.3 Video Frame Generation

With the decoding features d_i from T, we adopt ResBlocks [49] into the generator U to scale up d_i and synthesize into $\hat{o}_i$:

$$\hat{o}_i = U(d_i), \quad \hat{\mathcal{O}} = \{\hat{o}_1, \hat{o}_2, ..., \hat{o}_N\}. \tag{4.10}$$

We calculate the editing loss $\mathcal{L}_E$ by mean pixel difference using mean-square loss over each frame between $\mathcal{O}$ and $\hat{\mathcal{O}}$:

$$\mathcal{L}_E = \frac{1}{N} \sum_{i=1}^{N} \mathrm{MSELoss}(o_i, \hat{o}_i). \tag{4.11}$$

Dual Discriminator Apart from visual difference, we also consider the video quality of our generated $\hat{\mathcal{O}}$. We apply the dual discriminator D [93], where the frame discriminator D_a improves the single frame quality and the temporal discriminator D_t constrains the temporal consistency for a smooth output video $\hat{\mathcal{O}}$. We treat D_a as a binary classifier,

which discriminates that a target video frame o is from ground-truth $\mathcal{O}$ or our synthesized $\hat{\mathcal{O}}$. Simultaneously, D_t judges that if K consecutive frames are smooth and consistent enough to be a real video fragment as the binary discrimination. The video quality loss $\mathcal{L}_G$ is computed for both frame quality and temporal consistency:

$$\begin{aligned}
\mathcal{L}_{\hat{a}} &= \frac{1}{N}\sum_{i=1}^{N}\log(1 - D_a(\hat{o}_i)), \\
\mathcal{L}_{\hat{t}} &= \frac{1}{M}\sum_{i=1}^{M}\log(1 - D_t(\{\hat{o}_i, ..., \hat{o}_{i+K-1}\})), \\
\mathcal{L}_G &= \mathcal{L}_{\hat{a}} + \mathcal{L}_{\hat{t}},
\end{aligned} \tag{4.12}$$

where $M = N - K + 1$. On the other hand, the dual discriminator D is training to distinguish between $\mathcal{O}$ and $\hat{\mathcal{O}}$:

$$\begin{aligned}
\mathcal{L}_a &= \frac{1}{N}\sum_{i=1}^{N}(\log(1 - D_a(\hat{o}_i)) + \log(D_a(o_i))), \\
\mathcal{L}_t &= \frac{1}{M}\sum_{i=1}^{M}(\log(1 - D_t(\{\hat{o}_i, ..., \hat{o}_{i+K-1}\})) \\
&\qquad + \log(D_t(\{o_i, ..., o_{i+K-1}\}))), \\
\mathcal{L}_D &= \mathcal{L}_a + \mathcal{L}_t.
\end{aligned} \tag{4.13}$$

4.3.4 Learning of M^{3}L

Algo. 3 presents the learning process of M^{3}L. Since `LBVE` is also a sequential generation process, we apply the widely used teacher-forcing training trick, where we feed in the ground-truth target frame o_{i-1} instead of the predicted $\hat{o}_{i-1}$ from the previous timestamp to make the training more robust. We adopt the transformer T to model the source video and input instruction, and the frame generator U to generate the target video

Algorithm 3 Multimodal Multi-level Transformer (M^{3}L)

T, U: Transformer / Frame Generator
D: Dual Discriminator (D_a and D_t)

while TRAIN_M^{3}L **do**
    $\mathcal{S}$, $\mathcal{X}$, $\mathcal{O} \leftarrow$ sampled source / instruction / target
    $\{v_1, ..., v_N\} =$ 3D-ResNet($\mathcal{S}$)
    $e_{\mathcal{X}}$, $\{e_{w_1}, ..., e_{w_N}\} =$ RoBERTa($\mathcal{X}$)
    for $i \leftarrow 1$ to N **do** ▷ teacher-forcing training
        $d_i = T(\{o_1, ..., o_{i-1}\} \mid v, \{e_{\mathcal{X}}, e_w\})$ ▷ Eq. 4.9
        $\hat{o}_i = U(d_i)$

        $\mathcal{L}_E$, $\mathcal{L}_G \leftarrow$ visual difference / video quality loss ▷ Eq. 4.11 and 4.12
        Update T and U by minimizing $\mathcal{L}_G$+$\mathcal{L}_E$
        $\mathcal{L}_D \leftarrow$ discrimination loss ▷ Eq. 4.13
        Update D by maximizing $\mathcal{L}_D$
    end for
end while

frame. During training, we minimize the video quality loss $\mathcal{L}_G$ with the visual difference $\mathcal{L}_E$ to optimize M^{3}L. We also update the dual discriminator D, including the frame discriminator D_a and the temporal discriminator D_t, by maximizing $\mathcal{L}_D$. The overall optimization object can be summarized as:

$$\min_{T,U} \max_{D} \mathcal{L}_G + \mathcal{L}_E + \mathcal{L}_D. \tag{4.14}$$

4.4 Experiments

4.4.1 Experimental Setup

Datasets We build three new datasets specially designed for `LBVE`, including two diagnostic datasets (E-MNIST and E-CLEVR) and one human gesture dataset (E-JESTER):

- E-MNIST: Extended from Moving MNIST [88, 18], hand-written numbers are moving along a specific direction and will reverse their direction if bumping into a boundary. The instructions include two kinds of editing actions: *content replacing*

is to replace the specific number with the given one, and *semantic manipulation* changes the starting direction for different moving motions. There are 12K training pairs of source-target video.

- E-CLEVR: Following CATER [109], we create each frame and combine them as the video in our E-CLEVR upon the original CLEVR dataset [19]. The editing action includes changing the property of the specific object and placing the moving object into a particular final position. E-CLEVR contains plentiful object properties (*e.g.*, color, shape, and size) and different relative positions of the final target. We generate 11K examples for E-CLEVR.
- E-JESTER: We prepare pairs of clips by the same person as the source-target videos and collect the human-labeled instruction from 20BN-JESTER [20]. Each actor performs different kinds of gesture moving in front of the camera, which brings out 27 classes in total. We have the natural video whose scenario is preserved, but the semantics are changing with natural guided text, leading to 15K pairs.

Evaluation Metrics

- VAD: We apply 3D-CNN (ResNeXt [110] and I3D [111]) to compute the video activation distance (VAD) [81] as the mean L2 distance between video features. A lower VAD means that videos are more related to each other.
- OA: We consider the object accuracy (OA) for E-MNIST and E-CLEVR, calculated by the correctness of the presented objects in the target video from a pre-trained object detector.
- mIoU: We also evaluate E-MNIST and E-CLEVR via mean intersection over union (mIoU) over the positions of detected objects between generated and ground-truth results, which is averaged from each video frame.
- GA: We report the gesture accuracy (GA) for E-JESTER, which is calculated as the

	E-MNIST			E-CLEVR			E-JESTER	
	VAD↓	OA↑	mIoU↑	VAD↓	OA↑	mIoU↑	VAD↓	GA↑
pix2pix [112]	3.05	87.7	64.1	2.84	80.4	60.5	2.00	8.6
vid2vid [96]	2.30	87.5	77.9	2.21	80.5	69.3	1.62	82.0
E3D-LSTM [113]	2.10	90.4	81.3	2.11	83.1	72.2	1.55	83.6
M^{3}L	**1.90**	**93.2**	**84.7**	**1.96**	**84.5**	**78.4**	**1.44**	**89.3**

Table 4.1: The testing results of `LBVE` on E-MNIST, E-CLEVR, and E-JESTER.

gesture classification accuracy of the edited video by MFF[1]. A higher GA represents that it can follow the guided text and generate the corresponding type of gesture.

Baselines We consider the following methods, by concatenating the linguistic features, to carry out `LBVE` as the compared baselines:

- pix2pix [112] processes the source video frame-by-frame and assembles all of them as video synthesis.
- vid2vid [96] applies the temporal discriminator to consider several previous frames for video translation.
- E3D-LSTM [113] incorporates 3D CNN into LSTM for video prediction. We treat the input as the given video and predict the remaining part as the target video.

Implementation Detail We apply 3-layer ResBlocks [49] into the 3D-ResNet and the generator U with kernel size 3 and stride 1 in the first layer. In particular, we incorporate 1-layer self-attention for better frame generation into U following SAGAN [54]. The visual features dimension is 256 and the language features dimension is 1024 from RoBERTa [85]. Adam [55] is adopted to optimize our M^{3}L with the learning rate 3e-4 for $\mathcal{L}_E$, 1e-4 for $\mathcal{L}_G$ and $\mathcal{L}_D$.

[1]MFF: `https://github.com/okankop/MFF-pytorch`

Instruction	MLF	VAD↓	GA↑
✗	✗	1.99	4.7
✓	✗	1.50	85.4
✓	✓	**1.44**	**89.3**

Figure 4.4: The ablation study of M^{3}L on E-JESTER.

	E-MNIST			**E-CLEVR**		
MLF	VAD↓	OA↑	mIoU↑	VAD↓	OA↑	mIoU↑
✗	2.64	82.6	73.6	2.32	70.1	66.6
✓	**2.35**	**87.5**	**79.1**	**2.29**	**76.7**	**71.5**

Figure 4.5: The zero-shot results on E-MNIST and E-CLEVR.

4.4.2 Quantitative Results

Table 4.1 shows the overall testing results compared between the baselines and ours M^{3}L. pix2pix only adopts image-to-image translation, resulting in insufficient output video (*e.g.*, 64.1 mIoU on E-MNIST and 2.84 VAD on E-CLEVR). Even if vid2vid and E3D-LSTM consider temporal consistency, the lack of explicit cross-modal fusion still makes it difficult to perform `LBVE`. In contrast, our M^{3}L, which incorporates the multi-level fusion (MLF), achieves the best results across all metrics on all diagnostic datasets.

Similar trends can be found in the natural E-JESTER dataset. Although vid2vid and E3D-LSTM may have similar visual measurement scores to our approach, M^{3}L achieves the highest 89.3 GA. The significant improvement demonstrates that MLF benefits not only the visual quality but also the semantics of the predicted video.

4.4.3 Ablation Study

Table 4.4 presents the ablation study on E-JESTER. If without the given instruction, the model lacks the specific editing target and results in poor 1.99 VAD and 4.7 GA. The performance comprehensively improves when incorporating our proposed MLF. The multi-level modeling from MLF benefits not only the understanding between video and instruction but also leads to accurate frame generation.

	w/ MLF	w/o MLF	Tie
Video Quality	**67.1%**	27.1%	5.8%
Video-Instruction Alignment	**53.3%**	35.1%	11.6%
Similarity to GT Video	**59.6%**	28.9%	11.6%

Table 4.2: The human evaluation on E-JESTER.

Zero-shot Generalization In E-MNIST, there are 40 different object-semantic combinations[2]. We remove 10 of them in the training set (*e.g.*, number *1* with *upper left* or number *3* with *lower down*) and evaluate the complete testing set. For E-CLEVR, we filter out 12 kinds (*e.g.*, *small gray metal sphere* or *large purple rubber cube*) from the total 96 properties[3]. This testing scenario is widely used to evaluate new combinations of object-semantic pairs that are not seen during training. In Table 4.5, the model has a significant performance drop due to the lack of object properties or moving semantics. The MLF still improves the generalizability (*e.g.* mIoU from 66.6 to 71.5 on E-CLEVR), even if testing with zero-shot examples.

Inference Efficiency When using only the CPU, M^{3}L carries out E-JSTER (in 128x128) with about 11.9 FPS. With the acceleration from the GPU (TITAN X), it can further achieve 35.8 FPS, which is faster than the real-time requirement (24 FPS). These results show that our M^{3}L can perform the `LBVE` task for practical usage efficiently.

Human Evaluation Table 4.2 demonstrates the human evaluation between without and with MLF. We randomly sample 75 examples for each model and ask three following questions: (*i*) Which video has better quality; (*ii*) Which video corresponds more to the given instruction; (*iii*) Which video is more similar to the ground truth? Firstly, about 67% think that generated videos from MLF have better quality. Moreover, more than

[2]E-MNIST: 10 different numbers and 4 different directions
[3]E-CLEVR: 3 shapes, 8 colors, 2 materials, and 2 shapes

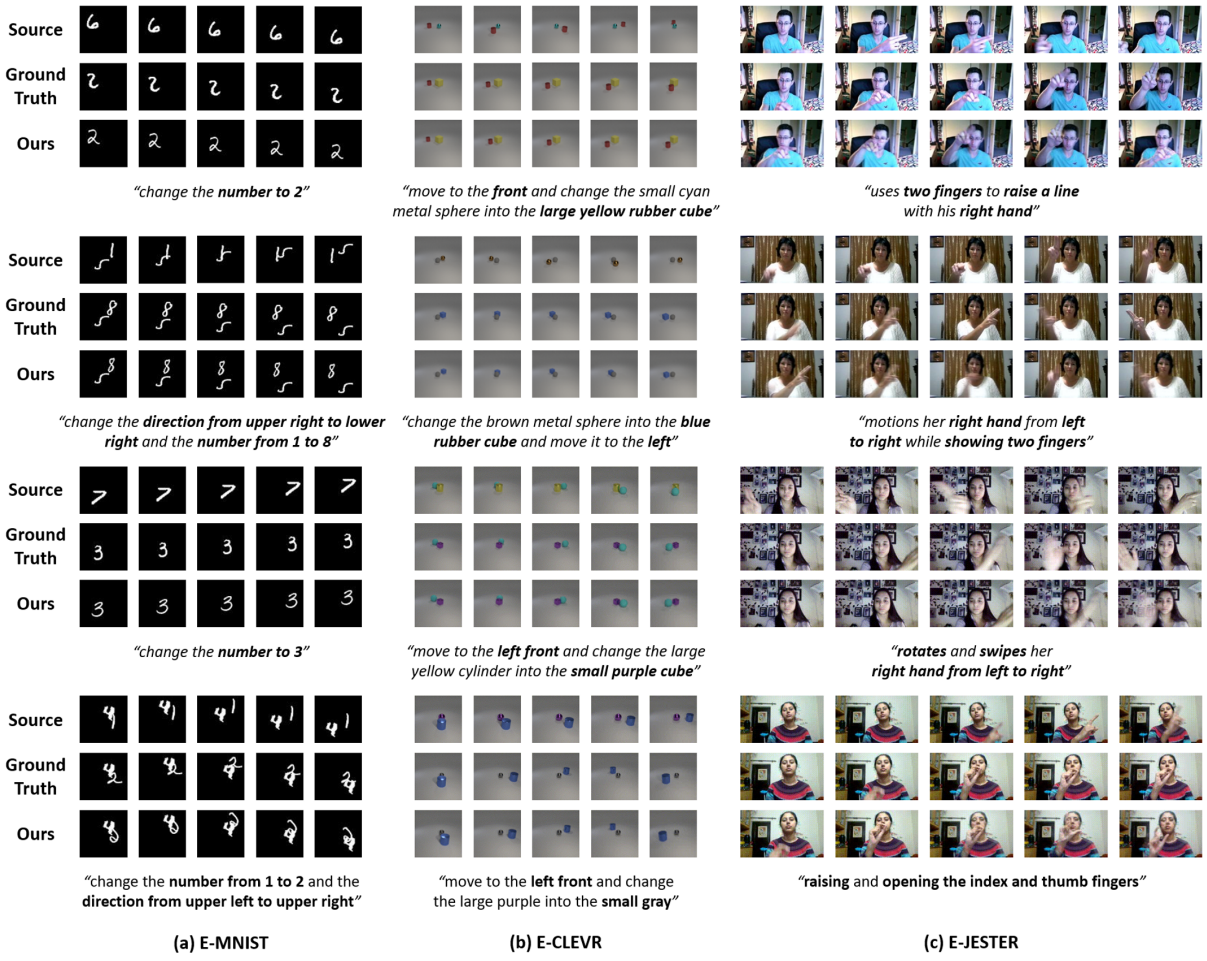


Figure 4.6: The qualitative examples on E-MNIST, E-CLEVR, and E-JESTER.

50% indicate that MLF can help correspond more to the instruction and is also more similar to the ground-truth video. The human evaluation results support that MLF not only helps improve the generating quality but also makes the target video more related to the guided text.

4.4.4 Qualitative Results

Fig. 4.6 shows the keyframes of the generated examples of `LBVE`. For E-MNIST, the instruction only tells the replacing number, but without the style, thus our M^{3}L replaces with another number *2*. The visualization examples of E-CLEVR show that M^{3}L can

change the specific object into the correct properties. It also has the spatial concept that can perceive the final related position and maintain the moving motion. The presented E-JESTER videos indicate that M^{3}L not only preserves a similar scenario (the background and the person) but also generates the visual motion of the corresponding gesture.

4.5 Summary

In this chapter, we introduce the language-based video editing task to edit a source video into a target video, guided by a textual instruction. The semantics are controlled by the text, while the scenario of the source should be preserved. We present multimodal multi-level transformer to fuse dynamic video perception and language understanding in a hierarchical manner. We construct three new benchmarks, including two diagnostic and one natural video with human-labeled text. The experiments indicate that our framework takes the first step into text-guided video editing and can support object replacement and motion alteration.

Chapter 5

Unifying Text-guided Video Completion

5.1 Introduction

Video prediction [21, 22, 114], which is built upon generative video modeling [18, 115, 116], anticipates the future by completing a video from the past frames or a static starting image [117, 118]. However, it may produce various outcomes, resulting in the difficulty of meeting human expectations. For the example in Fig. 5.1(a), the game agent can keep jumping to the right or move back and turn left. The limited guidance from only the first frame is insufficient to tell the intention. On the other hand, compared with video prediction, video rewind and infilling have been rarely studied [119, 120], but they are also crucial to general video completion.

