



UNIVERSITÀ
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

ON LANGUAGE AS AN INTERFACE FOR CONTROLLING IMAGE GENERATION

Zipeng Xu

Advisors

Prof. Dr. Nicu Sebe

University of Trento

July 2025

Abstract

The emergence of vision-language foundation models, such as CLIP and DALL·E, has significantly advanced the use of natural language for image generation and manipulation. These models learn broad alignments between visual and textual representations, enabling flexible and general-purpose multimodal capabilities. However, using language as an efficient and robust interface for image generation remains challenging, with limitations in controllability, semantic expressiveness, and visual fidelity.

This thesis focuses on several fundamental challenges in employing language as a control interface for image generation with foundation models. Specifically, it investigates (1) enhancing the controllability and precision of language-guided generation, (2) leveraging foundation models to explore and utilize generative latent spaces, and (3) developing a spectral perspective on CLIP embeddings to better analyze and improve generation quality. To address these aspects, we introduce three complementary approaches:

- *Predict, Prevent, and Evaluate (PPE) enhances the controllability and precision of language-guided image manipulation by modeling and regularizing attribute interactions with natural language.*
- *StylerDALL·E explores the latent space of generative models, formulating style transfer as latent translations, supervised through CLIP-based reinforcement learning to jointly preserve style and content in generation.*

- *SpectralCLIP analyzes the frequency spectrum of CLIP embeddings to suppress common artifacts in CLIP-guided generation, improving output robustness without compromising semantic alignment.*

Together, these contributions highlight the potential of natural language as a flexible and high-level interface for visual generation, grounded in the capabilities of vision-language foundation models. This thesis demonstrates how language, when appropriately modeled and guided, can effectively control diverse aspects of the generative process.

Keywords

Language-Guided Image Generation, Vision-Language Foundation Models, Image Manipulation, Image Style Transfer, Text-to-Image.

Contents

1	Introduction	3
1.1	Motivation and Challenges	3
1.2	Contributions and Overview	5
2	Disentangled Image Manipulation with Predictive Language Guidance¹	9
2.1	Problem Statement	9
2.2	Related Work	12
2.3	Background	14
2.4	Method	15
2.4.1	Predict	15
2.4.2	Prevent	20
2.4.3	Evaluate	22
2.5	Experiments	23
2.5.1	Implementation Details	23
2.5.2	Quantitative Results	24
2.5.3	Qualitative Results	28
2.6	Discussions and Conclusion	29
2.6.1	Discussions	29
2.6.2	Limitations	31
2.6.3	Conclusion	31

3	Language-Guided Image Style Transfer in Generative Latent Space²	33
3.1	Problem Statement	33
3.2	Related Work	38
3.3	Background	39
3.4	Method	41
	3.4.1 Architecture and self-supervised pre-training	42
	3.4.2 Style-specific fine-tuning	43
3.5	Experiments	46
	3.5.1 Training Details	46
	3.5.2 Experimental Results	47
3.6	Conclusion	55
4	Spectral Filtering for CLIP-Guided Image Generation³	57
4.1	Problem Statement	57
4.2	Related Work	61
4.3	Method	62
	4.3.1 Spectral based filtering	62
	4.3.2 Computing an image-text similarity	65
	4.3.3 Band selection	66
4.4	Experiments	68
	4.4.1 Style Transfer Results	68
	4.4.2 Comparison with Forget-to-Spell CLIP	70
	4.4.3 Hyperparameter Study	73
	4.4.4 Text-to-Image Generation	75
4.5	Discussions and Conclusion	76
	4.5.1 Discussions	76
	4.5.2 Limitations	81
	4.5.3 Conclusion	81

5 Conclusion and Future Work	83
Bibliography	87

List of Tables

2.1	The complete hierarchical human face attributes.	18
2.2	Quantitative comparisons (1/2) of disentangled text-driven image manipulation with StyleCLIP [123], using the evaluation metrics introduced in Sec. 2.4.3. For the <i>indicator</i> , lower is better. The text command is indicated under each sub-table. Specifically, we illustrate each individual item in the changes of predicted entangled attributes $\Delta d'_e$	25
2.2	Quantitative comparisons (2/2) of disentangled text-driven image manipulation with StyleCLIP [123], using the evaluation metrics introduced in Sec. 2.4.3. For the <i>indicator</i> , lower is better. The text command is indicated under each sub-table. Specifically, we illustrate each individual item in the changes of predicted entangled attributes $\Delta d'_e$	26
3.1	Quantitative comparisons with CLIPstyler [79] on different types of styles and datasets.	49
3.2	Preference scores of StylerDALLE user study.	52
4.1	Correspondences between the bands, frequency indexes and period (tokens). The corresponding periods are approximate numbers of tokens that are needed to complete a cosine wave cycle.	66