For better controllability and flexibility, we introduce text-guided video completion (`TVC`) in this chapter, where the partial frames and a given instruction jointly guide the video generation. As illustrated in Fig. 5.1(b), we consider three scenarios of video completion: *prediction* from the first frame, *rewind* from the last frame, and *infilling*

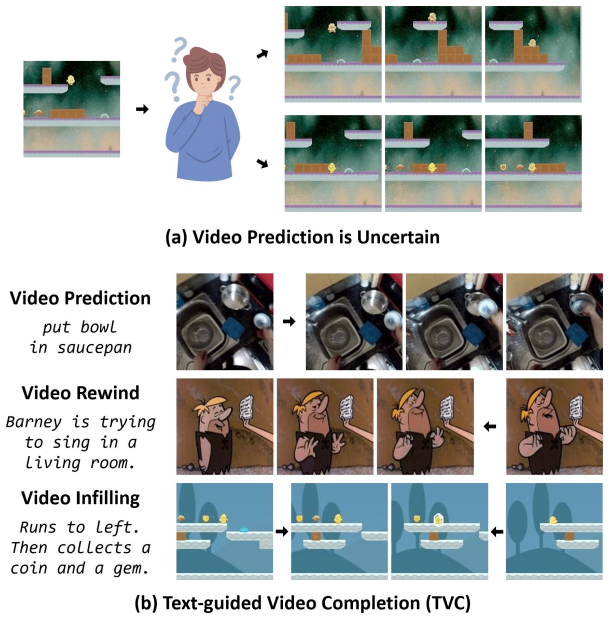


Figure 5.1: The text-guided video completion (TVC) task. (a) Video prediction may have different outcomes. (b) TVC performs video completion from the first frame (prediction), the last frame (rewind), or both (infilling), guided by the textual description.

between the head and tail. The missing (to-be-completed) event should follow the textual instruction, which better mimics how humans imagine after seeing and listening.

To tackle TVC, we present multimodal masked video generation (MMVG) to perform video completion, which represents video frames as discrete visual tokens by temporal-aware VQGAN [121, 122]. MMVG allows visual hints from different time points via the masking strategy, and it learns to produce the full target video. By varying the masking conditions, a single trained model can address all TVC tasks. We conduct the evaluation in diverse video scenarios, including Kitchen [24] (kitchen activities in the first-person view), Flintstones [25] (characters acting the assigned behavior), and MUGEN [26] (an agent playing the game). Experimental results demonstrate that instruction is necessary for controllable video completion, and MMVG unifies TVC with better temporal coherence.

Dataset	Video Scenario	Instruction Example
Kitchen	Egocentric Activity	*put plate on counter*
		turn tap water off
Flintstones	Animation	*Barney is in the living room, reading paper.*
		Wilma is riding in the car and talking.
MUGEN	Adventure Gaming	*Jumps down the stage. It runs from left to right and jumps on a worm.*

5.2 Related Work

Video Prediction There are various generative modeling methods have shown promising results: generative adversarial networks (GAN) [39, 123, 93], autoregressive transformers [105, 124, 125], and diffusion models [126, 127]. Upon that, video prediction [128, 116, 129, 130], which considers past frames to anticipate future observations, should maintain temporal dynamics from static images. Though the overall idea is similar to completing a video from partial frames, other tasks, such as rewind and infilling [131, 120, 119], are not extensively explored.

Text-guided Video Synthesis With large-scale datasets [132, 133, 134], recent pretrained models can generate vivid videos [82, 135, 136, 6, 137]. However, those methods that depend on autoregressive generation can only be guided chronologically. In contrast, `TVC` requires to perform video completion from arbitrary time points. Even if text-guided video-to-video [17, 138] can be controlled by language, it is still conditioned on a full video, where the temporal dynamics are usually provided. Different from them, our MMVG can regain the missing event from just partial guidance.

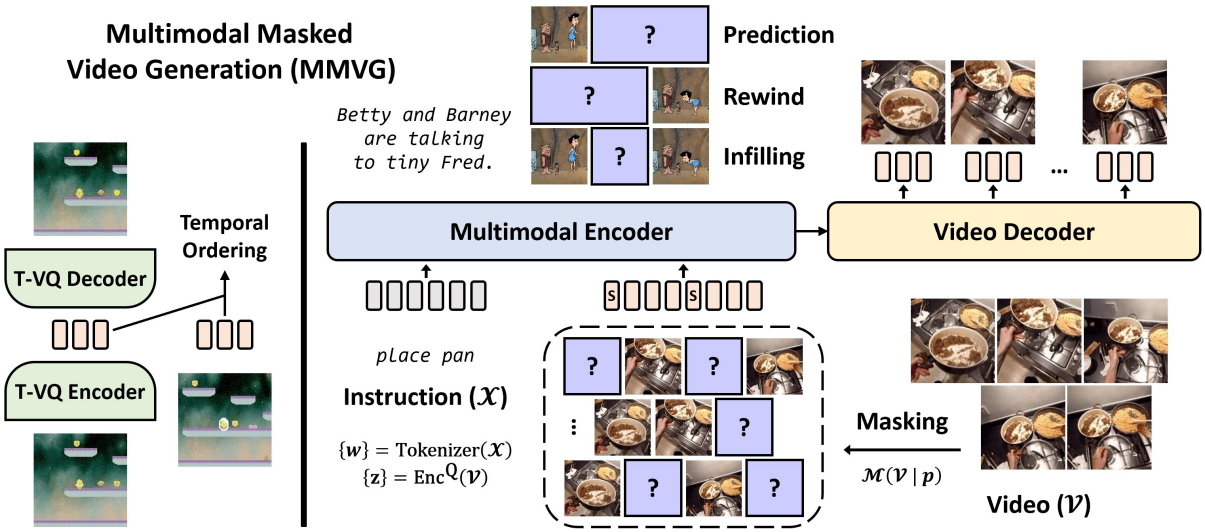


Figure 5.2: The overview of our multimodal masked video generation (MMVG). With temporal-aware VQGAN (T-VQ) for discrete visual representation, MMVG considers the instruction and partial frames from diverse time points through masking and learns to generate the complete video.

5.3 Multimodal Masked Video Generation (MMVG)

5.3.1 Overview

Task Definition TVC performs video completion from the first frame (prediction), the last frame (rewind), or the head and tail (infilling), conditioned on the textual instruction. During training, we have pairs of videos $\mathcal{V}$ and corresponding instructions $\mathcal{X}$. Specifically, $\mathcal{V}$ consists of N frames as $\{v_1, v_2, ..., v_N\}$. Our goal is to train a single unified model that generates the complete $\mathcal{V}$ given the partial frames from arbitrary time points and $\mathcal{X}$.

We illustrate our multimodal masked video generation in Fig. 5.2. To model the video along with language, we propose temporal-aware VQGAN to represent a frame as discrete visual tokens. We then adopt an effective masking strategy that masks different video parts for video completion learning. The multimodal encoder consumes the text and the partial-missing video, and the decoder learns to produce the complete video from arbitrary guided frames. By varying the masking conditions, MMVG learns to unify all

TVC tasks, including prediction, rewind, and infilling during training.

5.3.2 Temporal-aware VQGAN

VQGAN [121, 122] has already shown promising capability in representing data as discrete tokens. If VQGAN is directly applied onto videos, it will ignore the inner temporal coherence and treat each frame as an independent image, resulting in an unsmooth video reconstruction. Although TATS [125] attempts to handle this by making k consecutive frames altogether during VQ, it has to pre-define the constant k before training. Such constraint limits it from representing a frame at any timestamp. To address it with flexibility, we propose temporal-aware VQGAN (T-VQ) to inject the temporal relationship into the latent representation. We first follow VQGAN to learn the target tokens z_i by reconstructing a video frame v_i with the VQ encoder ($\mathrm{Enc}^{\mathrm{Q}}$) and decoder ($\mathrm{Dec}^{\mathrm{Q}}$):

$$
\begin{aligned}
\mathrm{z}_i &= \mathrm{q}(\mathrm{Enc}^{\mathrm{Q}}(v_i) \mid C), \\
\hat{v_i} &= \mathrm{Dec}^{\mathrm{Q}}(\mathrm{z}_i), \\
\mathcal{L}_{\mathrm{VQ}} &= \underbrace{||\hat{v_i} - v_i||_1}_{\text{reconstrcution}} + \underbrace{||\mathrm{sg}[\mathrm{Enc}^{\mathrm{Q}}(v_i)] - C_{\mathrm{z}_i}||_2^2}_{\text{codebook}} \\
&+ \underbrace{\beta||\mathrm{sg}[C_{\mathrm{z}_i}] - \mathrm{Enc}^{\mathrm{Q}}(v_i)||_2^2}_{\text{commit}} + \underbrace{||\mathcal{F}(\hat{v_i}) - \mathcal{F}(v_i)||_1}_{\text{matching}}.
\end{aligned}
\tag{5.1}
$$

The discrete latent code z_i is acquired from the quantization operation q [122], which adopts nearest neighbor search by the trainable codebook C. We consider the straight-through estimator over the stop-gradient operation sg and adopt β as 0.25 [121]. The adversarial training between the frame quality loss $\mathcal{L}_G$ and discrimination loss $\mathcal{L}_D$ are

further calculated from the discriminator D:

$$\begin{aligned} \mathcal{L}_G &= \log(1 - D(\hat{v}_i)), \\ \mathcal{L}_D &= \log(1 - D(\hat{v}_i)) + \log(D(v_i)). \end{aligned} \tag{5.2}$$

To inject the temporal relationship into z, T-VQ is trained with the introduced contrastive temporal reasoning:

$$\begin{aligned} o_i &= \mathrm{FC}^{\mathrm{T}}(\mathrm{z}_i, \mathrm{z}_j), \\ \mathcal{L}_{\mathrm{T}} &= \mathrm{BCELoss}(o_i, 0 \text{ if } i > j \text{ else } 1), \end{aligned} \tag{5.3}$$

where j is a random frame from the same video. FC^{T} is the MLP classifier, and BCELoss is the binary cross-entropy for before/after. By learning the temporal order from $\mathcal{L}_{\mathrm{T}}$, z facilitates an implicit temporal coherence, leading to smooth video modeling.

5.3.3 Generation from Masked Video

We propose the masking strategy $\mathcal{M}$ to obtain the masked videos $\overline{\mathcal{V}}$ from diverse time points. $\mathcal{M}$ masks out most video frames with the probability p and replaces each fragment as a unique [`SPAN`] token. For example, $\mathcal{M}$ reserves the second and the fifth frame, and masks all the others over a video length of 5:

$$\overline{\mathcal{V}} : \{[\mathtt{S}], v_2, [\mathtt{S}], v_5\} = \mathcal{M}(\mathcal{V} \mid p). \tag{5.4}$$

Our goal is to recover the missing part from $\overline{\mathcal{V}}$ and perform video completion, guided by the instruction $\mathcal{X}$.

Multimodal Encoder-Decoder To model between the vision and language modalities, we apply our Enc^{Q} over $\overline{\mathcal{V}}$ for the discrete visual tokens $\{[\texttt{S}], \text{z}_2, [\texttt{S}], \text{z}_5\}$. We also tokenize the text $\mathcal{X}$ into word tokens $\{w_i\}_{i=1}^{L}$ with the CLIP tokenizer [139], where L is the length of $\mathcal{X}$. As the same discrete space, MMVG achieves cross-modal fusion by the multimodal encoder (Enc^{M}) through the self-attention mechanism as the transformer [105]:

$$\begin{aligned} f_i^w, f_j^v &= \text{LP}^w(w_i), \text{LP}^v(\text{z}_j) \\ \{h\} &= \text{Enc}^{\text{M}}([\{f^w\}, \{f^v\}]), \end{aligned} \tag{5.5}$$

where it gets the features f by the linear projection (LP), and h is the encoding features.

After encoding the language hint and the partial-missing video from Enc^{M}, our video decoder (Dec^{M}) learns to produce all frames for comprising the complete video. Dec^{M} follows the vanilla autoregressive decoder, which first conducts self-attention over the past generated tokens and then predicts the discrete visual tokens as the video frame, conditioned on the encoded features h:

$$\begin{aligned} \hat{\text{z}_t} &= \text{Dec}^{\text{M}}(\{\hat{z_1}, ..., \hat{\text{z}_{t-1}}\} \mid \{h\}), \\ \mathcal{L}_t &= \text{CELoss}(\hat{z_t}, \text{z}_t), \\ \mathcal{L}_{\text{M}} &= \sum_{t=1}^{N} \mathcal{L}_t, \end{aligned} \tag{5.6}$$

where z_t is the ground-truth tokens of the frame v_t in the original $\mathcal{V}$. We calculate the video decoding loss $\mathcal{L}_{\text{M}}$ by the cross-entropy (CELoss) to learn video generation as classification. In the end, we can utilize Dec^{Q} to reconstruct all the frames as our completed videos $\hat{\mathcal{V}}$:

$$\hat{\mathcal{V}} = \text{Dec}^{\text{Q}}(\{\hat{\text{z}}\}_{t=1}^{N}). \tag{5.7}$$

Masking Strategy By varying the masking conditions through $\mathcal{M}$, MMVG learns how to complete a video from partial frames $\overline{\mathcal{V}}$ at arbitrary time points with the text, which overcomes the limitation of chronological guidance. To make $\mathcal{M}$ more effective, we apply an adaptive probability p instead of random sampling every time. Each video $\mathcal{V}$ keeps its own p, and all frames are equally initialized in the beginning. Based on the prediction error, we adjust the masking probability p_t of the t-th frame:

$$p_t = p_t + \alpha((\frac{\mathcal{L}_t}{\mathcal{L}_M}\sum p) - p_t), \tag{5.8}$$

where α is the adjusting rate. A larger video decoding loss $\mathcal{L}_t$ indicates that the t-th frame is more difficult to recover. MMVG learns more from those challenging cases and can bring better generative quality for video completion.

Unifying `TVC` during Inference After training with text and partial-missing video, MMVG learns to perform video completion over [`SPAN`] tokens. Then for inference, Enc^{M} takes the following as its input to support different tasks:

- `TVP`rediction: $[\{w\}, \{\text{z}_1, [\texttt{SPAN}]\}]$
- `TVR`ewind: $[\{w\}, \{[\texttt{SPAN}], \text{z}_N\}]$
- `TVI`nfilling: $[\{w\}, \{\text{z}_1, [\texttt{SPAN}], \text{z}_N\}]$

In this way, a single trained MMVG can unify all `TVC` tasks without the specific downstream fine-tuning.

5.3.4 Learning of MMVG

Algo. 4 illustrates the learning process of MMVG. We first train T-VQ over video frames for discrete visual tokens with contrastive temporal reasoning. Specifically, we minimize the VQ reconstruction loss $\mathcal{L}_{\text{VQ}}$ and frame quality loss $\mathcal{L}_G$ along with our

Algorithm 4 Multimodal Masked Video Generation (MMVG)

Enc^{Q}, Dec^{Q}: Temporal-aware VQ Encoder / Decoder
Enc^{M}, Dec^{M}: Multimodal Encoder / Decoder

Pre-train Enc^{Q} and Dec^{Q}
while TRAIN_MMVG **do**
    $\mathcal{V}$, $\mathcal{X}$, $p \leftarrow$ sampled video / instruction / probability
    $\overline{\mathcal{V}}$: $\{v_a, [\texttt{S}], v_b, ...\} = \mathcal{M}(\mathcal{V} \mid p)$ ▷ diverse guided masked frames
    $\{\text{z}_a, [\texttt{S}], \text{z}_b, ...\}, \{w\} = \text{Enc}^{\text{Q}}(\overline{\mathcal{V}})$, Tokenizer$(\mathcal{X})$
    $\{h\} = \text{Enc}^{\text{M}}([\{w\}, \{\text{z}_a, [\texttt{S}], \text{z}_b, ...\}])$
    for $t \leftarrow 1$ to N **do**
        $\hat{\text{z}}_t = \text{Dec}^{\text{M}}(\{\text{z}_1, ..., \text{z}_{t-1}\} \mid \{h\})$ ▷ teacher-forcing
        $\mathcal{L}_t \leftarrow$ video decoding loss ▷ Eq. 5.6
    end for
    $\hat{\mathcal{V}} = \text{Dec}^{\text{Q}}(\{\hat{\text{z}}_{t=1}^{N}\})$

    $\mathcal{L}_{\text{M}} = \sum_{t=1}^{N} \mathcal{L}_t$
    Update Enc^{M} and Dec^{M} by minimizing $\mathcal{L}_{\text{M}}$
    $p \leftarrow$ update masking probability ▷ Eq. 5.8
end while

temporal ordering loss $\mathcal{L}_{\text{T}}$ to optimize T-VQ. At the same time, we also update the discriminator D via the standard adversarial training by maximizing the discrimination loss $\mathcal{L}_D$. For video completion, the masking strategy $\mathcal{M}$ masks the video frames with the probability p and then acquires guided frames from diverse time points. MMVG regards text and partial-missing video by Enc^{M} for cross-modal fusion, and Dec^{M} further predicts the visual tokens of frames autoregressively as the complete video. As a sequential generation process, we apply the teacher-forcing trick, where the ground-truth z from the previous timestamp is fed to stabilize the training. Each video decoding loss $\mathcal{L}_t$ at timestamp t is summed up as $\mathcal{L}_{\text{M}}$ to optimize MMVG. According to $\mathcal{L}_t$, we update p for effective masking probability. The overall optimization can be summarized as:

$$\begin{aligned} \text{T-VQ:} &\min_{\text{Enc}^{\text{Q}},\text{Dec}^{\text{Q}},C,\text{FC}^{\text{T}}} \max_{D} \mathcal{L}_{\text{VQ}} + \mathcal{L}_G + \mathcal{L}_D + \mathcal{L}_{\text{T}} \\ \text{MMVG:} &\min_{\text{Enc}^{\text{M}},\text{Dec}^{\text{M}}} \mathcal{L}_{\text{M}} \end{aligned} \tag{5.9}$$

5.4 Experiments

5.4.1 Experimental Setup

Datasets We consider diverse video scenes with natural instructions for `TVC`:

- Kitchen [24] records 22K egocentric videos about kitchen activity, which have different lengths (4-16 frames) with narrations;
- Flintstones [25] contains 25K animation videos (15 frames) from *The Flintstones*, where each video description includes the characters and their behavior;
- MUGEN [26] is built from agents playing CoinRun [140], which consists of 375K gaming videos (16 frames) with detailed text annotations.

All video frames are resized into 128x128.

Evaluation Metrics We use the following metrics to evaluate `TVC` results: (i) FVD [141] computes the video features [111] distance to the ground truth; (ii) RCS [82] is the relative visual-text similarity to the instruction, compared to the ground-truth video. We fine-tune the CLIP model [139] on each dataset and adapt it to the video scene for a more precise alignment. Apart from automatic metrics, we also conduct a human evaluation (75 examples for each method) from aspects of video quality, instruction relevance, and ground-truth similarity.

Baselines

- VideoMAE [142] reconstructs the missing video cubes and performs `TVC` by masking all video frames except the first or the last (or both);
- TATS [125] produces videos as discrete visual tokens. Since TATS can only consider the past autoregressively, it requires specific training for each task.

We have MMVG$^{\text{U}}$ as the unified model that can support all `TVC` tasks with a single

Method	Text	Kitchen		Flintstones		MUGEN	
		FVD↓	RCS↑	FVD↓	RCS↑	FVD↓	RCS↑
VideoMAE [142]	✗	328.9	47.6	317.5	55.6	548.7	7.0
TATS [125]	✗	106.9	64.4	127.5	60.3	376.5	7.1
MMVG^{U}	✗	105.6	63.3	124.8	60.5	374.5	7.2
MMVG^{S}	✗	**103.8**	**64.5**	**123.8**	**60.8**	**369.4**	**7.3**
TATS [125]	✓	87.2	66.3	115.9	70.6	90.1	67.9
MMVG^{U}	✓	80.2	68.4	108.2	72.9	84.8	70.2
MMVG^{S}	✓	**75.6**	**68.8**	**106.3**	**73.7**	**83.3**	**71.1**

Table 5.1: The testing results of TVPrediction on Kitchen, Flintstones, and MUGEN.

training and MMVG^{S} to further train for each prediction, rewind, and infilling.

Implementation Detail For the vector quantization in T-VQ, we use a patch size 16, where a 128x128 video frame transforms into 8x8 discrete visual tokens. There are 1024 vocabularies in the codebook C, and the hidden embedding size is 256. We optimize T-VQ by Adam [55] with the learning rate 4.5e-6. MMVG is built in an encoder-decoder manner, where Enc^{M} is a transformer with 24 layers, 16 attention heads, and the hidden embedding size is 1024. Dec^{M} adopts a similar setting. The initial sample rate p of the masking strategy $\mathcal{M}$ is 0.9 with an adjusting rate α as 0.1. We optimize MMVG through the mixed precision [143] also with the learning rate 4.5e-6.

5.4.2 Main Results

TVPrediction As presented in Table 5.1, VideoMAE attempts to produce all frames at once, which makes it difficult to maintain video temporal consistency (*e.g.*, high 328.9 FVD on Kitchen). TATS is inherently designed for prediction as it generates the frames one after one. However, our unified MMVG^{U} still performs better than TATS, which supports that learning from diverse time points will not hurt the prediction from the past.

Method	Text	Kitchen		Flintstones		MUGEN	
		FVD↓	RCS↑	FVD↓	RCS↑	FVD↓	RCS↑
VideoMAE [142]	✗	365.9	48.2	335.5	55.9	545.2	7.1
TATS [125]	✗	<u>107.7</u>	<u>62.7</u>	127.6	<u>60.2</u>	<u>350.8</u>	<u>7.2</u>
MMVG^{U}	✗	109.8	62.6	<u>124.3</u>	59.7	356.4	7.0
MMVG^{S}	✗	**105.9**	**63.6**	**123.8**	**60.5**	**347.8**	**7.2**
TATS [125]	✓	89.8	63.3	116.3	70.4	<u>89.8</u>	<u>68.7</u>
MMVG^{U}	✓	<u>83.2</u>	<u>66.9</u>	<u>113.2</u>	<u>71.6</u>	93.1	68.4
MMVG^{S}	✓	**79.7**	**68.1**	**107.2**	**72.9**	**88.7**	**70.0**

Table 5.2: The testing results of TVRewind, where TATS requires a specific training for TVR.

Without text, there are too many possible outcomes from just the beginning, where the predicted results may not meet the expectation (*e.g.*, high 370 FVD on MUGEN). The instruction as guidance makes it related to the expected ground-truth result. We can drive MUGEN to run, jump, or collect coins as the textual descriptions to achieve more controllability (*e.g.*, lower 84.8 FVD by MMVG^{U}). The higher 70.2 RCS also shows that it can produce videos that match the instruction. The specific trained MMVG^{S} further improves itself through training prediction as completion from the head.

TVRewind Rewind from the last allows the model to imagine what happened along with an appropriate opening. In addition, the objects may not display on the last frame (*e.g.*, the spoons and forks for "*close drawer*"), which makes it more challenging to complete. Similar to prediction, in Table 5.2 VideoMAE cannot have feasible rewind results. Language is still essential to remind the past and establish an adequate beginning, where we can find a significant performance gap between with and without text (*e.g.*, 90 *vs.* 350 FVD on MUGEN). Our unified MMVG^{U} achieves comparable results to TATS and even outperforms on Kitchen and Flintstones. Thanks to the learning of completion from partial frames, the autoregressive model can still accomplish video rewind without specific training. If following TATS design to train MMVG^{U} for rewind, MMVG^{S} gains

Method	Text	Kitchen		Flintstones		MUGEN	
		FVD↓	RCS↑	FVD↓	RCS↑	FVD↓	RCS↑
VideoMAE [142]	✗	246.9	54.7	211.5	60.6	494.9	7.8
TATS [125]	✗	71.5	72.7	119.5	66.7	328.2	8.4
MMVG^{U}	✗	71.5	73.4	121.8	66.3	328.4	7.8
MMVG^{S}	✗	**68.5**	**73.6**	**118.5**	**67.9**	**324.3**	**8.4**
TATS [125]	✓	57.4	77.6	95.8	78.2	58.9	73.6
MMVG^{U}	✓	59.8	77.8	92.8	78.3	59.2	73.2
MMVG^{S}	✓	**56.0**	**78.1**	**91.6**	**79.6**	**57.2**	**74.1**

Table 5.3: The testing results of `TVInfilling`, where TATS requires a specific training for `TVR`.

more improvement and utterly surpasses it.

`TVInfilling` With guidance from the head and tail, we find a noticeable improvement even without instruction (*e.g.*, lower FVDs on Kitchen) in Table 5.3, which is helpful in temporal video modeling. To capture the precise missing event, we still require the language hint for more controllability. Our unified MMVG^{U} achieves comparable performance to TATS again, where the latter is specifically trained for the infilling task. It indicates that completion from partial frames at different time points still helps, and MMVG^{S} further outperforms on `TVInfilling`.

5.4.3 Ablation Study

Video Generation/Prediction We further evaluate the classic video generation on UCF-101 [144], which is more challenging to generate natural human videos. Table 5.4 supports that our MMVG can produce videos with higher visual similarity (higher IS) as well as temporal alignment (lower FVD) to the ground truth. For video prediction, we apply the widely-used BAIR [145], where the model has to anticipate how a robot pushes objects from the given first frame. MMVG again surpasses TATS. Though both

Method	UCF-101		BAIR
	IS↑	FVD↓	FVD↓
VidoeGPT [124]	24.7	-	103.3
TATS [125]	57.6	420	88.6
MMVG	**58.3**	**395**	**85.2**

Table 5.4: The testing results of video generation on UCF-101 and prediction on BAIR.

Method	UCF-101 (FVD↓)				
	K=+1	+2	+5	±1	±2
RaMViD [119]	349.7	300.6	260.5	215.4	162.5
MMVG	**316.3**	**258.5**	**194.6**	**183.2**	**120.3**

Table 5.5: The testing results of video prediction and infilling on UCF-101.

generation and prediction generate video frames chronologically, the ability to recover arbitrary missing frames for video completion still empowers MMVG with a stronger temporal coherence.

Video Infilling We follow RaMViD [119] to evaluate video infilling in Table 5.5, where K=+1 means given the starting frame, and K=±2 should provide the first and last two frames. MMVG outperforms RaMViD on all K, and the performance gap gets even larger when more guided frames are accessible. Despite having a similar masking strategy, it shows that generating frames autoregressively still brings superior results.

Text-to-Video Generation We pre-train MMVG using WebVid [134], which contains 2.5M natural text-video pairs. We adopt the masking strategy to treat the pre-training as video completion in Table 5.6. Surprisingly, MMVG can generate videos that are more related to the texts on MSRVTT [146] than CogVideo, even though using twice less data. These results encourage the effectiveness of completion from partial frames.

Human Evaluation We study the video quality (Q.), the relevance to the instructions (T.), and the similarity to the ground-truth video (GT) of the produced videos from the human aspect (75 samples for each method). Table 5.7 is calculated as the mean ranking score (from 1 to 3, the higher is better) on MUGEN. MMVG without text even generates

Zero-shot		MSRVTT	
Method	#Pre-train	FID↓	CLIP-S↑
NUWA [135]	3.9M	47.7	0.2439
CogVideo [136]	5.4M	<u>23.6</u>	<u>0.2631</u>
MMVG	2.5M	**23.4**	**0.2644**

Table 5.6: The testing results of zero-shot text-to-video generation on MSRVTT.

Method	Text	Q.	T.	GT
MMVG	✗	**2.03**	1.56	1.55
TATS [125]	✓	1.94	<u>2.11</u>	<u>2.19</u>
MMVG	✓	<u>2.03</u>	**2.33**	**2.36**

Table 5.7: The human evaluation of TVP on MUGEN from video quality (Q.), instruction relevance (T.), and ground-truth similarity (GT).

higher-quality videos than TATS with text, which supports that completion from partial frames benefits temporal coherence. However, the lowest GT illustrates that language guidance is crucial for controllability. With instruction, MMVG can anticipate the future as the text (the highest T.) and meet the ground truth (the highest GT).

5.4.4 Qualitative Results

We demonstrate the qualitative examples for TVC on Kitchen, Flintstones, and MUGEN in Fig. 5.3. We also depict video generation on UCF-101, video prediction on BAIR, and text-to-video prediction on WebVid. The details are discussed in the caption.

5.5 Summary

In this chapter, we introduce the text-guided video completion task that generates the full video from partial frames, controlled by instruction. We present multimodal masked video generation with an effective masking strategy to learn visual guidance at any time point. By varying the masking conditions, our framework can deal with all prediction, rewind, and infilling tasks within a single model. Experiments on various video scenarios show that we can achieve effective language-guided video completion as well as generative video modeling.

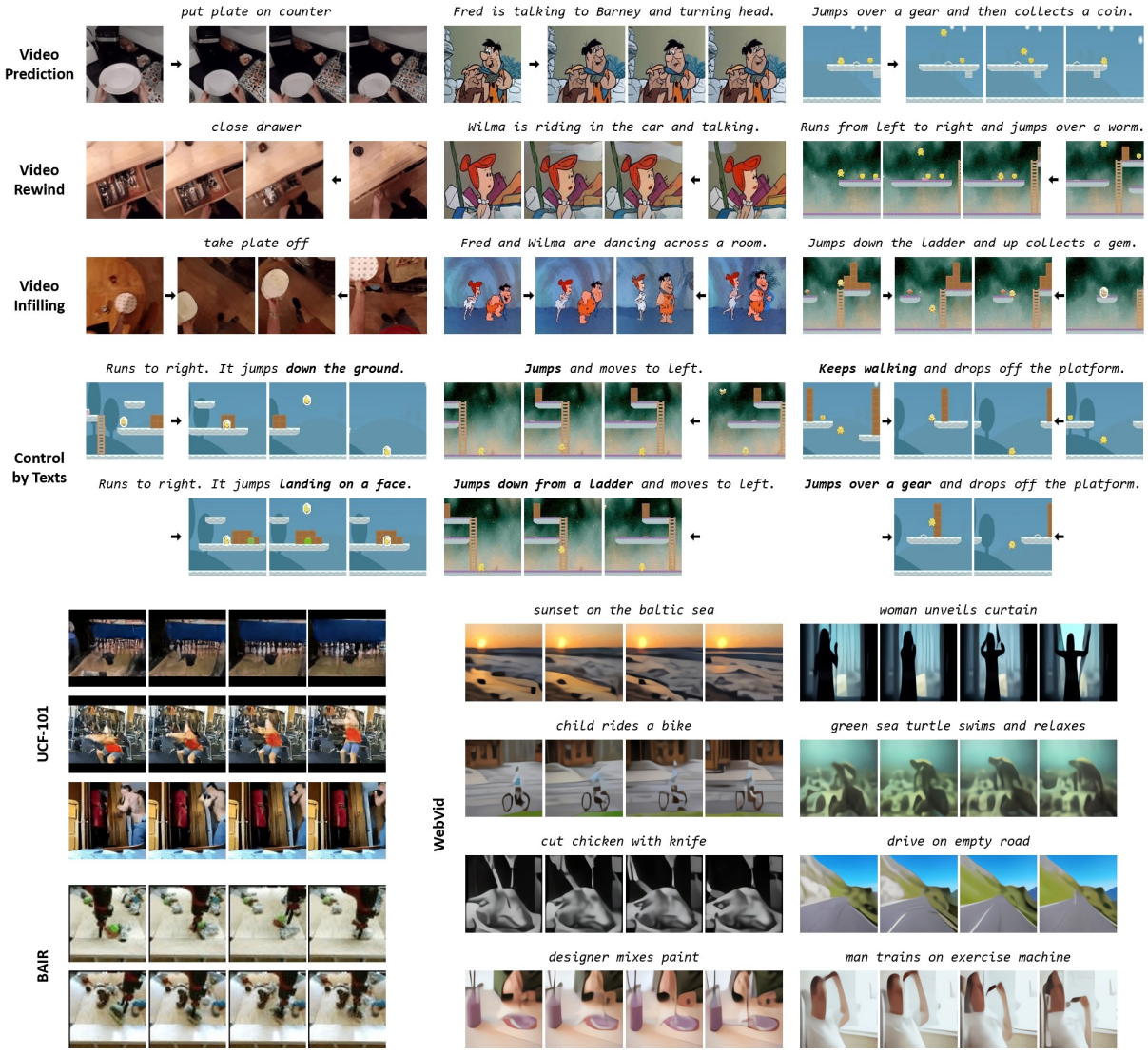


Figure 5.3: The qualitative examples on Kitchen, Flintstones, and MUGEN. We also illustrate video generation on UCF-101, video prediction on BAIR, and text-to-video prediction on WebVid. Thanks to the learning of completion from partial frames at diverse time points, a single trained MMVG can support all TVC tasks. From the same visual guidance, MMVG controls TVC results using different texts. We make it "*jump down the ground*" or "*land on a face*". We also have the behavior as "*keep walking*" or "*jump over a gear*" to recover the missing middle event. Furthermore, MMVG can use language to produce natural dynamics for diverse scenes.

Part III

Toward Natural Visual Manipulation

Chapter 6

Text-guided 3D Human Generation

6.1 Introduction

3D human modeling [147, 148] has been widely used for engaging interaction in gaming, film, and animation. The customization of these characters is crucial for creativity and scalability, which highlights the importance of controllability. In this chapter, we introduce text-guided 3D human generation (`T3H`), as illustrated in Fig. 6.1. `T3H` generates a 3D human with a customized outfit, guided via the fashion description. Though text-to-3D [149, 150] has shown attractive 3D generation results through the success of neural rendering [151], these methods apply iterative inference optimization by external guidance, which is inefficient for usage.

To tackle these above issues, we propose compositional cross-modal human (CCH) to learn `T3H` from 2D-only image collections. Inspired by EVA3D [152], CCH divides the human body into different parts and employs individual volume rendering. We extract the fashion semantics from the description and adopt cross-modal attention to fuse body volumes with textual features, where each part can learn to perceive its correlated fashion patterns. Then these compositional volumes can jointly render a 3D human with the

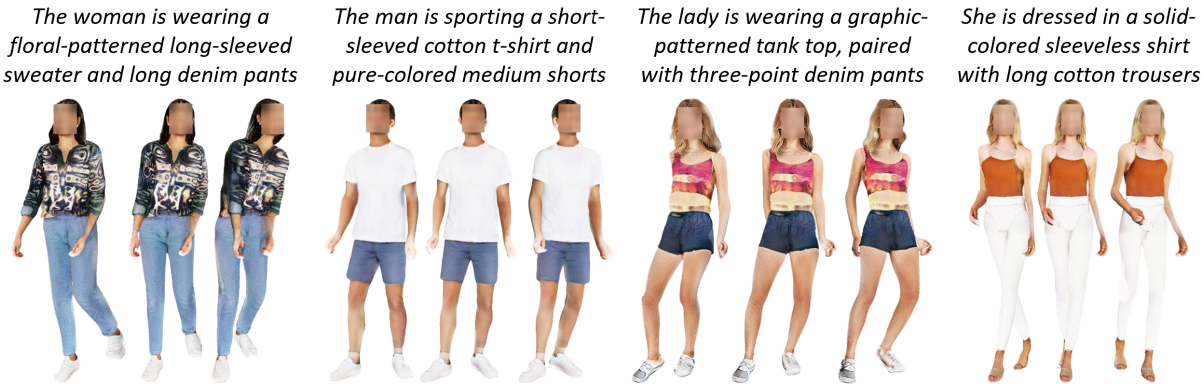


Figure 6.1: The text-guided 3D human generation (T3H) task, which generates a 3D human with a customized outfit, guided by the fashion description.

desired fashion efficiently.

We conduct experiments on DeepFashion [28, 153] and SHHQ [29], which contain human images with diverse fashion descriptions. The patterns include various types of shapes (*sleeveless*, *medium short*, *long*, etc.), fabrics (*denim*, *cotton*, *furry*, etc.), and colors (*floral*, *graphic*, *pure color*, etc.) for the upper and lower clothing. The experiments indicate that language is necessary to make 3D human generation controllable, and our CCH can lead to an effective and efficient T3H.

Fashion Attribute	Instruction Example
Shape/Fabric/Color	*She is dressed in a long-sleeved chiffon shirt with striped three-point shorts*
	He is sporting a short-sleeved cotton t-shirt and graphic-patterned long trousers

6.2 Related Work

3D Generation Different representations have been explored for 3D shapes, such as mesh [154, 155], voxel grid [156, 157], point cloud [158, 159], and implicit field [160, 161]. With the differentiable neural rendering of neural radiance field (NeRF) [151, 162], it can

be guided by various objectives. Text-to-3D draws appreciable attraction via external text-visual alignments [149, 163] and pre-trained text-to-image [150, 7]. However, existing methods take numerous iterations to optimize a NeRF model, which is time-consuming for practical usage.