4.2	Average cosine similarity between the stylized images and the textual description of the style, measured both on the original CLIP space ($\uparrow$) and on the learn-to-spell CLIP (in short referred to as CLIP-Spell) ($\downarrow$).	70
4.3	User preference of the three style transfer methods with respect to the overall quality of the generated images ($\uparrow$) and the presence of artifacts ($\uparrow$).	71
4.4	Band combination used for each style.	80

List of Figures

2.1	Comparisons on disentangled image manipulation between the StyleCLIP [123] baseline and our Predict, Prevent, and Evaluate (PPE). Ours manages to manipulate only the command-attribute (as indicated under each column) while remaining unchanged to the others.	10
2.2	To predict entangled attributes in face editing, we construct a hierarchical attribute structure with the help of BERT [33].	16
2.3	Illustration of the predicted entangled attributes for various commands in text-driven image manipulation.	20
2.4	Illustration of (a) the effect of the entanglement loss and (b) the expected result provided by the entanglement loss. . .	21
2.5	Qualitative comparison with StyleCLIP [123] using different text commands (indicated on the top). Ours achieves more disentangled manipulation results as only the desired attribute is manipulated while others are maintained well.	24
2.6	Image manipulation results from StyleCLIP [123] and ours, using gradually increasing manipulation strengths. Our method presents better fore-and-aft consistency along the change of manipulation strength.	27
2.7	Hyper-parameters study of PPE.	30
2.8	Multiple attribute manipulations with PPE.	31

3.1	With a large-scale pretrained vector-quantized tokenizer (e.g., the dVAE of DALL-E) and CLIP, StylerDALLE can transfer various types of styles (indicated on the top), from abstract art styles to specific artist styles and more. . . .	34
3.2	A schematic illustration of StylerDALLE . We propose a NAT to translate tokens of a half-resolution <i>content</i> image to tokens of a full-resolution <i>stylized</i> image. StylerDALLE consists of three parts: 1) image tokenization preprocessing, to obtain the discrete visual tokens from a pretrained image tokenizer; 2) self-supervised pretraining, to train the NAT to predict full-resolution content images from half-resolution content images; and 3) style-specific fine-tuning, to let the model adding style details in the up-scaling prediction via Reinforcement Learning with language-prompted CLIP guidance. Specifically, in our NAT, the decoder input is duplicated from the tokens of the half-resolution image, following the natural positional correspondence (as highlighted by orange rectangles in the left part) with the full-resolution image.	40
3.3	Qualitative results of StylerDALLE-1 on various styles: (a) Monet Sunrise, (b) Picasso cubism, (c) Van Gogh blue color, (d) Van Gogh purple color, (e) warm and relaxing, (f) ukiyo-e print, (g) fauvism, (h) pixel art illustration. The style references at the top are for illustration only (not used as input to the model).	45

3.4	Qualitative results of StylerDALLE -Ru on various styles: (a) Monet, (b) Monet Paris, (c) Monet Venice, (d) Van Gogh, (e) Van Gogh Irises, (f) Van Gogh Almond, (g) Van Gogh Starry Night, (h) Van Gogh Sunflowers. The style references at the top are for illustration only (not used as input to the model).	46
3.5	Comparisons with language-guided methods. The styles are (a) fauvism, (b) Monet, (c) Monet Impression Sunrise, (d) oil painting, (e) watercolor painting, (f) Van Gogh, (g) Van Gogh blue color. The style references at the top are for illustration only (not used as input to the model).	48
3.6	Illustrations of the StylerDALLE user study details. . . .	51
3.7	Comparisons with reference image-based methods. StylerDALLE is able to transfer more abstract concepts, e.g., specific painting strokes, and is less likely to produce semantic errors.	52
3.8	Ablation study on StylerDALLE	54
4.1	Text-guided style transfer results using either a CLIP-based image-text similarity or SpectralCLIP. While using the standard CLIP leads to the generation of undesirable artifacts (written words and unrelated visual entities as <i>highlighted</i>), the proposed SpectralCLIP prevents the artifacts while maintaining other style features.	58

4.2	An illustration of SpectralCLIP. To transfer a “cartoon” style to the leftmost content image, CLIPstyler generates many cartoon-like artifacts, spreading over the whole image (central figure). The corresponding spectral representation is a composition of frequencies with different periods. Removing the frequencies corresponding to the artifact scales (SpectralCLIP) prevents the generation of these unwanted artifacts (right figure).	63
4.3	Style transfer results using either the CLIP or SpectralCLIP to condition the generation. SpectralCLIP effectively prevents artifact generation while achieving similar style features, e.g., color, paint stroke (artifacts are <i>highlighted</i>).	67
4.4	Comparisons among text-guided style transfer results, generated using CLIPstyler with (a) CLIP, (b) forget-to-spell CLIP [107], and (c) SpectralCLIP, respectively (artifacts are <i>highlighted</i>).	68
4.5	The SpectralCLIP user study questions used to evaluate two tasks: (a) the overall image quality, and (b) the possible appearance of artifacts.	72
4.6	The effect of the patch-rejection threshold τ in the CLIPstyler patch-loss (artifacts are <i>highlighted</i> , zoom in to see details).	73
4.7	The effects of different band-stop filters (artifacts are <i>highlighted</i>).	74
4.8	Text-to-image generation results using (a) VQGAN+CLIP, and (b) VQGAN+SpectralCLIP.	76
4.9	Different styles result in artifacts at different scales which are tricky to predict.	77