3D Human Generation To reconstruct a 3D human, early works count on off-the-shelf tools to predict the camera depth [164, 165] or estimate a 3D human texture [166, 167] via the UV mapping [168]. With the promising success of NeRF, recent works [169, 170] adopt volume rendering for 3D humans from multi-view videos [171, 172]. Since the data are difficult to collect, the 3D-aware generation [173, 174] learns 3D modeling from the collection of human images [175, 152]. In place of arbitrary outputs, we introduce the first controllable 3D human generation that also learns from a 2D collection.

6.3 Compositional Cross-modal Human (CCH)

6.3.1 Overview

Task Definition For data efficiency, a 2D collection $\mathcal{D} = \{\mathcal{V}, \mathcal{T}\}$ is provided, where $\mathcal{V}$ is the human image, and $\mathcal{T}$ is its fashion description. The goal of `T3H` is to learn the neural rendering that maps $\mathcal{T}$ into a 3D human with the fashion patterns of $\mathcal{V}$.

Background Neural radiance field (NeRF) [151] defines implicit 3D as $\{c, \sigma\} = F(x, d)$. The query point x in the viewing direction d holds the emitted radiance c and the volume density σ. To get the RGB value $C(r)$ of certain rays $r(t)$, volume rendering is calculated

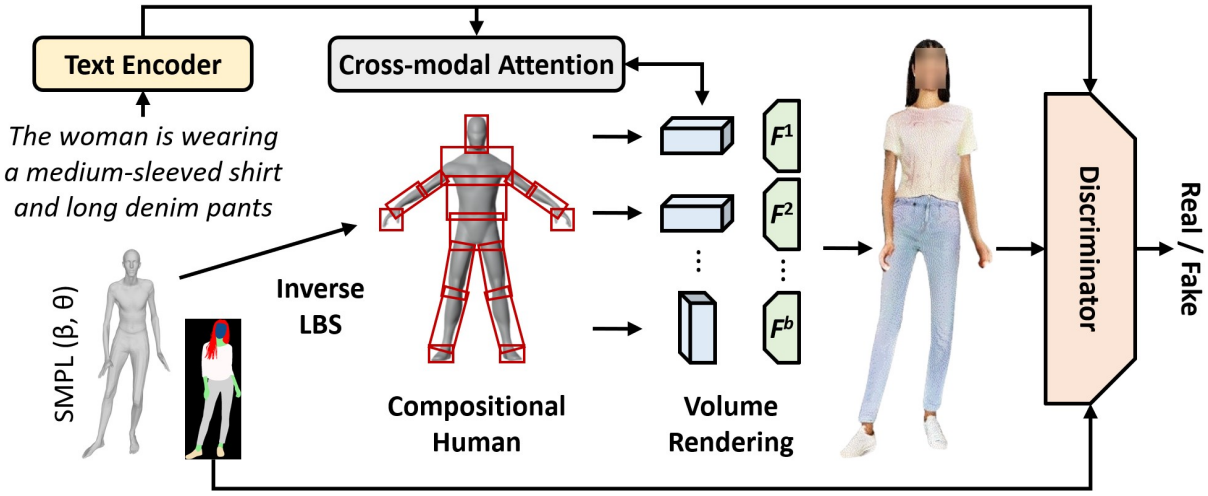


Figure 6.2: The overview of our compositional cross-modal human (CCH). CCH extracts fashion semantics from the description and adopts cross-modal attention in compositional body volumes for controllable 3D human rendering.

along a ray r from the near bound t_{n} to the far bound t_{f}:

$$T(t) = \exp(-\int_{t_{\mathrm{n}}}^{t} \sigma(r(s))ds),$$
$$C(r) = \int_{t_{\mathrm{n}}}^{t_{\mathrm{f}}} T(t)\sigma(r(t))c(r(t), d)dt, \tag{6.1}$$

where $T(t)$ stands for their accumulated transmittance. StyleSDF [176] then replaces σ with single distance field (SDF) $d(x)$ for a better surface, where $\sigma(x) = \alpha^{-1}\mathrm{sigmoid}(\frac{-d(x)}{\alpha})$ and α is a learnable scalar that controls the tightness of the density around the boundary.

SMPL [177] defines the human body as $\{\beta, \theta\}$ to control its shape and pose. We consider Linear Blend Skinning (LBS) as the transformation from the canonical into the observation space for the point x to $\sum_{k=1}^{K} h_k H_k(\theta, J)x$, where h_k is the scalar of the blend weight and H_k is the transformation matrix of the kth joint. Inverse LBS transforms the observation back to the canonical space as a similar equation but with an inverted H.

6.3.2 Compositional Human

Following EVA3D [152], we split the human body into 16 parts. As shown in Fig. 6.2, each body part holds its own bounding box $\{o^b_{\min}, o^b_{\max}\}$. To leverage the human prior for a target pose θ, we transform these pre-defined bounding boxes with SMPL's transformation matrices H_k. Ray $r(t)$ is sampled for each pixel on the canvas. For a ray that intersects bounding boxes, we pick up its near and far bounds (t_{n} and t_{f}) and sample N points as follows: $t_i \sim \mathcal{U}\left[t_{\mathrm{n}} + \frac{i-1}{N}(t_{\mathrm{f}} - t_{\mathrm{n}}), t_{\mathrm{n}} + \frac{i}{N}(t_{\mathrm{f}} - t_{\mathrm{n}})\right]$.

We then transform these sampled points back to the canonical space with inverse LBS. For shape generalization, we consider not only pose transformation but also blend shapes ($B^P(\theta)$ and $B^S(\beta)$) [178]. $\mathbb{N}$ contains K nearest vertices v of the target SMPL mesh for the sample point ray $r(t_i)$:

$$
\begin{aligned}
g_k &= \frac{1}{||r(t_i) - v_k||}, \\
M_k &= \left(\sum_{k=1}^{K} g_k H_k\right) \begin{bmatrix} I & B^P_k + B^S_k \\ 0 & I \end{bmatrix}, \\
\begin{bmatrix} x_i \\ 1 \end{bmatrix} &= \sum_{v_k \in \mathbb{N}} \frac{g_k}{\sum_{v_k \in \mathbb{N}} g_k} (M_k)^{-1} \begin{bmatrix} r(t_i) \\ 1 \end{bmatrix},
\end{aligned} \tag{6.2}
$$

where g_k is the inverse weight of the vertex v_k and M_k is the transformation matrix. The x_i can be used for further volume rendering.

6.3.3 Cross-modal Attention

During rendering, if the canonical point x_i with the viewing direction d_i is inside the bth bounding box, it will be treated as:

$$\begin{aligned} \hat{x}_i^b &= \frac{2x_i - (o_{\max}^b + o_{\min}^b)}{o_{\max}^b - o_{\min}^b}, \\ f_i^b &= \text{Linear}(\hat{x}_i^b, d_i), \end{aligned} \tag{6.3}$$

where a linear mapping is applied to acquire preliminary features f^b. To exhibit the desired fashion in the final rendering, we extract the word features by the text encoder as $\{w_l\}$ from $\mathcal{T}$. We then fuse the textual features with f_i^k via cross-modal attention:

$$\begin{aligned} p_l &= \frac{\exp(f_i^b \; W^b \; w_l^{\mathrm{T}})}{\sum_{\iota=1}^{L} \exp(f_i^b \; W^b \; w_\iota^{\mathrm{T}})}, \\ \text{CA}(f_i^b \mid \{w\}) &= \sum_{l=1}^{L} p_l w_l, \end{aligned} \tag{6.4}$$

where L is the length of $\mathcal{T}$ and W^b is the learnable matrix.

Each body part has its individual volume rendering F^b, which consists of stacked multilayer perceptrons (MLPs) with the SIREN activation [179]. Since the point x_i may fall into multiple boxes $\mathbb{B}_i$, we follow EVA3D to apply the mixture function [180]:

$$\begin{aligned} \{c_i^b, \sigma_i^b\} &= F^b(\text{CA}(x_i^b, d_i \mid \{w\})), \\ u_b &= \exp(-m(\hat{x}_i^b(x)^n + \hat{x}_i^b(y)^n + \hat{x}_i^b(z)^n)), \\ \{c_i, \sigma_i\} &= \frac{1}{\sum_{b\in\mathbb{B}} u_b} \sum_{b\in\mathbb{B}} u_b \{c_i^b, \sigma_i^b\}, \end{aligned} \tag{6.5}$$

where m and n are hyperparameters. With $\{c_i, \sigma_i\}$, we adopt Eq. 6.1 to render the RGB value of ray $r(t)$. Through all sampled rays r, we then have our final human rendering

R, where the overall process can be simplified as $R = G(\beta, \theta \mid \mathcal{T})$.

Semantic Discrimination For a ground-truth $\{\mathcal{V}, \mathcal{T}\}$, we parse the 2D human image as the segmentation $\mathcal{S}$ [181], which provides the reliable body architecture. To obtain its fashion map Q, we apply cross-modal attention between $\mathcal{S}$ and $\mathcal{T}$:

$$\begin{aligned} \{e_{i,j}\} &= \mathrm{Conv}(\mathcal{S}), \\ Q_{i,j} &= \sum_{l=1}^{L} \frac{\exp(e_{i,j} W w_l^{\mathrm{T}})}{\sum_{\iota=1}^{L} \exp(e_{i,j} W w_\iota^{\mathrm{T}})} w_l, \end{aligned} \tag{6.6}$$

where e is the same dimension as f, W is the learnable attention matrix, and Q perceives which human body part should showcase what fashion patterns. We concatenate the rendered human R (or the ground-truth $\mathcal{V}$) with Q and feed them into our discriminator D to perform binary classification:

$$D(R \mid \mathcal{T}) = \mathrm{BC}([\mathrm{Conv}(R), Q]), \tag{6.7}$$

where D can provide alignments of both the human pose and fashion semantics.

6.3.4 Learning of CCH

We include the non-saturating loss with R1 regularization [182] for adversarial learning over the ground-truth $\{\mathcal{V}\}$:

$$\begin{aligned} U(u) &= -\log(1 + \exp(-u)), \\ \mathcal{L}_{\mathrm{adv}} &= U(G(\beta, \theta \mid \mathcal{T}) \mid \mathcal{T}) \\ &\quad + U(-D(\mathcal{V} \mid \mathcal{T})) + \lambda |\nabla D(\mathcal{V} \mid \mathcal{T})|^2. \end{aligned} \tag{6.8}$$

Algorithm 5 Compositional Cross-modal Human (CCH)

G, D: Generator / Discriminator

while TRAIN_CCH **do**
  $\mathcal{V}$, $\mathcal{T} \leftarrow$ sampled human / description
  $\{\beta, \theta\} \leftarrow$ estimated SMPL parameters of $\mathcal{V}$
  $\{x_i\} \leftarrow$ canonical points via inverse LBS ▷ Eq. 6.2
  $f_i^b \leftarrow$ rendering features inside the bth box ▷ Eq. 6.3
  $\{w_l\} \leftarrow$ extracted textual features of $\mathcal{T}$
  CA $\leftarrow$ fusion via cross-modal attention ▷ Eq. 6.4
  $\{c_i, \sigma_i\} \leftarrow$ mixture radiance / density ▷ Eq. 6.5
  $R \leftarrow$ final rendering human ▷ Eq. 6.1

  $\mathcal{L}_{\text{adv}}$, $\mathcal{L}_{\text{off}}$, $\mathcal{L}_{\text{eik}} \leftarrow$ discrimination / offset / derivation loss ▷ Eq. 6.9
  $\mathcal{L}_{\text{all}} \leftarrow$ overall training loss ▷ Eq. 6.10
  Update G by minimizing $\mathcal{L}_{\text{all}}$
  Update D by maximizing $\mathcal{L}_{\text{all}}$
end while

Following EVA3D, we also append the minimum offset loss $\mathcal{L}_{\text{off}}$ to maintain a plausible human shape as the template mesh. $\mathcal{L}_{\text{eik}}$ penalizes the derivation of delta SDFs to zero and makes the estimated SDF physically valid [183]:

$$\begin{aligned} \mathcal{L}_{\text{off}} &= ||\Delta d(x)||_2^2, \\ \mathcal{L}_{\text{eik}} &= ||\nabla(\Delta d(x))||_2^2. \end{aligned} \tag{6.9}$$

The learning process of our CCH is at Algo. 5, where the overall optimization can be:

$$\begin{aligned} \mathcal{L}_{\text{all}} &= \mathcal{L}_{\text{adv}} + 1.5 \cdot \mathcal{L}_{\text{off}} + 0.5 \cdot \mathcal{L}_{\text{eik}}, \\ &\min_G \max_D \mathcal{L}_{\text{all}}. \end{aligned} \tag{6.10}$$

6.4 Experiments

6.4.1 Experimental Setup

Datasets We coduct experiments on DeepFashion [153] and SHHQ [29] for `T3H`. DeepFashion contains 12K human images with upper and lower clothing descriptions. Since there are no annotations in SHHQ, we first fine-tune GIT [184] on DeepFashion and then label for 40K text-human pairs. We follow OpenPose [185] and SMPLify-X [186] to estimate the human keypoints and its SMPL parameters. The resolution is resized into 512x256, where all faces are blurred prior to training.

Evaluation Metrics We apply metrics from both visual and semantic prospects. we adopt frechet inception distance (FID) [81] and Depth [187] to calculate visual and geometry similarity, compared to the ground-truth image. We treat percentage of correct keypoints (PCK@0.5) [188] as the correctness of the generated pose. To investigate the textual relevance, we follow CLIP score (CLIP-S) [189] for the text-visual similarity, where CLIP is fine-tuned for a more accurate alignment. We further train a fashion classifier on DeepFashion labels[1] and assess fashion accuracy (FA) of the generated human.

Baseliens

- Latent-NeRF [7] brings NeRF to the latent space and guides its generation by the given object and a text-to-image prior;
- TEXTure [190] paints a 3D object from different viewpoints via leveraging the pre-trained depth-to-image diffusion model;
- CLIP-O is inspired by AvatarCLIP [191], which customizes a human avatar from the description with CLIP text-visual alignment. We apply this guidance to optimize

[1]There are six targets for FA, including the shape, fabric, and color of the upper and lower clothing.

Method	DeepFashion					SHHQ				
	FID↓	Depth↓	PCK↑	CLIP-S↑	FA↑	FID↓	Depth↓	PCK↑	CLIP-S↑	FA↑
L-NeRF [7]	69.65	0.029	74.21	22.50	65.88	72.25	0.038	73.40	22.21	67.42
TEXTure [190]	37.05	0.016	86.35	<u>23.38</u>	<u>67.50</u>	48.61	0.021	85.50	<u>24.45</u>	<u>68.23</u>
CLIP-O [191]	<u>25.48</u>	<u>0.013</u>	<u>87.89</u>	21.88	61.96	<u>34.21</u>	**0.016**	<u>87.31</u>	21.401	66.80
CCH	**21.13**	**0.012**	**88.35**	**25.02**	**72.03**	**32.85**	<u>0.016</u>	**87.62**	**27.85**	**76.19**

Table 6.1: The testing results of T3H on DeepFashion and SHHQ.

a pre-trained EVA3D [152] for faster inference.

Implementation Detail We divide a human body into 16 parts and deploy individual StyleSDF [176] for each volume rendering, and two following MLPs then estimate SDF and RGB values. We sample N=28 points for each ray and set (m, n) to (4, 8) for mixture rendering. The text encoder is initialized from CLIP and subsequently trained with CCH. We treat Adam [55] with the learning rates 2e-5 for G and 2e-4 D.

6.4.2 Quantitative Results

Table 6.1 shows the pose-guided T3H results on DeepFashion and SHHQ, where we feed the estimated human mesh as the input object into Latent-NeRF and TEXTure. Though Latent-NeRF and TEXTure can portray body shapes in multiple angles from its latent NeRF space, the rendering is clearly counterfeit (higher FID and Depth). CLIP-O relies on EVA3D to produce feasible 3D humans, but the CLIP alignment is insufficient to demonstrate detailed patterns (lower FA).

A similar trend can be found on SHHQ. Latent-NeRF and TEXTure exhibit related fashion patterns but are hard to present realistic humans. CLIP-O produces a sharp human body with the correct pose, but not the assigned fashion by the inexplicit alignment from CLIP. Without those above drawbacks, our CCH learns to extract fashion seman-

Text	CA	SD	FID↓	CLIP-S↑	FA↑
✗	✗	✗	25.67	9.63	36.63
✓	✗	✗	24.62	21.07	69.17
✓	✓	✗	<u>21.96</u>	<u>24.10</u>	<u>80.02</u>
✓	✓	✓	**21.27**	**25.21**	**80.77**

Table 6.2: The ablation study of CCH on DeepFashion.

Method	Quality	Relevance
L-NeRF [7]	1.82	2.37
TEXTure [190]	2.38	<u>2.51</u>
CLIP-O [191]	**2.93**	2.20
CCH	<u>2.87</u>	**2.92**

Table 6.3: The human comparison with baselines on DeepFashion.

Method	Time (sec)	GPU (MB)
L-NeRF [7]	755.7	<u>11250</u>
TEXTure [190]	<u>103.7</u>	12530
CLIP-O [191]	181.6	15988
CCH	**0.372**	**6258**

Table 6.4: The time and GPU cost when performing `T3H` on TITAN RTX.

tics along with the compositional human generation, leading to comprehensive superiority across all metrics.

6.4.3 Ablation Study

We study each component effect of CCH in Table 6.2. Without the guided description, the model lacks the target fashion and results in a poor FA. When applying the traditional training [1], conditional GAN is insufficient to extract fashion semantics for effective `T3H`. On the other hand, our cross-modal attention constructs a better fusion between fashion patterns and volume rendering. Moreover, semantic discrimination benefits fine-grained alignment and leads to comprehensive advancement.

Human Evaluation Apart from automatic metrics, we conduct the human evaluation (75 samples for each method) with aspects of 3D quality and fashion relevance. Table 6.3 shows the mean ranking score (from 1 to 4, the higher is the better). CLIP-O and CCH are built upon EVA3D, which provides an articulate human body for superior 3D quality.

Figure 6.3: The qualitative comparison between baselines and our CCH.

Even if Latent-NeRF and TEXTure take pre-trained diffusion models to acquire visual guidance, CCH exhibits more corresponding fashion via cross-modal fusion.

Inference Efficiency Table 6.4 shows the inference time and GPU cost on a single TITAN RTX. All baselines take more than 100 seconds as they require multiple iterations to optimize the 3D model from an external alignment. In contrast, we extract fashion semantics and carry out `T3H` in one shot. Without updating the model, we save the most GPU memory.

6.4.4 Qualitative Results

We demonstrate the qualitative comparison in Fig. 6.3. Both Latent-NeRF and TEXTure fail to capture "*three-point*", where the rendered lower clothing is incorrectly depicted as long pants. since CLIP provides an overall but inexplicit alignment to the description, CLIP-O is limited and exhibits vague "*denim*" or "*long-sleeved*". This ob-

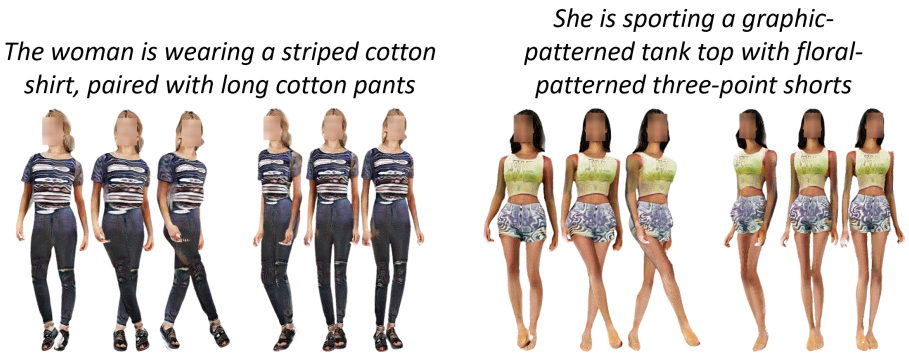


Figure 6.4: The qualitative examples of pose-control T3H.

servation further indicates the flaw of CLIP in detailed fashion patterns. In contrast, our CCH adopts cross-modal attention with NeRF, contributing to high-quality T3H with fine-grained fashion controllability.

Pose-control Human Since CCH is generating 3D humans from the given SMPL parameters, as illustrated in Fig. 6.4, we can control T3H with a specific pose. Different fashion descriptions make a human body present diverse appearances; various poses then guide the character to express rich body language. This flexibility in controlling appearance and pose allows for better practical customization.

Animatable Human In addition to static poses, CCH can benefit from dynamic motions to achieve animatable T3H. Fig. 6.5 adopts MotionDiffuse [192] to create the assigned action also from the text and apply it to our produced 3D models. In this way, we prompt them to "*raise arms*" or "*walk*" for favorable dynamic scenarios.

6.5 Summary

In this chapter, we introduce the text-guided 3D human generation task, aiming to create a 3D human based on a fashion description. To learn this from 2D collections, we

Figure 6.5: The qualitative examples of animatable T3H with text-guided motion.

present compositional cross-modal human, which fuses compositional human rendering with textual semantics via cross-modal attention. Hence we can build a concrete body with the corresponding fashion. Experiments across various fashion attributes support that our framework effectively performs 3D human generation with higher efficiency.

Chapter 7

Guiding Instruction-based Image Editing

7.1 Introduction

Instruction-based image editing [12, 11] improves the controllability and flexibility of image manipulation via natural commands without elaborate descriptions [8, 193] or regional masks [68, 194]. Due to the data scarcity of the input-goal-instruction triplet, InsPix2Pix [30] collects a curated IPr2Pr dataset. InsPix2Pix then applies a pre-trained CLIP text encoder [139] to lead the diffusion model along with the input image. However, instructions are sometimes too brief but ambiguous and insufficient to guide toward the intended goal, where this deficiency limits its effectiveness.

Large language models (LLMs) [195, 196] have shown significant advancement in diverse language tasks. Learning from large-scale corpora, LLMs contain latent visual knowledge and creativity, which can assist various vision-and-language tasks [197]. Upon LLMs, multimodal large language models (MLLMs) can treat images as input naturally and provide visual-aware responses to serve as multimodal assistants [198, 199].

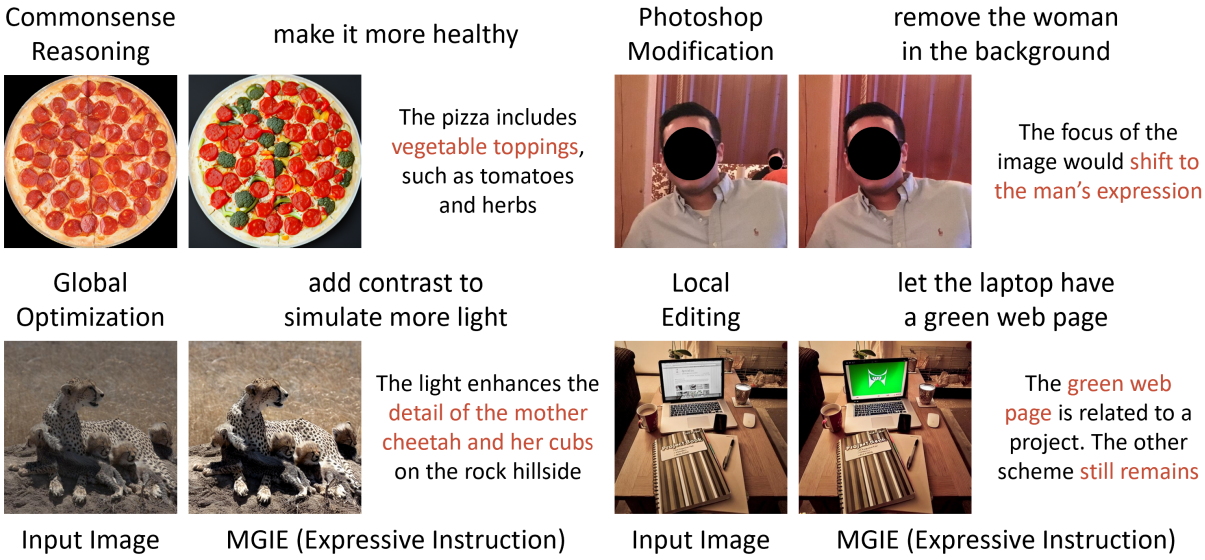


Figure 7.1: We introduce MLLM-guided image editing (MGIE) to improve instruction-based image editing for various editing aspects.

In this chapter, we introduce MLLM-guided image editing (MGIE) to deal with the insufficient guidance issue of instructions. MGIE consists of an MLLM and a diffusion model. The MLLM learns to derive concise expressive instructions and offers explicit visual-related guidance. The diffusion model is jointly trained and performs image editing with the latent imagination. For the example in Fig. 7.1, it is difficult to capture what "*healthy*" means without additional context. Our MGIE can precisely connect "*vegetable toppings*" with the pizza and lead to the related editing as human expectation.

We train MGIE also on IPr2Pr. The evaluation considers different editing aspects in EVR [32], GIER [33], MA5k [34], and MagicBrush [35], such as Photoshop-style modification, global photo optimization, and local object alteration. All should be guided by human instruction. Experimental results indicate that our MGIE significantly strengthens instruction-based image editing with reasonable expressive instructions, where visual-aware guidance is crucial to such improvements.

Dataset	Edit Aspect	Instruction Example
EVR	Photoshop-style	*change the background to blue*
	Modification	*on frozen lake with snowy mountains*
GIER	Photoshop-style	*edit out skiers on right*
	Modification	*lighten out yellow tone*
MA5k	Global Photo	*add contrast to simulate more light*
	Optimization	*need to clarified, more focus*
MagicBrush	Local Object	*put buildings in the background*
	Alteration	*make the face happy*

7.2 Related Work

Instruction-based Image Editing With promising large-scale training, diffusion models [126, 200, 5] can accomplish image transformation via controlling the cross-modal attention maps for the global caption [201, 8, 202]. Local image editing allows fine-grained manipulation by inpainting target regions with user-provided [68, 194, 203] or predicted masks [204, 205] while preserving the remaining areas. Different from them, instruction-based image editing accepts straight commands, such as "*add fireworks to the sky*". Recent methods learn from synthetic input-goal-instruction triples [30] to follow editing instructions. However, the CLIP text encoder is pre-trained for static descriptions but not the crucial transformation in editing.

Large Language Models (LLMs) for Vision With robust text understanding, previous works adapt LLMs for input prompts and reason downstream vision-and-language tasks [206, 197, 207]. Through visual features alignment with instruction tuning, multimodal large language models (MLLMs) can perceive images and provide adequate responses [208, 198, 199]. Recently, studies also adopt MLLMs for generating chat-related images [209, 210]. However, they can only produce images from scratch, which are dis-

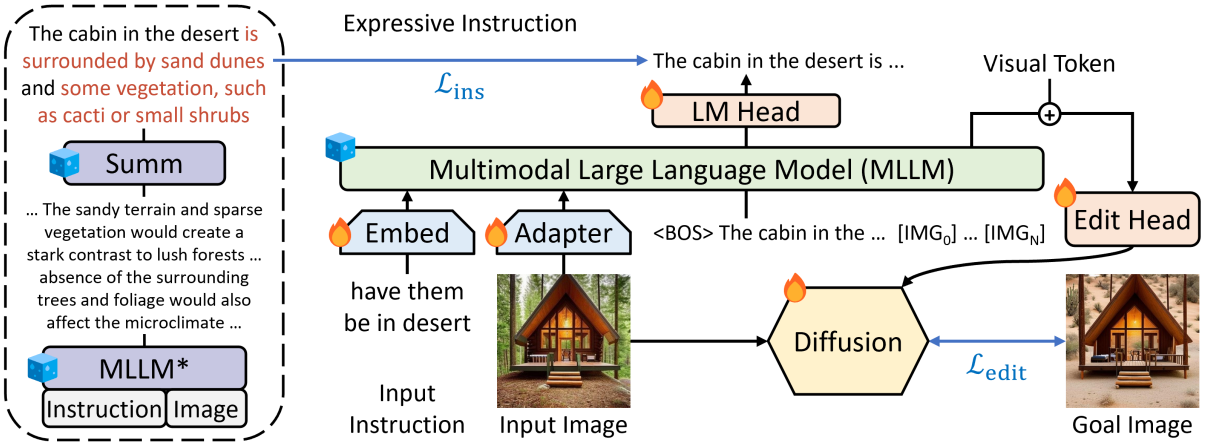


Figure 7.2: The overview of our MLLM-guided image editing (MGIE), which leverages an MLLM to enhance instruction-based image editing. MGIE learns to derive concise expressive instructions and provides explicit visual-related guidance for the intended goal. The diffusion model jointly trains and achieves image editing with the latent imagination through the edit head in an end-to-end manner.

tinct from inputs. Our MGIE is the first to leverage MLLMs and improve image editing with derived expressive instructions.

7.3 MLLM-guided Image Editing (MGIE)

7.3.1 Overview

Background Multimodal large language models (MLLMs) empower LLMs to perceive images and provide reasonable responses. Initialized from a pre-trained LLM, the MLLM contains the visual encoder (*e.g.*, CLIP-L [139]) to extract the visual features f, and an adapter $\mathcal{W}$ to project f into the language modality. We follow the training of LLaVA [198], which is summarized as:

$$\begin{aligned} \mathcal{C} &= \{x_1, x_2, ..., x_l\}, \quad f = \text{Enc}_{\text{vis}}(\mathcal{V}) \\ x_t &= \text{MLLM}(\{x_1, ...x_{t-1}\} \mid \mathcal{W}(f)), \end{aligned} \tag{7.1}$$

where l is the length of the word token in $\mathcal{C}$. $\mathcal{C}$ can be the image caption (Features Alignment) or the multimodal instruction-following data (Instruction Tuning). The MLLM follows the standard autoregressive training for the next token prediction and can serve as a visual assistant for various tasks like visual question answering and complex reasoning.

As illustrated in Fig. 7.2, we propose MLLM-guided image editing (MGIE) to edit an input image $\mathcal{V}$ into a goal image $\mathcal{O}$, by a given instruction $\mathcal{X}$. To handle imprecise instructions, MGIE contains the MLLM and learns to derive explicit yet concise expressive instructions. To bridge the language and visual modality, we add special `[IMG]` tokens and adopt the edit head to transform them. They serve as the latent visual imagination from the MLLM and guide our diffusion model to achieve the intended editing goal.

7.3.2 Concise Expressive Instruction

From features alignment and instruction tuning, the MLLM can offer visual-related responses with its cross-modal perception. For image editing, we use this prompt "*what will this image be like if* `[instruction]` ?" as the language input with the image and derive a detailed explanation of the editing command. However, those explanations are always too lengthy and involve redundant descriptions, which even mislead the intention. To obtain succinct narrations, we apply a pre-trained summarizer and make the MLLM learn to generate the summarized outputs. We treat this explicit yet concise guidance as expressive instruction $\mathcal{E}$:

$$\begin{aligned} \mathcal{E} &= \text{Summ}(\text{MLLM}^*([\texttt{prompt}, \mathcal{X}] \mid \mathcal{W}(f))) \\ &= \{w_1, w_2, ..., w_l\}, \\ w'_t &= \text{MLLM}(\{w_1, ..., w_{t-1}\} \mid \mathcal{W}(f)), \\ \mathcal{L}_{\text{ins}} &= \sum\nolimits_{t=1}^{l} \text{CELoss}(w'_t, w_t), \end{aligned} \tag{7.2}$$

where we apply the cross-entropy loss (CELoss) to train the MLLM via teacher forcing. $\mathcal{E}$ can provide a more concrete idea than $\mathcal{X}$ such as linking "*dessert*" with "*sand dunes*" and "*cacti or small shrubs*", which mitigates the comprehension gap for reasonable image editing. This strategy further helps enhance our efficiency. During inference, the trained MGIE straightforwardly derives concise $\mathcal{E}$ instead of rolling out lengthy narrations (22.7 *vs.* 64.5 tokens).

MGIE now can acquire a visual imagination of the editing intention but is confined to the language modality. To bridge the gap, we append N visual tokens `[IMG]` after $\mathcal{E}$, where their word embeddings are trainable, and the MLLM also learns to generate them through its language modeling (LM) head. Inspired by GILL [209], these visual tokens are treated as visual-related instruction understanding in $\mathcal{E}$ and establish a connection between the language and vision modalities.

7.3.3 Image Editing via Latent Imagination

We adopt the edit head $\mathcal{T}$ to transform `[IMG]` tokens into actual visual guidance. $\mathcal{T}$ is a sequence-to-sequence model, which maps the sequential visual tokens from the MLLM to the semantically meaningful latent $\mathcal{U} = \{u_1, u_2, ..., u_L\}$ as the editing guidance:

$$u_t = \mathcal{T}(\{u_1, ..., u_{t-1}\} \mid \{e_{\texttt{[IMG]}} + h_{\texttt{[IMG]}}\}), \tag{7.3}$$

where e is the word embedding and h is the hidden state (from the last layer of MLLM before the LM head) of `[IMG]`. Specifically, the transformation over e can be treated as a general representation in the visual modality, and h is an instance-aware visual imagination for such editing intention.

To guide image editing with the visual imagination $\mathcal{U}$, we consider a latent diffusion model $\mathcal{F}$ [5], which includes the variational autoencoder (VAE) and addresses denoising

diffusion in the latent space. Our goal of $\mathcal{F}$ is to generate the latent goal $o = \text{Enc}_{\text{VAE}}(\mathcal{O})$ from preserving the latent input $v = \text{Enc}_{\text{VAE}}(\mathcal{V})$ and following the editing guidance $\{u\}$. The diffusion process keeps adding noises to o as z_t, where the noise level is increasing over timesteps t. We then learn the UNet ϵ_θ to predict the added noise [126]. As latent diffusion model (LDM), we inject the visual imagination into ϵ_θ via the cross-attention layer $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{\text{dim}}}) \cdot V$ with

$$Q = W_Q^{(i)} \cdot \varphi_i(z_t), K = W_K^{(i)} \cdot \{u\}, V = W_V^{(i)} \cdot \{u\}, \tag{7.4}$$

where φ is the flattened operation, $W_Q^{(i)}$, $W_K^{(i)}$, and $W_V^{(i)}$ are learnable attention matrices. Following InsPix2Pix, we also concatenate v with z_t. In this way, our $\mathcal{F}$ can condition both $\mathcal{V}$ and $\mathcal{U}$ to perform image editing. We take classifier-free guidance [211], and the score estimation s_θ is extrapolated to keep away from the unconditional $\varnothing$, where the editing loss $\mathcal{L}_{\text{edit}}$ is calculated as:

$$\begin{aligned} s_\theta(z_t, v, \{u\}) &= s_\theta(z_t, \varnothing, \varnothing) \\ &\quad + \alpha_\mathcal{V} \cdot (s_\theta(z_t, v, \varnothing) - s_\theta(z_t, \varnothing, \varnothing)) \\ &\quad + \alpha_\mathcal{X} \cdot (s_\theta(z_t, v, \{u\}) - s_\theta(z_t, v, \varnothing)), \\ \mathcal{L}_{\text{edit}} &= \mathbb{E}_{o,v,\{u\},\epsilon\sim\mathcal{N}(0,1),t} \left[||\epsilon - \epsilon_\theta(z_t, t, v, \{u\})||_2^2 \right], \end{aligned} \tag{7.5}$$

where $\alpha_\mathcal{V}$ and $\alpha_\mathcal{X}$ are the weights of the guidance scale for the image and the instruction. Similar to InsPix2Pix, we randomly make $v = \varnothing$, $\{u\} = \varnothing$, or both $= \varnothing$ for 5% of data during training. After we have the generated latent o' through the denoising process by ϵ_θ, we can obtain the editing result $O' = \text{Dec}_{\text{VAE}}(o')$. During inference, we use $\alpha_\mathcal{V} = 1.5$ and $\alpha_\mathcal{X} = 7.5$.