4.10	By CLIPstyler, different styles lead to different types of artifacts. Therefore, in SpectralCLIP, we filter different band combinations (indicated on the bottom).	78
4.11	Leftmost column: images generated using CLIPstyler and three different styles. All the other columns show the results obtained using SpectralCLIP and one among c_1, c_2 or c_3 as the filter. In case of “outsider art”, c_1 is the best band combination, while c_2 is the best for “cartoon”, and c_3 for “digital art”.	79
4.12	Spectral analysis w.r.t. textual artifacts.	80

Publications

This thesis consists of the following publications:

- Chapter 2:
 - **Zipeng Xu**, Tianwei Lin, Hao Tang, Fu Li, Dongliang He, Nicu Sebe, Radu Timofte, Luc Van Gool, Errui Ding. Predict, Prevent, and Evaluate: Disentangled Text-Driven Image Manipulation Empowered by Pre-Trained Vision-Language Model. In Proceedings of IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR), 2022.
- Chapter 3:
 - **Zipeng Xu**, Enver Sangineto, Nicu Sebe. StylerDALLE: Language-Guided Style Transfer Using a Vector-Quantized Tokenizer of a Large-Scale Generative Model. In Proceedings of IEEE / CVF International Conference on Computer Vision (ICCV), 2023.
- Chapter 4:
 - **Zipeng Xu**, Songlong Xing, Enver Sangineto, Nicu Sebe. SpectralCLIP: Preventing Artifacts in Text-Guided Style Transfer from a Spectral Perspective. In Proceedings of IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2024.

Chapter 1

Introduction

1.1 Motivation and Challenges

Natural language has long been recognized as one of the most expressive and accessible tools for human communication, enabling abstract, compositional, and efficient expression across diverse contexts [165, 27, 26, 66, 43]. In the context of image generation, it serves as a uniquely powerful interface: users can describe desired content, modifications, or styles using free-form text, without requiring visual references or technical editing skills [131, 112, 137, 139, 130, 117, 179, 34, 35, 14, 168, 169, 123, 59, 51, 12, 144, 79, 8, 94, 6, 18, 62, 69, 74, 75, 82, 83, 88, 124, 127, 133, 145, 170, 80]. This paradigm—controlling image generation through language—has gained substantial momentum with the development of large-scale vision-language foundation models such as CLIP [129] and DALL·E [131], which have significantly expanded the capabilities of language-driven visual systems.

Trained on large-scale image-text datasets, vision-language foundation models such as CLIP and DALL·E exhibit strong alignment between visual and linguistic semantics [181, 91, 106, 146, 148, 68, 4, 87, 86, 98, 118, 85, 156, 178, 149, 40, 157, 160, 29, 39, 22, 5, 161, 2]. CLIP learns a joint embedding space for images and text, while DALL·E and its vari-

ants generate images directly from language prompts using autoregressive or diffusion-based decoding. These models offer broad generalization and multimodal grounding, forming a solid foundation for building language-driven image generation systems. However, when repurposed as control interfaces, they also introduce subtle technical and semantic challenges [123, 79, 180, 182, 53, 144, 45, 64, 159, 184, 174, 37].

First, language-guided image editing often leads to entangled changes in visual attributes [123, 168, 169]. For example, modifying an image to depict “gray hair” may unintentionally affect other age-related features, such as wrinkles or skin tone. This reflects a broader challenge: semantic attributes in natural images often co-occur, making it difficult to isolate the effect of a single instruction. Traditional disentanglement methods rely on manually annotated attributes or predefined editing directions, which lack flexibility and scalability [142, 1, 65, 143, 154, 155]. While recent approaches have begun to incorporate natural language via models like CLIP, they still struggle to disentangle complex semantic correlations without extensive supervision or trial-and-error. This raises the need for mechanisms that can anticipate and minimize such entanglements when using language as the primary interface for control.

Second, language-guided style transfer faces significant challenges in capturing abstract or high-level artistic aesthetics—such as “painted in Fauvist style” or “inspired by Van Gogh” [79, 46]. Traditional style transfer methods typically rely on reference images to extract low-level features like texture and color, which are insufficient for conveying more conceptual or compositional aspects of artistic styles [52, 63, 152, 163, 176, 120, 101, 31, 162]. While set-based methods attempt to learn styles from collections of artworks, they require extensive data collection and are limited to predefined style categories [140, 78, 16]. Although recent generative models and vision-language models offer new opportunities for using language as

a style descriptor, effectively translating free-form textual style prompts into faithful visual outputs remains difficult. Achieving stylization that faithfully reflects abstract textual prompts—while preserving content and avoiding visual artifacts—remains an open challenge for current vision-language generative models.

Third, optimizing images based on CLIP similarity often introduces visual artifacts—such as repeated patterns, hallucinated objects, or unintended textual elements [79, 107, 28]. These issues are particularly evident in text-guided style transfer, where models may inadvertently render words from the prompt or generate over-specific visual symbols. Such artifacts arise from the entangled nature of visual and textual concepts in the CLIP embedding space, which lacks explicit disentanglement between semantic content and written language [54, 81, 107]. While the origins of these artifacts remain underexplored, some of these repeated artifacts appear to correlate with specific frequency components in CLIP encodings, motivating the exploration of spectral perspectives as a potential means to analyze and mitigate such effects during generation.

Together, these challenges highlight the gap between the general capabilities of vision-language foundation models and the demands of precise, abstract, and robust control in image generation—underscoring the need for dedicated methods that treat language as a genuine interface for controllable synthesis.

1.2 Contributions and Overview

This thesis investigates the use of natural language as a high-level interface for controlling image generation, with a focus on enhancing controllability, semantic precision, and visual robustness. While vision-language foundation models have unlocked new possibilities for multimodal synthesis, em-

ploying language as a precise and interpretable control mechanism remains challenging. We address these limitations through three complementary contributions:

- **Disentangled Image Manipulation with Predictive Language Guidance.** We propose *Predict, Prevent, and Evaluate (PPE)* [171], a framework for language-guided image manipulation that anticipates and regularizes attribute entanglement. By modeling the co-occurrence patterns of natural-language instructions and visual attributes using CLIP, PPE improves edit specificity while minimizing undesired changes.
- **Language-Guided Image Style Transfer in Generative Latent Space.** We introduce *StylerDALLE* [172], a token-based style transfer approach that operates in the discrete latent space of a pretrained generative model. By combining a non-autoregressive translator with CLIP-guided reinforcement learning, the model effectively applies abstract, text-defined styles while preserving content.
- **Spectral Filtering for CLIP-Guided Image Generation.** We develop *SpectralCLIP* [173], a method to reduce visual and textual artifacts in CLIP-guided generation by identifying and filtering frequency-specific components in CLIP embeddings. This approach enhances generation quality without sacrificing semantic alignment.

Each of these contributions addresses a key limitation in current language-driven generative methods, and collectively they demonstrate how natural language can serve as a powerful, flexible interface for visual synthesis.

The remainder of this thesis is structured as follows. Chapter 2 (*Disentangled Image Manipulation with Predictive Language Guidance*) intro-

duces our framework for disentangled language-guided editing. Chapter 3 (*Language-Guided Image Style Transfer in Generative Latent Space*) presents a token-based approach for abstract style transfer using language supervision. Chapter 4 (*Spectral Filtering for CLIP-Guided Image Generation*) discusses artifact suppression through frequency-based filtering of CLIP embeddings. Finally, Chapter 5 provides an integrative discussion and outlines potential future directions.

Chapter 2

Disentangled Image Manipulation with Predictive Language Guidance¹

2.1 Problem Statement

Disentangled image manipulation [102, 142, 1, 90, 47, 49, 65, 143, 154, 155] aiming at changing the desired attributes of the image while keeping the others unchanged, has long been studied for its research significance and application value. Reaching this target is not easy, especially when attributes naturally entangle in the real world. Therefore, concrete attribute annotations are of vital importance, making disentangled image manipulation a labor-consuming task.

Several works [102, 90, 47, 49] use an encoder-decoder architecture and need manual annotations on multiple attributes of images. The models encode the original image and the manipulating attribute, then decode the manipulated image. Specifically, they use an attribute-specific loss to encourage the manipulation of a specific attribute while discouraging the others. The loss comes from pre-trained classifiers for all annotated attributes. Many recent works focus on latent space image manipulation

¹This chapter is based on our published work: Xu et al., "Predict, Prevent, and Evaluate: Disentangled Text-Driven Image Manipulation Empowered by Pre-Trained Vision-Language Model", CVPR 2022.



Figure 2.1: Comparisons on disentangled image manipulation between the StyleCLIP [123] baseline and our Predict, Prevent, and Evaluate (PPE). Ours manages to manipulate only the command-attribute (as indicated under each column) while remaining unchanged to the others.

since large-scale pre-trained GANs, *e.g.*, StyleGANs [71, 72], can generate high-quality images from well-disentangled latent spaces. Despite the convenience of directly using the pre-trained GANs to generate images, all these methods need human annotations [142, 1, 65, 143, 154, 155]. Moreover, the available manipulating attributes are limited to the annotated set.