Algorithm 6 MLLM-guided Image Editing (MGIE)

$\mathcal{T}$: Edit Head
$\mathcal{F}$: Diffusion Model

while TRAIN_MGIE **do**
    $\mathcal{V}$, $\mathcal{X}$, $\mathcal{O}$ $\leftarrow$ sampled input / instruction / goal
    $\{w\}$ $\leftarrow$ summarized explanation
    $\{w'\} = \text{MLLM}(\mathcal{V} \mid \mathcal{X})$
    $\mathcal{U} = \mathcal{T}(\{\texttt{[IMG]}\})$
    $\mathcal{O}' = \mathcal{F}(\mathcal{V},\mathcal{U})$

    $\mathcal{L}_{\text{ins}}$, $\mathcal{L}_{\text{edit}}$ $\leftarrow$ instruction / editing loss ▷ Eq. 7.2 and 7.5
    $\mathcal{L}_{\text{all}}$ $\leftarrow$ overall training loss ▷ Eq. 7.6
    Update MLLM, $\mathcal{T}$, and $\mathcal{F}$ by minimizing $\mathcal{L}_{\text{all}}$
end while

7.3.4 Learning of MGIE

Algo. 6 presents the learning process of the proposed MGIE. The MLLM learns to derive concise $\mathcal{E}$ via the instruction loss $\mathcal{L}_{\text{ins}}$. With the latent imagination from `[IMG]`, $\mathcal{T}$ transforms their modality and guides $\mathcal{F}$ to synthesize the resulting image. The editing loss $\mathcal{L}_{\text{edit}}$ is applied for diffusion training. Most weights can be frozen (the self-attention blocks inside the MLLM), leading to parameter-efficient end-to-end training. The overall optimization can be summarized as:

$$\begin{gathered}\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{ins}} + 0.5 \cdot \mathcal{L}_{\text{edit}}, \\ \min_{\text{MLLM},\mathcal{T},\mathcal{F}} \mathcal{L}_{\text{all}}.\end{gathered} \tag{7.6}$$

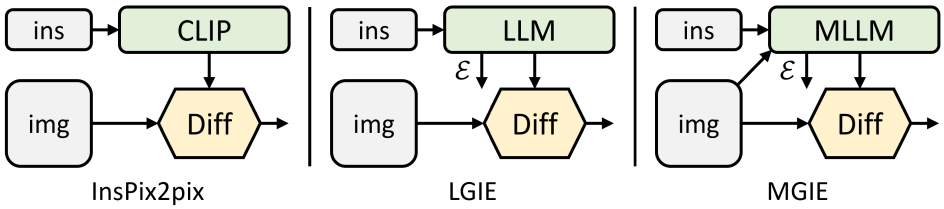
7.4 Experiments

7.4.1 Experimental Setup

Datasets and Evaluation Metrics We use IPr2Pr [30] as our training data, which contains 1M synthesized (by GPT-3 [195] and Pr2Pr [8]) image-instruction pairs. For

a comprehensive evaluation, we consider various editing aspects. EVR [32] crawls 5.7K triples from PhotoshopRequest. We treat the standard pixel difference (L1) and visual features similarity from DINO [212] or the CLIP visual encoder (CVS) between generated images and ground-truth goals as the evaluation metrics. GIER [33] collects a larger-scale 29.9K triples also from online forums. Since there are more examples of global optimization, we apply L1, CVS, and Structural Similarity Index (SSIM). MA5k [34] consists of 24.8K triples and aims at changing the contrast, brightness, or saturation of a whole photo. We leverage L1, SSIM, and Learned Perceptual Image Patch Similarity (LPIPS) [213] as the photo difference. MagicBrush [35] labels 10.5K triples. We follow them to use L1, DINO, CVS, and text-visual features similarity (CTS) [189] between goal captions and resulting images.

Baselines We treat InsPix2Pix [30], built upon the CLIP text encoder with a diffusion model for instruction-based image editing, as our baseline. We consider a similar LLM-guided image editing (LGIE) model, where LLaMA-7B [196] is adopted for expressive instructions $\mathcal{E}$ from instruction-only inputs but without visual perception.



Implementation Detail The MLLM and diffusion model $\mathcal{F}$ are initialized from LLaVA-7B [198] and StableDiffusion-v1.5 [5]. Note that only the word embeddings and the LM head in the MLLM are trainable. We use $N = 8$ visual tokens. The edit head $\mathcal{T}$ is a 4-layer transformer, which transforms language features into editing guidance. We adopt AdamW [214] to optimize MGIE with the learning rate 5e-4 for MLLM and 1e-4 for $\mathcal{F}$.

	EVR			GIER		
Method	L1↓	DINO↑	CVS↑	L1↓	SSIM↑	CVS↑
InsPix2Pix	0.189	67.82	81.38	<u>0.144</u>	<u>57.51</u>	86.634
LGIE	**0.159**	<u>69.71</u>	**82.04**	0.152	56.86	<u>86.99</u>
MGIE	<u>0.163</u>	**71.49**	<u>81.73</u>	**0.135**	**59.24**	**88.59**

	MA5k			MagicBrush			
Methd	L1↓	SSIM↑	LPIPS↓	L1↓	DINO↑	CVS↑	CTS↑
InsPix2Pix	0.176	58.92	0.359	0.101	71.46	85.22	29.34
LGIE	<u>0.144</u>	<u>64.60</u>	<u>0.327</u>	<u>0.084</u>	<u>80.90</u>	<u>88.87</u>	<u>30.10</u>
MGIE	**0.133**	**66.25**	**0.298**	**0.082**	**82.22**	**91.14**	**30.40**

Table 7.1: The zero-shot testing results. All methods are trained only on IPr2Pr [30].

7.4.2 Quantitative Results

Table 7.1 shows the zero-shot results, where all methods are trained only on IPr2Pr. For EVR and GIER that involve Photoshop-style modifications, expressive instructions can reveal concrete goals instead of brief but ambiguous commands, which makes the editing results more similar to intentions (*e.g.*, higher 82.0 CVS on EVR). For global photo optimization on MA5k, InsPix2Pix is hard to deal with due to the scarcity of related training triples. With access to images, MGIE derives explicit instructions such as "*which regions should brighten*" or "*what objects are more distinct"*, which brings a significant performance boost (*e.g.*, lower 0.3 LPIPS). Similar results are found on MagicBrush. MGIE also achieves the best performance from the precise visual imagination and modifies the designated targets as the goals (*e.g.*, 30.4 CTS).

Trade-off between $\alpha_{\mathcal{X}}$ and $\alpha_{\mathcal{V}}$ There are two goals in image editing: manipulate the target as the instruction and preserve the remaining as the input image. Fig. 7.3 plots the trade-off curves between the instruction ($\alpha_{\mathcal{X}}$) and input consistency ($\alpha_{\mathcal{V}}$). We adopt $\alpha_{\mathcal{X}}$ as 7.5 and vary $\alpha_{\mathcal{V}}$ in $[1.0, 2.2]$. X-axis shows the CLIP directional similarity as how

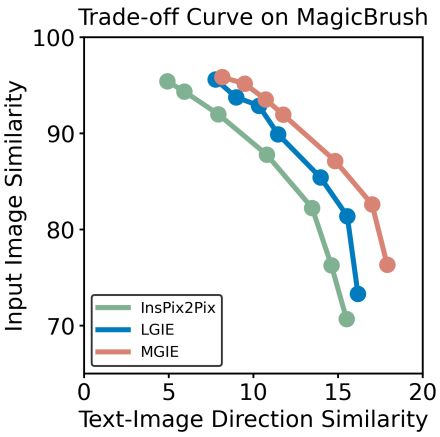


Figure 7.3: The trade-off curve of image editing. We set $\alpha_{\mathcal{X}}$ as 7.5 and vary $\alpha_{\mathcal{V}}$ in $[1.0, 2.2]$.

Arch.	Method	MA5k SSIM↑	MA5k LPIPS↓	MagicBrush DINO↑	MagicBrush CVS↑	MagicBrush CTS↑
InsPix2Pix		**58.92**	**0.359**	71.46	85.22	29.34
FZ	LGIE	57.26	0.372	67.53	82.49	28.79
	MGIE	57.54	0.366	**71.65**	**86.00**	**29.43**
FT	LGIE	60.11	0.357	71.04	85.47	29.37
	MGIE	**61.38**	**0.348**	**74.79**	**87.12**	**29.68**
E2E	LGIE	64.60	0.327	80.90	88.87	30.10
	MGIE	**66.25**	**0.298**	**82.22**	**91.14**	**30.40**

Table 7.2: The ablation study of how to utilize expressive instructions $\mathcal{E}$. FZ directly treats $\mathcal{E}$ as the inputs to frozen InsPix2Pix; FT fine-tunes InsPix2Pix and makes $\mathcal{E}$ adaptive; E2E learns $\mathcal{E}$ along with the MLLM and jointly trains the diffusion model.

much the editing follows the instruction; Y-axis is the features similarity to the input image from the CLIP visual encoder. Our MGIE surpasses all baselines by learning with explicit visual-related guidance. This supports robust improvement, whether requiring higher input correlation or edit relevance.

7.4.3 Ablation Study

We investigate different architectures to utilize expressive instructions $\mathcal{E}$ in Table 7.2. FZ directly uses $\mathcal{E}$ as the input prompts to the frozen InsPix2Pix. However, the scenario still differs from the trained editing instructions, which makes it difficult to deal with. LGIE even hurts the performance as it may mislead due to the shortage of visual perception. FT fine-tunes InsPixPix and adapts it to $\mathcal{E}$. These increased results support that image editing can benefit from explicit guidance along the derivation of instructions. E2E updates the editing diffusion model in conjunction with the LM, which learns to extract applicable guidance and discard irrelevant narration simultaneously. E2E can also avoid the potential error that may be propagated from $\mathcal{E}$, leading to the most improvements.

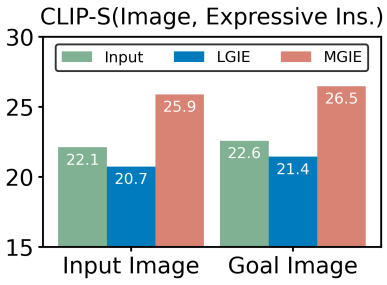


Figure 7.4: The CLIP-Score across images and expressive instructions.

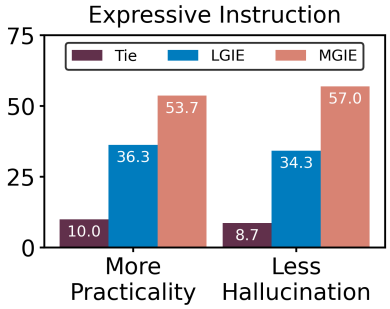


Figure 7.5: The human evaluation of expressive instruction quality.

Why MLLM Guidance is helpful? Fig. 7.4 presents the CLIP-Score between input or ground-truth goal images and $\mathcal{E}$. A higher CLIP-S to input images indicates that instructions are relevant to the editing source. Better alignment with goal images provides explicit and correlated edit guidance. Without access to visual perception, $\mathcal{E}$ from LGIE is limited to general language imagination, which is not tailored to the source image. By contrast, MGIE is more aligned with inputs/goals, which explains why our $\mathcal{E}$ is more helpful and achieves the greatest improvements

Human Evaluation We conduct a human evaluation (100 samples for each method) to study generated $\mathcal{E}$ and image editing results. Fig. 7.5 plots the quality of generated $\mathcal{E}$. Precise guidance is informative and aligns with the intended goal (More Practicality); at the same time, it should avoid incorrect or unrelated explanations (Less Hallucination). Firstly, over 53% support that MGIE provides more practical $\mathcal{E}$. Meanwhile, 57% indicate that our MGIE can prevent irrelevant descriptions from language-derived hallucinations in LGIE since it perceives the image to have a precise goal for editing. Fig. 7.6 compares the image editing results, in terms of instruction following, ground-truth relevance, and overall quality. The ranking score ranges from 1 to 3, the higher is better. With derived $\mathcal{E}$, LGIE and MGIE both outperform the baseline. Additionally, since we can provide

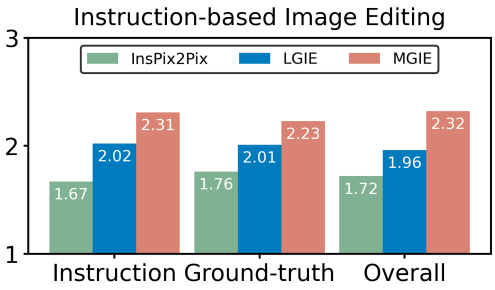


Figure 7.6: The human evaluation of image editing results from instruction following, ground-truth relevance, and overall quality.

concrete and visual-aware guidance, MGIE has the best human preference in all aspects.

7.4.4 Qualitative Results

Fig. 7.7 demonstrates the qualitative comparison between baselines and MGIE on all EVR, GIER, MA5k, and MagicBrush datasets. The details are discussed in the caption.

7.5 Summary

In this chapter, we investigate the guidance gap issue for instruction-based image editing. We leverage multimodal large language models to derive expressive instructions, which provides more explicit editing guidance. The diffusion model then learns to edit the input image via this latent imagination. We conduct extensive studies from various editing aspects and demonstrate that our framework can significantly improve performance yet maintain a competitive efficiency.

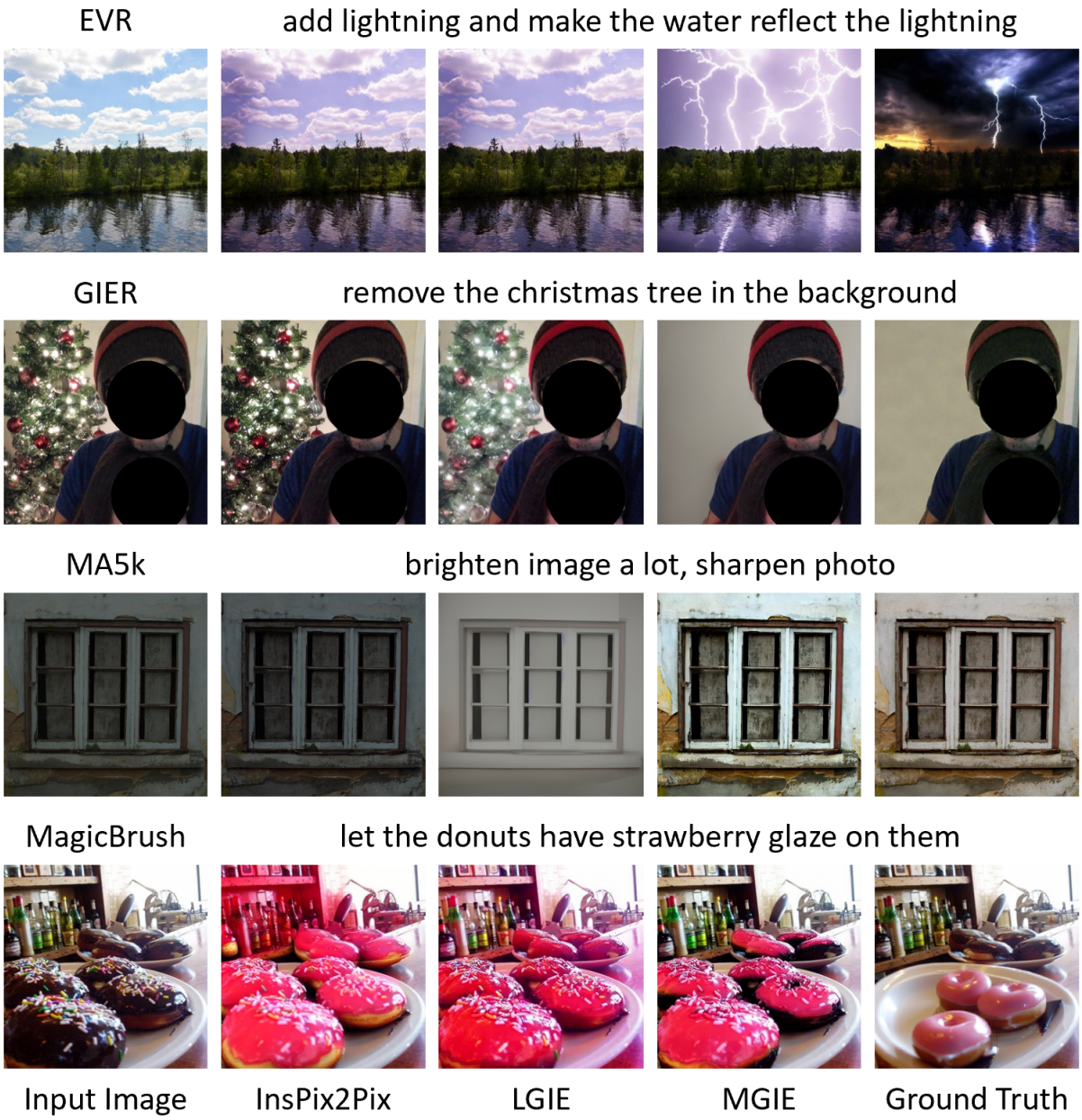


Figure 7.7: The qualitative comparison between baselines and our MGIE. (*i*) MGIE can depict the clear "*lightning*" in the sky and its reflection on the water; (*ii*) Although LGIE accurately targets the Christmas tree, only MGIE removes it in the background; (*iii*) InsPix2Pix fails to adjust the brightness, and LGIE makes the whole photo white and obviously distinct. In contrast, MGIE follows the instruction to brighten as well as sharpen it; (*iv*) MGIE puts the "*glaze*" only on the donuts, but baselines even draw the entire image in strawberry pink.

Chapter 8

Conclusion and Future Work

8.1 Summary

This Ph.D. dissertation has covered various topics and techniques to achieve controllable visual editing via natural language, including images, videos, and natural scenarios.

Chapter 1 introduced the high demand for visual editing tools but also exposed their drawbacks of prior knowledge and complicated operations. These prerequisites limit their accessibility and deter new users from moving forward. In contrast, language is the most straightforward form of communication. If a system can follow human instructions and perform visual editing automatically, it will significantly improve its controllability and bring a vast application impact.