Recently, the rise of the large-scale pre-trained vision-language model CLIP [129] has brought a new insight. Since CLIP provides effective signals about the semantic similarity of image and text, various manipulations [123, 51, 136] can be performed with a text command and a CLIP-based loss, instead of exhaustive human annotations. Nevertheless, achieving disentangled image manipulation is still tricky. For instance, StyleCLIP [123] introduces three methods: latent optimization and latent mapper take no consideration of achieving disentangled results; global direction, which is based on the more disentangled $\mathcal{S}$ latent space [167], needs human trials-and-errors to find appropriate parameters in each case to reach the expected effects. To only manipulate a desired attribute, TeDiGAN [168, 169] merely revise the latent vectors of layers corresponding

to that attribute. Yet, they have to figure out in advance the relations between attributes and layers in StyleGAN.

In this work, we explore achieving disentangled image manipulation with as less human labor as possible. We propose a novel framework, *i.e.*, Predict, Prevent, and Evaluate (PPE), to approach the target by leveraging the power of CLIP in depth. Firstly, we propose to **Predict** the possibly entangled attributes for given text commands. We assume that the entanglements result from the distributions of attributes in the real world. Therefore, we draw support from CLIP to find the attributes that appear most frequently in the command-related images, then regard the attributes of high co-occurrence frequency as the possibly entangled attributes. Secondly, we introduce a novel entanglement loss to **Prevent** entanglements during training. The loss punishes the changes of the possibly entangled attributes before and after the manipulation, so as to enforce the model to find a less disentangled manipulating direction. Lastly, based on the predicted entangled attributes, we introduce a new evaluation metric to simultaneously **Evaluate** the manipulation effect and the entanglement condition. The manipulation effect is measured based on the change of command-attribute while the entanglement condition is based on the change of the entangled attributes, before and after manipulation. All the changes are estimated according to the CLIP distance between the texts of attributes and the images.

To evaluate, we implement our method based on the simple and versatile latent mapper from StyleCLIP and conduct experiments on the challenging face editing task, using the large-scale human face dataset CelebA-HQ [105, 70]. Qualitative and quantitative results indicate that we achieve superior disentangled performance compared to the StyleCLIP baseline. Meanwhile, we show that our results present a better linear consistency.

To conclude, our main contributions are as follows:

- We propose to predict entangled attributes for disentangled image manipulation.
- We propose a novel entanglement loss to prevent entangled manipulations during training.
- We propose a new evaluation metric that jointly measures the manipulation effect and the entanglement condition for disentangled image manipulation.
- By applying our method to the versatile StyleCLIP baseline, we manage to achieve disentangled image manipulation with very little manual labor. We conduct extensive experiments on the CelebA-HQ dataset and find that our qualitative and quantitative results are rather impressive.

2.2 Related Work

Disentangled Image Manipulation. Many works study learning a disentangled representation [19, 76, 21, 48, 90], so that disentangled image manipulation can be solved from the source. Due to the costly labor, a key challenge of such works is reducing the supervision for learning the desired disentanglement. Therefore, weakly-supervised and unsupervised methods have been explored [113, 49, 47, 103]. Despite progress, all these methods are trained for a fixed set of attributes, thus supporting limited numbers of manipulations.

Recently, growing numbers of works focus on latent space image manipulation [65, 142, 67, 150, 175, 167] because of the remarkable large scale GANs like StyleGAN [71, 72], which can generate high-resolution images with well disentangled latent space. Thereby, these works firstly invert the image into the latent space through the GAN inversion method [72, 185] or

an involved encoder [135, 114, 3], then accordingly compute the latent vector that can derive the manipulation result through the pre-trained large scale GANs. For each manipulating attribute, manual annotations are required, *e.g.*, on images [142, 1] and on unsupervisedly discovered directions in the latent space [65, 143, 154, 155].

Text-Driven Image Manipulation. There are studies that explore image manipulation with text commands as a guide. Some previous works [36, 110, 83, 84] use GAN-based encoder-decoder architectures, which encode the original image and text command, disentangle the semantics of the two modalities and decode the manipulated image. Instead of training a generator individually, the recent TediGAN [168, 169] and StyleCLIP [123] use pre-trained StyleGAN to generate images from manipulated latent vectors. To reach disentangled manipulation, TediGAN pre-defines an attribute-to-layer map and only changes the attribute-corresponding layers in StyleGAN. Besides, TediGAN conducts instance-level manipulation, which means the model is only applicable to one image that the model was optimized for. The latent mapper method in StyleCLIP is more general as the trained model can be applied to manipulate any in-domain image, but the results are usually entangled. The global direction method in StyleCLIP can realize a disentangled manipulation, but it requires manual trials-and-errors to find appropriate thresholds in the method. Our method proposes to achieve disentangled image manipulations with less manual effort by deeply leveraging the power from large scale pre-trained models. More than StyleCLIP, which merely minimizes the CLIP distances between command texts and manipulated images, we propose to predict, prevent, and evaluate entanglements via CLIP.

Large Scale Vision-Language Models. Following the success of large scale pre-trained language models, *e.g.*, BERT [33], various large scale pre-trained vision-language models [181, 91, 106, 146, 148] are proposed. The recent CLIP [129] is especially remarkable because it is trained from 400 million text-image pairs and is powerful. CLIP learns a multi-modal embedding space, which can be used to measure the semantic similarity of image and text. Using text descriptions as prompts enables CLIP the strong ability of zero-shot transfer to downstream tasks. Besides, stunning text-guided image synthesis results [131, 44, 51, 136] are enabled by CLIP through utilizing the embedding space.

2.3 Background

StyleCLIP [123] proposes a flexible latent mapper method for text-driven image manipulation. It is trained for a specific text command and is applicable for any image in the domain of the pre-trained StyleGAN [72]. For the text command t_{cmd} , the method learns a mapper network $M_{t_{cmd}}$ to yield a manipulation direction in the $\mathcal{W}+$ space given the latent image embedding $w \in \mathcal{W}+$. Then the manipulated image is obtained from a pre-trained StyleGAN generator G as $i' = G(w + M_{t_{cmd}}(w))$.

To train the mapper network for the purpose of achieving the text-driven manipulating effect, a CLIP loss L_C is introduced to minimize the distance between the text command t_{cmd} and manipulated image i' . L_C is formalized as:

$$\mathcal{L}_C = D_{CLIP}(i', t_{cmd}), \quad (2.1)$$

where D_{CLIP} is the cosine distance between the embeddings of its two arguments in CLIP space. In addition, the method uses L_2 loss to norm the manipulation direction and L_{ID} loss [135] to maintain the identity of

person. Hence, the overall loss is formulated as:

$$\mathcal{L}_{StyleCLIP} = \mathcal{L}_C + \lambda_{L2}\mathcal{L}_{L2} + \lambda_{ID}\mathcal{L}_{ID}, \quad (2.2)$$

where λ_{L2} and λ_{ID} are the loss coefficients. Although the method can simply and efficiently achieve text-driven image manipulation without human annotations, its loss cannot distinguish between entangled and disentangled manipulations, and the manipulated results are always entangled.

2.4 Method

The proposed Predict, Prevent, and Evaluate (PPE) framework consists of three parts: 1) we design a mechanism to **Predict** the entangled attributes for given text commands; 2) based on the predicted attributes, we introduce a novel entanglement loss to **Prevent** entanglements during training, and 3) we propose a new evaluation metric to **Evaluate** disentangled text-driven image manipulation. All methods leverage the power from the large-scale pre-trained vision-language model CLIP.

2.4.1 Predict

We predict the entangled attributes under the assumption that entanglements result from the frequent co-occurrence of attributes in real-world images. To this end, we aggregate the images most relevant to the text command, look for the attributes that appear most frequently in the images and predict them as the entangled attributes.

Prerequisite. A predefined attribute set that includes basic visual characteristics is the prerequisite. For manipulating human faces, we need human face attributes. To obtain useful human face attributes, we firstly draw support from the large-scale pre-trained language model BERT [33].



Figure 2.2: To predict entangled attributes in face editing, we construct a hierarchical attribute structure with the help of BERT [33].

In concrete, we let BERT predict specific attributes under different categories with designed prompts like “a face with [MASK] eyes”. By substituting “eyes” with other keywords of face characteristics, we derive various face attributes in a category-to-attribute fashion.

After further sorting and adding binary attributes like “with earrings”, we construct a hierarchical attribute structure (see Fig. 2.2) that serves for the subsequent procedures in Predict. In Table 2.1, we give a complete version of the hierarchical attribute structure for human face. The construction steps are as follows:

- Firstly, based on common sense, we establish the category-items, *e.g.*, gender and hair.

- Secondly, we use the large-scale pre-trained language model BERT [33] to mine useful human face attributes under each category. We design a series of prompts to let the pre-trained BERT predict specific attributes for every category, using an open-source tool, *i.e.*, LAMA [134]. We find prompts like “a face/person with a/an [MASK] [X]”, “the person’s [X] is/are [MASK]” and “a [MASK] [X]” are helpful, where [MASK] is the word that BERT has to predict and [X] is the concrete category (*e.g.*, hair, nose, mouth and etc.) in practice.
- Furthermore, among the many predicted attributes, we select the representative ones and sort them into more fine-grained categories, such as hair color and hair length.
- Lastly, we include some binary attributes, especially drawing on the previously proposed human face attributes in the CelebA dataset [105].