In Part I, we started with images. To manipulate an input image, the model has to understand language as object properties and spatial relations, and generate all pixels of the resulting image. However, collecting numerous image pairs with instructions is still challenging. Chapter 2 imitated human step-by-step modification and incorporated counterfactual thinking to cope with the data scarcity issue. We proposed self-supervised counterfactual reasoning (SSCR), which allows us to consider unseen text-image pairs in

a self-supervised scenario. The experiments on iterative image editing support that our SSCR can improve generalizability without additional training data. Chapter 3 studied how to utilize textual descriptions to lead artistic style transfer. We presented contrastive language visual artist (CLVA) that employs patch-wise discrimination and contrastive reasoning to jointly align text with style patterns. We demonstrated our CLVA's ability to express visual attributes and emotional effects while maintaining high efficiency for text-guided style transfer.

Unlike images, dynamic videos in Part II are even more difficult to modify. We should change only the semantics but preserve the scenario, such as the same scene or human. Chapter 4 considered the hierarchical conveyance between instructions and videos. The multimodal multi-level transformer (M^3L) fuses video perception and text comprehension at multiple levels. Experimental results showed that our M^3L can achieve content replacement and motion manipulation on both diagnostic and natural benchmarks. We investigated the video completion task in Chapter 5. Apart from the initial frames, our multimodal mask video generation (MMVG) learns to generate the full video from visual guidance at arbitrary time points, guided by language. By varying masking conditions, MMVG unified all predictions, rewinds, and infilling across diverse video scenarios. This learning from completion also benefited general video generation.

Built upon images and videos, we then endeavored towards two specific tasks for natural visual manipulation in Part III. Chapter 6 brought 3D human generation. We proposed compositional cross-modal human (CCH) to learn this from 2D image collections. CCH integrates visual rendering with fashion descriptions to depict concrete 3D characters. Experiments across various fashion attributes highlighted the effectiveness as well as the efficiency of our CCH. Chapter 7 unveiled the gap of insufficient guidance in instruction-based editing. With latent visual knowledge from large language models, we presented MLLM-guided image editing (MGIE). MGIE learns to derive expressive in-

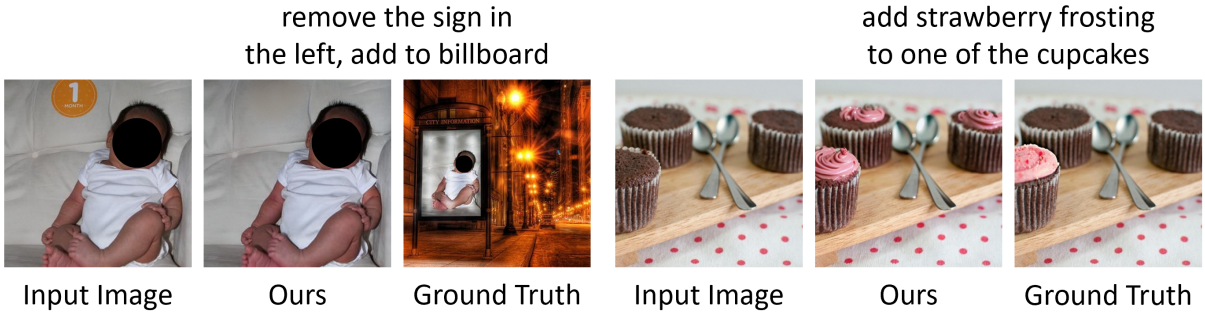


Figure 8.1: The qualitative examples of compositional instructions (left) and numerical grounding (right).

structions and jointly carry out the editing task. We conducted evaluations from various editing aspects, and MGIE all contributed to obvious performance boosts.

Overall, the outline of this dissertation aligns with the prevailing trend observed in the research community: making realistic visual editing controllable yet more accessible for practical usage. We envision that our approaches will inspire further study of generative AI and be adopted for real-world applications.

8.2 Future Directions

My ultimate goal is to build a generalist AI artist that can comprehend human instruction at any granularity and manipulate visual content with its creativity. To enhance the practical utility of the model, several challenges need to be tackled at this moment. With the end goal in mind, I identify the following research avenues to explore further.

Compositional Instructions We have demonstrated that with visual-aware guidance from the MLLM, the editing model can modify images according to provided instructions. However, when the text contains multiple simultaneous goals, the model may struggle with this compositional instruction. Fig. 8.1 requires removing the sign as well as putting the entire photo into a billboard. However, we only delete the sign but not the

subsequent action successfully. How to break down complex compositional instructions and accomplish all of them is an important top for future research.

Numerical Grounding The fine-grained perception of visual content using language is still not robust. As shown in Fig. 8.1, we add frosting to all cupcakes instead of just one. This counting issue highlights the necessity of numerical-aware grounding for more accurate editing targets.

Guidance from Multiple Modalities Besides language, there are still different modalities in our daily lives that can interact with visual content. For example, sound conveys the vibe and is a vital element of videos. Referring, such as points, boxes, and free-form drawings can express precise regions for editing goals. Leveraging multimodal guidance can mitigate the potential ambiguity and create a barrier-free human-AI interaction.

Efficient Inference Current visual generative methods are built upon large models, which demand server-level computing resources (*e.g.*, >10GB GPU RAM). For instance, the popular StableDiffusion [5] performs 50-100 denoising steps, which takes around 10 seconds. However, this requirement is costly for mobile phones or even personal PCs. Users may not be willing to wait for such a long time in front of their screens. To make it affordable and eliminate device restrictions, we should establish approaches for more efficient (in terms of both time and memory) inference.

Bibliography

[1] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, *Generative Adversarial Text to Image Synthesis*, in *International Conference on Machine Learning (ICML)*, 2016.

[2] Y. Li, M. R. Min, D. Shen, D. Carlson, and L. Carin, *Video Generation from Text*, in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[3] S. Shinagawa, K. Yoshino, S. Sakti, Y. Suzuki, and S. Nakamura, *Interactive Image Manipulation with Natural Language Instruction Commands*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[4] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu, *Language-Based Image Editing with Recurrent Attentive Models*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-Resolution Image Synthesis with Latent Diffusion Models*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman, *Make-A-Video: Text-to-Video Generation without Text-Video Data*, in *arXiv:2209.14792*, 2022.

[7] G. Metzer, E. Richardson, O. Patashnik, R. Giryes, and D. Cohen-Or, *Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[8] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, *Prompt-to-Prompt Image Editing with Cross Attention Control*, in *International Conference for Learning Representations (ICLR)*, 2023.

[9] J. Z. Wu, Y. Ge, X. Wang, W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou, *Tune-A-Video: One-Shot Tuning of Image Diffusion Models for Text-to-Video Generation*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[10] N. J. Roese, *Counterfactual Thinking*, in *Psychological Bulletin*, 1997.

[11] T.-J. Fu, X. E. Wang, S. Grafton, M. Eckstein, and W. Y. Wang, *SSCR: Iterative Language-Based Image Editing via Self-Supervised Counterfactual Reasoning*, in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[12] A. El-Nouby, S. Sharma, H. Schulz, D. Hjelm, L. E. Asri, S. E. Kahou, Y. Bengio, and G. W.Taylor, *Tell, Draw, and Repeat: Generating and Modifying Images Based on Continual Linguistic Instruction*, in *International Conference on Computer Vision (ICCV)*, 2019.

[13] J.-H. Kim, N. Kitaev, X. Chen, M. Rohrbach, B.-T. Zhang, Y. Tian, D. Batra, and D. Parikh, *CoDraw: Collaborative Drawing as a Testbed for Grounded Goal-driven Communication*, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[14] T.-J. Fu, X. E. Wang, and W. Y. Wang, *Language-Driven Artistic Style Transfer*, in *European Conference on Computer Vision (ECCV)*, 2022.

[15] C. Wu, M. Timm, and S. Maji, *Describing Textures using Natural Language*, in *European Conference on Computer Vision (ECCV)*, 2020.

[16] P. Achlioptas, M. Ovsjanikov, K. Haydarov, M. Elhoseiny, and L. Guibas, *ArtEmis: Affective Language for Visual Art*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[17] T.-J. Fu, X. E. Wang, S. Grafton, M. Eckstein, and W. Y. Wang, *M^3L: Language-based Video Editing via Multi-Modal Multi-Level Transformer*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] N. Srivastava, E. Mansimov, and R. Salakhutdinov, *Unsupervised Learning of Video Representations using LSTMs*, in *International Conference on Machine Learning (ICML)*, 2015.

[19] J. Johnson, B. Hariharan, L. van der Maaten, F.-F. Li, L. Zitnick, and R. Girshick, *CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, *The Jester Dataset: A Large-Scale Video Dataset of Human Gestures*, in *International Conference on Computer Vision (ICCV)*, 2019.

[21] Z. Hao, X. Huang, and S. Belongie, *Controllable Video Generation with Sparse Trajectories*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] J.-T. Hsieh, B. Liu, D.-A. Huang, L. Fei-Fei, and J. C. Niebles, *Learning to Decompose and Disentangle Representations for Video Prediction*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[23] T.-J. Fu, L. Yu, N. Zhang, C.-Y. Fu, J.-C. Su, W. Y. Wang, and S. Bell, *Tell Me What Happened: Unifying Text-guided Video Completion via Multimodal Masked Video Generation*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[24] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, *Scaling Egocentric Vision: The EPIC-KITCHENS Dataset*, in *European Conference on Computer Vision (ECCV)*, 2018.

[25] T. Gupta, D. Schwenk, A. Farhadi, D. Hoiem, and A. Kembhavi, *Imagine This! Scripts to Compositions to Videos*, in *European Conference on Computer Vision (ECCV)*, 2018.

[26] T. Hayes, S. Zhang, X. Yin, G. Pang, S. Sheng, H. Yang, S. Ge, Q. Hu, and D. Parikh, *MUGEN: A Playground for Video-Audio-Text Multimodal Understanding and GENeration*, in *European Conference on Computer Vision (ECCV)*, 2022.

[27] T.-J. Fu, W. Xiong, Y. Nie, J. Liu, B. Oğuz, and W. Y. Wang, *Text-guided 3D Human Generation from 2D Collections*, in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[28] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, *DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] J. Fu, S. Li, Y. Jiang, K.-Y. Lin, C. Qian, C. C. Loy, W. Wu, and Z. Liu, *StyleGAN-Human: A Data-Centric Odyssey of Human Generation*, in *European Conference on Computer Vision (ECCV)*, 2022.

[30] T. Brooks, A. Holynski, and A. A. Efros, *InstructPix2Pix: Learning to Follow Image Editing Instructions*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[31] T.-J. Fu, W. Hu, X. Du, W. Y. Wang, Y. Yang, and Z. Gan, *Guiding Instruction-based Image Editing via Multimodal Large Language Models*, in *International Conference on Learning Representations (ICLR)*, 2024.

[32] H. Tan, F. Dernoncourt, Z. Lin, T. Bui, and M. Bansal, *Expressing Visual Relationships via Language*, in *Annual Meetings of the Association for Computational Linguistics (ACL)*, 2019.

[33] J. Shi, N. Xu, T. Bui, F. Dernoncourt, Z. Wen, and C. Xu, *A Benchmark and Baseline for Language-Driven Image Editing*, in *Asian Conference on Computer Vision (ACCV)*, 2020.

[34] J. Shi, N. Xu, Y. Xu, T. Bui, F. Dernoncourt, and C. Xu, *Learning by Planning: Language-Guided Global Image Editing*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[35] K. Zhang, L. Mo, W. Chen, H. Sun, and Y. Su, *MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing*, in *arXiv:2306.10012*, 2023.

[36] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, *Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] S. Reed, A. van den Oord, N. Kalchbrenner, S. G. Colmenarejo, Z. Wang, D. Belov, and N. de Freitas, *Parallel Multiscale Autoregressive Density Estimation*, in *International Conference on Machine Learning (ICML)*, 2017.

[38] F. Tan, S. Feng, and V. Ordonez, *Text2Scene: Generating Compositional Scenes from Textual Descriptions*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Networks*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[40] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, *StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networkss*, in *International Conference on Computer Vision (ICCV)*, 2017.

[41] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. Hes, *AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[42] G. Laput, M. Dontcheva, G. Wilensky, W. Chang, A. Agarwala, J. Linder, and E. Adar, *PixelTone: A Multimodal Interface for Image Editing*, in *Conference on Human Factors in Computing Systems (CHI)*, 2013.

[43] M.-M. Cheng, S. Zheng, W.-Y. Lin, J. Warrell, V. Vineet, P. Sturgess, N. Crook, N. Mitra, and P. Torr, *ImageSpirit: Verbal Guided Image Parsing*, in *ACM Transactions on Graphics (ToG)*, 2013.

[44] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, *Counterfactual Fairness*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[45] S. Garg, V. Perot, N. Limtiaco, A. Taly, E. H. Chi, and A. Beutel, *Counterfactual Fairness in Text Classification through Robustness*, in *Conference on AI, Ethics, and Society (AIES)*, 2019.

[46] R. Zmigrod, S. J. Mielke, H. Wallach, and R. Cotterell, *Counterfactual Data Augmentation for Mitigating Gender Stereotypes in Languages with Rich Morphology*, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[47] T.-J. Fu, X. Wang, M. Peterson, S. Grafton, M. Eckstein, and W. Y. Wang, *Counterfactual Vision-and-Language Navigation via Adversarial Path Sampling*, in *European Conference on Computer Vision (ECCV)*, 2020.

[48] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[49] T. Miyato and M. Koyama, *cGANs with Projection Discriminator*, in *International Conference on Learning Representations (ICLR)*, 2018.

[50] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*, in *International Conference on Machine Learning (ICML)*, 2015.

[51] R. J. Williams and D. Zipser, *A Learning Algorithm for Continually Running Fully Recurrent Neural Networks*, in *Neural Computation*, 1989.

[52] S. Bird and E. Loper, *NLTK: The Natural Language Toolkit*, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

[53] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the Inception Architecture for Computer Vision*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[54] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, *Self-Attention Generative Adversarial Networks*, in *Proceedings of Machine Learning Research (PMLR)*, 2019.

[55] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, in *International Conference on Learning Representations (ICLR)*, 2015.

[56] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, *BLEU: a Method for Automatic Evaluation of Machine Translation*, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[57] L. A. Gatys, A. S. Ecker, and M. Bethge, *A Neural Algorithm of Artistic Style*, in *arXiv:1508.06576*, 2015.

[58] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, *Universal Style Transfer via Feature Transforms*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[59] X. Huang and S. Belongie, *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization*, in *International Conference on Computer Vision (ICCV)*, 2017.

[60] D. Y. Park and K. H. Lee, *Arbitrary Style Transfer with Style-Attentional Networks*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[61] X. Li, S. Liu, J. Kautz, and M.-H. Yang, *Learning Linear Transformations for Fast Arbitrary Style Transfer*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[62] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, *Neural Style Transfer: A Review*, in *arXiv:1705.04058*, 2017.

[63] F. Luan, S. Paris, E. Shechtman, and K. Bala, *Deep Photo Style Transfer*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[64] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, *A Closed-form Solution to Photorealistic Image Stylization*, in *European Conference on Computer Vision (ECCV)*, 2018.

[65] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. A. Efros, and R. Zhang, *Swapping Autoencoder for Deep Image Manipulation*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[66] L. Shi, K. Shuang, S. Geng, P. Su, Z. Jiang, P. Gao, Z. Fu, G. de Melo, and S. Su, *Contrastive Visual-Linguistic Pretraining*, in *arXiv:2007.13135*, 2020.

[67] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, *Zero-Shot Text-to-Image Generation*, in *arXiv:2102.12092*, 2021.

[68] 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*, in *International Conference on Machine Learning (ICML)*, 2022.

[69] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, *StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery*, in *International Conference on Computer Vision (ICCV)*, 2021.

[70] R. Gal, O. Patashnik, H. Maron, G. Chechik, and D. Cohen-Or, *StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators*, in *arXiv:2108.00946*, 2021.

[71] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, *Analyzing and Improving the Image Quality of StyleGAN*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[72] G. Kwon and J. C. Ye, *CLIPstyler: Image Style Transfer with a Single Text Condition*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[73] L. A. Gatys, A. S. Ecker, and M. Bethge, *Texture Synthesis Using Convolutional Neural Networks*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[74] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays, *TextureGAN: Controlling Deep Image Synthesis with Texture Patches*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[75] Y. Xia, D. He, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma, *Dual Learning for Machine Translation*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

[76] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, in *International Conference on Computer Vision (ICCV)*, 2017.

[77] Z. Yi, H. Zhang, P. Tan, and M. Gong, *DualGAN: Unsupervised Dual Learning for Image-to-Image Translation*, in *International Conference on Computer Vision (ICCV)*, 2017.

[78] S. Liu, T. Lin, D. He, F. Li, M. Wang, X. Li, Z. Sun, Q. Li, and E. Ding, *AdaAttN: Revisit Attention Mechanism in Arbitrary Neural Style Transfer*, in *International Conference on Computer Vision (ICCV)*, 2021.

[79] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncel, *Image Quality Assessment: From Error Visibility to Structural Similarity*, in *Transactions on Image Processing (TIP)*, 2004.

[80] J. Johnson, A. Alahi, and L. Fei-Fei, *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*, in *European Conference on Computer Vision (ECCV)*, 2016.

[81] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[82] C. Wu, L. Huang, Q. Zhang, B. Li, L. Ji, F. Yang, G. Sapiro, and N. Duan, *GODIVA: Generating Open-DomaIn Videos from nAtural Descriptions*, in *arXiv:2104.14806*, 2021.

[83] B. Li, X. Qi, T. Lukasiewicz, and P. H. S. Torr, *ManiGAN: Text-Guided Image Manipulation*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[84] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, in *International Conference on Learning Representations (ICLR)*, 2015.

[85] 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*, in *arXiv:1907.11692*, 2019.

[86] Y. Pan, Z. Qiu, T. Yao, H. Li, and T. Mei, *To Create What You Tell: Generating Videos from Captions*, in *ACM Multimedia (ACMMM)*, 2017.

[87] T. Marwah, G. Mittal, and V. N. Balasubramanian, *Attentive Semantic Video Generation using Captions*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[88] Y. LeCun, C. Cortes, and C. Burges, *MNIST Handwritten Digit Database*, 2010.

[89] R. Villegas, J. Y. amd Seunghoon Hong, X. Lin, and H. Lee, *Decomposing Motion and Content for Natural Video Sequence Prediction*, in *International Conference on Learning Representations (ICLR)*, 2017.

[90] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, *Video Pixel Networks*, in *International Conference on Machine Learning (ICML)*, 2017.

[91] M. Saito, E. Matsumoto, and S. Saito, *Temporal Generative Adversarial Nets with Singular Value Clipping*, in *International Conference on Computer Vision (ICCV)*, 2017.

[92] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, *MoCoGAN: Decomposing Motion and Content for Video Generation*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[93] A. Clark, J. Donahue, and K. Simonyan, *Adversarial Video Generation on Complex Datasets*, in *arXiv:1907.06571*, 2019.

[94] D. L. abd Zhaowen Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, and T. Huang, *Robust Video Super-Resolution with Learned Temporal Dynamics*, in *International Conference on Computer Vision (ICCV)*, 2017.

[95] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, *Deep Video Super-Resolution Network Using Dynamic Upsampling Filters Without Explicit Motion Compensation*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[96] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, *Video-to-Video Synthesis*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[97] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, *Few-shot Video-to-Video Synthesis*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[98] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, *Coherent Online Video Style Transfer*, in *International Conference on Computer Vision (ICCV)*, 2017.

[99] Y. Deng, F. Tang, W. Dong, H. Huang, C. Ma, and C. Xu, *Arbitrary Video Style Transfer via Multi-Channel Correlation*, in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.

[100] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, *Deep Video Inpainting*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[101] R. Xu, X. Li, B. Zhou, and C. C. Loy, *Deep Flow-Guided Video Inpainting*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[102] F. A. Reda, G. Liu, K. J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro, *SDC-Net: Video Prediction using Spatially-Displaced Convolution*, in *European Conference on Computer Vision (ECCV)*, 2018.

[103] X. Li, T.-K. L. Wong, R. T. Q. Chen, and D. Duvenaud, *Scalable Gradients for Stochastic Differential Equations*, in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[104] V. L. Guen and N. Thome, *Disentangling Physical Dynamics from Unknown Factors for Unsupervised Video Prediction*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[105] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, *Attention is All you Need*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[106] C. Liu, Z. Lin, X. Shen, J. Yang, X. Lu, and A. L. Yuille, *Recurrent Multimodal Interaction for Referring Image Segmentation*, in *International Conference on Computer Vision (ICCV)*, 2017.

[107] L. Ye, M. Rochan, Z. Liu, and Y. Wang, *Cross-Modal Self-Attention Network for Referring Image Segmentation*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[108] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov, *Gated-Attention Architectures for Task-Oriented Language Grounding*, in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[109] R. Girdhar and D. Ramanan, *CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning*, in *International Conference on Learning Representations (ICLR)*, 2020.

[110] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, *Aggregated Residual Transformations for Deep Neural Networks*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[111] J. Carreira and A. Zisserman, *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[112] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-Image Translation with Conditional Adversarial Nets*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[113] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, *Eidetic 3D LSTM: A Model for Video Prediction and Beyond*, in *International Conference on Learning Representations (ICLR)*, 2019.

[114] M. Mathieu, C. Couprie, and Y. LeCun, *Deep Multi-Scale Video Prediction Beyond Mean Square Error*, in *International Conference on Learning Representations (ICLR)*, 2018.

[115] E. Denton and R. Fergus, *Stochastic Video Generation with a Learned Prior*, in *International Conference on Machine Learning (ICML)*, 2018.

[116] D. Weissenborn, O. Tackstrom, and J. Uszkoreit, *Scaling Autoregressive Video Models*, in *International Conference on Learning Representations (ICLR)*, 2020.

[117] Q. Hu, A. Waelchli, T. Portenier, M. Zwicker, and P. Favaro, *Video Synthesis from a Single Image and Motion Stroke*, in *arXiv:1812.01874*, 2018.

[118] J. Zhang, C. Xu, L. Liu, M. Wang, X. Wu, Y. liu, and Y. Jiang, *DTVNet: Dynamic Time-lapse Video Generation via Single Still Image*, in *European Conference on Computer Vision (ECCV)*, 2020.

[119] T. Höppe, A. Mehrjou, S. Bauer, D. Nielsen, and A. Dittadi, *Diffusion Models for Video Prediction and Infilling*, in *Transactions on Machine Learning Research (TMLR)*, 2022.

[120] V. Voleti, A. Jolicoeur-Martineau, and C. Pal, *MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[121] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, *Neural Discrete Representation Learning*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[122] P. Esser, R. Rombach, and B. Ommer, *Taming Transformers for High-Resolution Image Synthesis*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[123] D. Acharya, Z. Huang, D. P. Paudel, and L. V. Gool, *Towards High Resolution Video Generation with Progressive Growing of Sliced Wasserstein GANs*, in *arXiv:1810.02419*, 2018.

[124] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, *VideoGPT: Video Generation using VQ-VAE and Transformers*, in *arXiv:2104.10157*, 2021.

[125] S. Ge, T. Hayes, H. Yang, X. Yin, G. Pang, D. Jacobs, J.-B. Huang, and D. Parikh, *Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer*, in *European Conference on Computer Vision (ECCV)*, 2022.

[126] J. Ho, A. Jain, and P. Abbeel, *Denoising Diffusion Probabilistic Models*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[127] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, *Video Diffusion Models*, in *arXiv:2204.03458*, 2022.

[128] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev, *Latent Video Transformer*, in *arXiv:2006.10704*, 2020.

[129] M. Babaeizadeh, M. T. Saffar, S. Nair, S. Levine, C. Finn, and D. Erhan, *FitVid: Overfitting in Pixel-Level Video Prediction*, in *arXiv:2106.13195*, 2021.

[130] A. Gupta, S. Tian, Y. Zhang, J. Wu, R. Martín-Martín, and L. Fei-Fei, *MaskViT: Masked Visual Pre-Training for Video Prediction*, in *arXiv:2206.11894*, 2022.

[131] Q. Xu, H. Zhang, W. Wang, P. N. Belhumeur, and U. Neumann, *Stochastic Dynamics for Video Infilling*, in *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[132] 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*, in *arXiv:2111.02114*, 2021.

[133] X. Wang, J. Wu, J. Chen, L. Li, Y.-F. Wang, and W. Y. Wang, *VATEX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research*, in *International Conference on Computer Vision (ICCV)*, 2019.

[134] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, *Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval*, in *International Conference on Computer Vision (ICCV)*, 2021.

[135] C. Wu, J. Liang, L. Ji, F. Yang, Y. Fang, D. Jiang, and N. Duan, *NÜWA: Visual Synthesis Pre-training for Neural visUal World creAtion*, in *European Conference on Computer Vision (ECCV)*, 2022.

[136] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang, *CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers*, in *arXiv:2205.15868*, 2022.

[137] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, and T. Salimans, *Imagen Video: High Definition Video Generation with Diffusion Models*, in *arXiv:2210.02303*, 2022.

[138] Y. Xu, B. AlBahar, and J.-B. Huang, *Temporally Consistent Semantic Video Editing*, in *European Conference on Computer Vision (ECCV)*, 2022.

[139] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, *Learning Transferable Visual Models From Natural Language Supervision*, in *International Conference on Machine Learning (ICML)*, 2021.

[140] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, *Quantifying Generalization in Reinforcement Learning*, in *International Conference on Machine Learning (ICML)*, 2019.

[141] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, and S. G. Marcin Michalski, *Towards Accurate Generative Models of Video: A New Metric & Challenges*, in *International Conference on Learning Representations (ICLR)*, 2019.

[142] Z. Tong, Y. Song, J. Wang, and L. Wang, *VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[143] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, *Mixed Precision Training*, in *International Conference for Learning Representations (ICLR)*, 2018.

[144] K. Soomro, A. R. Zamir, and M. Shah, *UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild*, in *arXiv:1212.0402*, 2012.

[145] F. Ebert, C. Finn, A. X. Lee, and S. Levine, *Self-Supervised Visual Planning with Temporal Skip Connections*, in *Conference on Robot Learning (CoRL)*, 2017.

[146] J. Xu, T. Mei, T. Yao, and Y. Rui, *MSR-VTT: A Large Video Description Dataset for Bridging Video and Language*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[147] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, *Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop*, in *International Conference on Computer Vision (ICCV)*, 2019.

[148] X. Gao, J. Yang, J. Kim, S. Peng, Z. Liu, and X. Tong, *MPS-NeRF: Generalizable 3D Human Rendering from Multiview Images*, in *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

[149] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, *Zero-Shot Text-Guided Object Generation with Dream Fields*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[150] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, *DreamFusion: Text-to-3D using 2D Diffusion*, in *International Conference on Learning Representations (ICLR)*, 2023.

[151] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*, in *European Conference on Computer Vision (ECCV)*, 2020.

[152] F. Hong, Z. Chen, Y. Lan, L. Pan, and Z. Liu, *EVA3D: Compositional 3D Human Generation from 2D Image Collections*, in *International Conference on Learning Representations (ICLR)*, 2023.

[153] Y. Jiang, S. Yang, H. Qiu, W. Wu, C. C. Loy, and Z. Liu, *Text2Human: Text-Driven Controllable Human Image Generation*, in *Transactions on Graphics (TOG)*, 2022.

[154] L. Gao, J. Yang, T. Wu, Y.-J. Yuan, H. Fu, Y.-K. Lai, and H. Zhang, *SDM-NET: Deep Generative Network for Structured Deformable Mesh*, in *Special Interest Group on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia)*, 2019.

[155] P. Henderson, V. Tsiminaki, and C. H. Lampert, *Leveraging 2D Data to Learn Textured 3D Mesh Generation*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[156] M. Tatarchenko, A. Dosovitskiy, and T. Brox, *Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs*, in *International Conference on Computer Vision (ICCV)*, 2017.

[157] J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas, *GRASS: Generative Recursive Autoencoders for Shape Structures*, in *Transactions on Graphics (TOG)*, 2017.

[158] C.-L. Li, M. Zaheer, Y. Zhang, B. Poczos, and R. Salakhutdinov, *Point Cloud GAN*, in *International Conference on Learning Representations (ICLR)*, 2018.

[159] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, *PointFlow: 3D Point Cloud Generation with Continuous Normalizing Flows*, in *International Conference on Computer Vision (ICCV)*, 2019.

[160] Z. Chen and H. Zhang, *Learning Implicit Fields for Generative Shape Modeling*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[161] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, *DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[162] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, *Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[163] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, *CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[164] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan, *High-quality Streamable Free-viewpoint Video*, in *Transactions on Graphics (TOG)*, 2015.

[165] K. Guo, P. Lincoln, P. Davidson, J. Busch, X. Yu, M. Whalen, G. Harvey, S. Orts-Escolano, R. Pandey, J. Dourgarian, M. DuVall, D. Tang, A. Tkach, A. Kowdle, E. Cooper, M. Dou, S. Fanellov, G. Fyffe, C. Rhemannv, J. Taylor, P. Debevec, and S. Izadi, *The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting*, in *Transactions on Graphics (TOG)*, 2019.

[166] X. Xu and C. C. Loy, *3D Human Texture Estimation from a Single Image with Transformers*, in *International Conference on Computer Vision (ICCV)*, 2021.

[167] T. L. Gomes, T. M. Coutinho, R. Azevedo, R. Martins, and E. R. Nascimento, *Creating and Reenacting Controllable 3D Humans with Differentiable Rendering*, in *Winter Conference on Applications of Computer Vision (WACV)*, 2022.

[168] A. Shysheya, E. Zakharov, K.-A. Aliev, R. Bashirov, E. Burkov, K. Iskakov, A. Ivakhnenko, Y. Malkov, I. Pasechnik, D. Ulyanov, A. Vakhitov, and V. Lempitsky, *Textured Neural Avatars*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[169] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou, *Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[170] S.-Y. Su, F. Yu, M. Zollhoefer, and H. Rhodin, *A-NeRF: Articulated Neural Radiance Fields for Learning Human Shape, Appearance, and Pose*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[171] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman, *HumanNeRF: Free-viewpoint Rendering of Moving People from Monocular Video*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[172] M. Chen, J. Zhang, X. Xu, L. Liu, Y. Cai, J. Feng, and S. Yan, *Geometry-Guided Progressive NeRF for Generalizable and Efficient Neural Human Rendering*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[173] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. D. Mello, O. Gallo, L. Guibas, J. Tremblay, S. Khamis, T. Karras, and G. Wetzstein, *Efficient Geometry-aware 3D Generative Adversarial Networks*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[174] J. Gu, L. Liu, P. Wang, and C. Theobalt, *StyleNeRF: A Style-based 3D-Aware Generator for High-resolution Image Synthesis*, in *International Conference on Learning Representations (ICLR)*, 2022.

[175] Z. Yang, S. Li, W. Wu, and B. Dai, *3DHumanGAN: Towards Photo-Realistic 3D-Aware Human Image Generation*, in *arXiv:2212.07378*, 2022.

[176] R. Or-El, X. Luo, M. Shan, E. Shechtman, J. J. Park, and I. Kemelmacher-Shlizerman, *StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[177] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, *Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image*, in *European Conference on Computer Vision (ECCV)*, 2016.

[178] Z. Zheng, T. Yu, Y. Liu, and Q. Dai, *PaMIR: Parametric Model-Conditioned Implicit Representation for Image-based Human Reconstruction*, in *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[179] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, *Implicit Neural Representations with Periodic Activation Functions*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[180] S. Lombardi, T. Simon, G. Schwartz, M. Zollhoefer, Y. Sheikh, and J. Saragih, *Mixture of Volumetric Primitives for Efficient Neural Rendering*, in *Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2021.

[181] MMHuman3D, "OpenMMLab 3D Human Parametric Model Toolbox and Benchmark." `https://github.com/open-mmlab/mmhuman3d`, 2021.

[182] L. Mescheder, A. Geiger, and S. Nowozin, *Which Training Methods for GANs do actually Converge?*, in *International Conference on Machine Learning (ICML)*, 2018.

[183] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, *Implicit Geometric Regularization for Learning Shapes*, in *International Conference on Machine Learning (ICML)*, 2020.

[184] J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang, *GIT: A Generative Image-to-text Transformer for Vision and Language*, in *Transactions on Machine Learning Research (TMLR)*, 2022.

[185] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*, in *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[186] G. Pavlakos, V. Choutas, N. Ghorbani, A. A. Timo Bolkart, A. Osman, D. Tzionas, and M. J. Black, *Expressive Body Capture: 3D Hands, Face, and Body from a Single Image*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[187] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, *Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer*, in *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[188] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, *2D Human Pose Estimation: New Benchmark and State of the Art Analysis*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[189] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi, *CLIPScore: A Reference-free Evaluation Metric for Image Captioning*, in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[190] E. Richardson, G. Metzer, Y. Alaluf, R. Giryes, and D. Cohen-Or, *TEXTure: Text-Guided Texturing of 3D Shapes*, in *Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2023.

[191] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu, *AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars*, in *Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2022.

[192] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu, *MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model*, in *arXiv:2208.15001*, 2022.

[193] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or, *Null-text Inversion for Editing Real Images using Guided Diffusion Models*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[194] O. Avrahami, D. Lischinski, and O. Fried, *Blended Diffusion for Text-driven Editing of Natural Images*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[195] 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*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[196] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, *LLaMA: Open and Efficient Foundation Language Models*, in *arXiv:2302.13971*, 2023.

[197] C. Wu, S. Yin, W. Qi, X. Wang, Z. Tang, and N. Duan, *Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models*, in *arXiv:2303.04671*, 2023.

[198] H. Liu, C. Li, Q. Wu, and Y. J. Lee, *Visual Instruction Tuning*, in *arXiv:2304.08485*, 2023.

[199] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, *MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models*, in *arXiv:2304.10592*, 2023.

[200] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, *Hierarchical Text-Conditional Image Generation with CLIP Latents*, in *arXiv:2204.06125*, 2022.

[201] 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 *International Conference for Learning Representations (ICLR)*, 2022.

[202] 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*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[203] S. Wang, C. Saharia, C. Montgomery, J. Pont-Tuset, S. Noy, S. Pellegrini, Y. Onoe, S. Laszlo, D. J. Fleet, R. Soricut, J. Baldridge, M. Norouzi, P. Anderson, and W. Chan, *Imagen Editor and EditBench: Advancing and Evaluating Text-Guided Image Inpainting*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[204] O. Bar-Tal, D. Ofri-Amar, R. Fridman, Y. Kasten, and T. Dekel, *Text2LIVE: Text-Driven Layered Image and Video Editing*, in *European Conference on Computer Vision (ECCV)*, 2022.

[205] G. Couairon, J. Verbeek, H. Schwenk, and M. Cord, *DiffEdit: Diffusion-based Semantic Image Editing with Mask Guidance*, in *International Conference on Learning Representations (ICLR)*, 2023.

[206] Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, and A. Smola, *Multimodal Chain-of-Thought Reasoning in Language Models*, in *arXiv:2302.00923*, 2023.

[207] Z. Yang, L. Li, J. Wang, K. Lin, E. Azarnasab, F. Ahmed, Z. Liu, C. Liu, M. Zeng, and L. Wang, *MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action*, in *arXiv:2303.11381*, 2023.

[208] J. Li, D. Li, S. Savarese, and S. Hoi, *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*, in *International Conference on Machine Learning (ICML)*, 2023.

[209] J. Y. Koh, D. Fried, and R. Salakhutdinov, *Generating Images with Multimodal Language Models*, in *arXiv:2305.17216*, 2023.

[210] Q. Sun, Q. Yu, Y. Cui, F. Zhang, X. Zhang, Y. Wang, H. Gao, J. Liu, T. Huang, and X. Wang, *Generative Pretraining in Multimodality*, in *arXiv:2307.05222*, 2023.

[211] J. Ho and T. Salimans, *Classifier-Free Diffusion Guidance*, in *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[212] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, *Emerging Properties in Self-Supervised Vision Transformers*, in *International Conference on Computer Vision (ICCV)*, 2021.

[213] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[214] I. Loshchilov and F. Hutter, *Decoupled Weight Decay Regularization*, in *International Conference for Learning Representations (ICLR)*, 2019.