Aggregate. This step aims to aggregate the images that are most relevant to the text command. Specifically, we propose a method based on CLIP. At first, we rank all the images in the training set *w.r.t.* their distance to the text command t_{cmd} in the CLIP space. For an image $i \in I$, its ranking score is formalized as:

$$score(i) = D_{CLIP}(i, t_{cmd}). \quad (2.3)$$

Images are ranked by their scores from small to large.

Besides, we use a zero-shot CLIP classifier to exclude the images that are classified as irrelevant in the ranked list. For single attribute manipulation, the classification labels can be obtained via a command-to-category and category-to-attributes pipeline. Take command “blue eyes” as an example, we firstly find its category $\langle eyes\ color \rangle$ with the help of an NLP tool, then the labels $\{“blue\ eyes”, “brown\ eyes”, \dots\}$ can be easily obtained via a

Category	Attributes
gender	male, female
hair color	black hair, blond hair, brown hair, grey hair, red hair
hair length	long hair, short hair, no hair
hair style	curly hair, straight hair, bald, wavy hair, receding hairline
eye color	blue eyes, brown eyes, black eyes, grey eyes, green eyes
eye status	open eyes, closed eyes
eye shape	narrow eyes, wide eyes, big eyes, small eyes, round eyes
nose shape	big nose, long nose, pointed nose, small nose, hooked nose, short nose, thick nose, thin nose, pinched nose, flat nose
face shape	pointy face, round face, square face, oval face, long face
skin color	white skin, black skin, yellow skin
mouth status	open mouth, closed mouth
mouth size	big mouth, small mouth
eyebrows	round eyebrows, high eyebrows, arched eyebrows, long eyebrows, thick eyebrows, dark eyebrows, straight eyebrows, thin eyebrows, short eyebrows
beard	goatee, mustache, no beard, sideburns, 5 o'clock shadow
earrings	with/without earrings
makeup	with/without makeup
smile	with/without smile
lipstick	with/without lipstick
wrinkles	with/without wrinkles
glasses	with/without glasses
bangs	with/without bangs
rosy cheeks	with/without rosy cheeks
bags under eyes	with/without bags under eyes
high cheekbones	with/without high cheekbones

Table 2.1: The complete hierarchical human face attributes.

category-to-attribute map according to the hierarchical attribute structure. Particularly, binary attributes like “*with earrings*” will trigger binary labels as { “*with earrings*”, “*without earrings*” }. Afterwards, we select up to top-100 images in the left ranked list to form the command-relevant image-set

I' .

Find. The last step is to find the attributes that appear most frequently in the command-relevant image set I' , except for attributes in the same category as commands. First, we rank the attributes by the sum of their CLIP distances to the images in I' . The ranking score is formalized as:

$$score_{comd}(t_{attr}) = \sum_{i' \in I'} D_{CLIP}(i', t_{attr}). \quad (2.4)$$

Meanwhile, we consider the ranking results *w.r.t.* the full image set I . Similarly, the ranking score is:

$$score_{full}(t_{attr}) = \sum_{i \in I} D_{CLIP}(i, t_{attr}). \quad (2.5)$$

By sorting the scores in descending order, we get r_{comd} and r_{full} . In addition, we need to find attributes that only appear frequently in the command-relevant images. For instance, “square face” is common for all images and thus ranks high in r_{full} , thus it may not be the entangled one even if it ranks high in r_{comd} . To this end, we adjust the r_{comd} with r_{full} . In concrete, the final ranking score is:

$$score_{final}(t_{attr}) = \frac{r_{comd}(t_{attr})}{\min(r_{full}(t_{attr}), R)}, \quad (2.6)$$

where R is a hyper-parameter to determine if the rankings in r_{full} are high or not. Eventually, top- N attributes in the final ranked list (obtained by sorting the $score_{final}$ from small to large) are predicted as the entangled attributes $\{t_{entg^n}\}_{n=1}^N$.

Analysis. In Fig. 2.3, we illustrate some of the predictions and the corresponding manipulation results from the latent mapper of StyleCLIP. As



Figure 2.3: Illustration of the predicted entangled attributes for various commands in text-driven image manipulation.

can be seen, our method predicts the entangled attributes well, *e.g.*, “wide eyes” and “with makeup” for blue eyes, “grey eyes” and “white skin” for grey hair, “short hair” and “with makeup” for with earrings, and “grey hair” and “closed eyes” for with wrinkles.

2.4.2 Prevent

For a disentangled manipulation, the command-corresponding attribute should change while other attributes should be maintained, especially for the possibly entangled ones. Therefore, based on the predicted entangled attributes $\{t_{entg^n}\}_{n=1}^N$, we introduce a novel entanglement loss that punishes the changes of entangled attributes after the manipulation. The changes are measured by the CLIP distances between images and texts of entangled

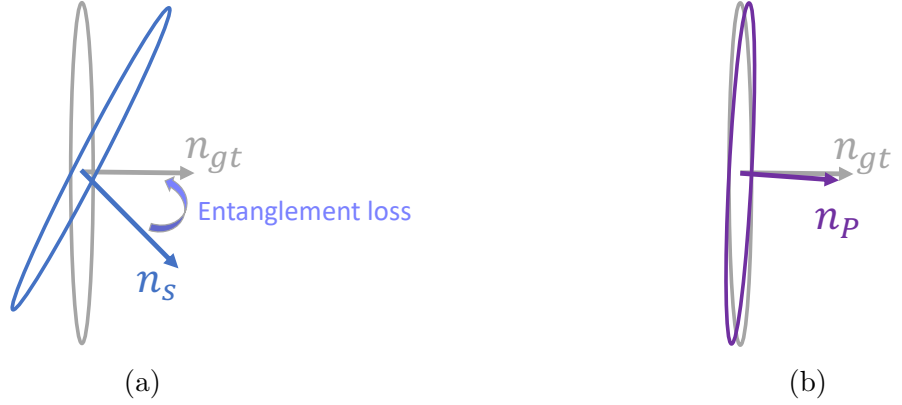


Figure 2.4: Illustration of (a) the effect of the entanglement loss and (b) the expected result provided by the entanglement loss.

attributes, thus the proposed entanglement loss is formulated as:

$$\mathcal{L}_E = \frac{1}{N} \sum_n (D_{CLIP}(i, t_{entg^n}) - D_{CLIP}(i', t_{entg^n}))^2, \quad (2.7)$$

where $i = G(w)$ is the original image and i' is the manipulated image as introduced in Sec. 2.3.

Together with the losses described in Eq. (2.2), our overall loss is defined as below:

$$\mathcal{L}_{PPE} = \mathcal{L}_C + \lambda_{L2}\mathcal{L}_{L2} + \lambda_{ID}\mathcal{L}_{ID} + \lambda_E\mathcal{L}_E, \quad (2.8)$$

where λ_E is the coefficient for the entanglement loss.

We give an illustration in Fig. 2.4, where we assume there is a hyperplane in the latent space that separates having the attribute or not², n_{gt} is the unit normal vector of the hyperplane corresponding to the command attribute, n_s is the vector found by StyleCLIP method and n_p is from PPE. As shown, the proposed entanglement loss is to constrain the model to find less entangled manipulating directions.

²The assumption draws from InterFaceGAN [142].

2.4.3 Evaluate

For disentangled text-driven image manipulation, we propose a new evaluation metric, *i.e.*, an indicator that evaluates the manipulation and the entanglement effects simultaneously, based on the predicted entangled attributes.

Firstly, for each text command, we quantify the manipulation effect as:

$$\Delta d_c = D_{CLIP}(i, t_{cmd}) - D_{CLIP}(i', t_{cmd}), \quad (2.9)$$

i.e., the change of command attribute in images measured by CLIP. The larger the Δd_c is, the closer the manipulated image to the text command is in CLIP space, indicating the manipulation reaches the command-required effect.

Meanwhile, for each predicted entangled attribute, we measure the entanglement effect by:

$$\Delta d_{e^n} = D_{CLIP}(i, t_{entg^n}) - D_{CLIP}(i', t_{entg^n}), \quad (2.10)$$

i.e., the change of entangles attribute in images estimated by CLIP. The larger the Δd_{e^n} is, the closer the manipulated image to the text of the entangled attribute is in CLIP space, indicating the manipulated image is entangled with the command-relevant attribute.

To reach disentangled manipulations, we expect Δd_c to be as large as possible while $\{|\Delta d_{e^n}|\}$ to be as small as possible. Thereby, we formalize the indicator as:

$$indicator = \frac{\frac{1}{N} \sum_{n=1}^N |norm(\Delta d_{e^n})|}{norm(\Delta d_c)}, \quad (2.11)$$

where N is the number of the predicted entangled attributes and $norm(\cdot)$ is to make Δd_c and $\{\Delta d_{e^n}\}$ comparable. In concrete, they are normalized

individually as:

$$norm(\Delta d_t) = \frac{\Delta d_t}{\max_{i \in I} D_{CLIP}(i, t) - \min_{i \in I} D_{CLIP}(i, t)}, \quad (2.12)$$

where t is in $\{t_{cmd}, t_{entg^1}, \dots, t_{entg^N}\}$ and I is image set.

As described in Eq. (2.11), assuming Δd_c is greater than 0 (as it should be), a high *indicator*, e.g., 0.5, indicates an entangled manipulation, because when its Δd_c increases, its $\{\Delta d_{e^n}\}$ grows correspondingly and significantly. By contrast, a lower *indicator* indicates a better disentangled manipulation, since its $\{|\Delta d_{e^n}|\}$'s changes are not as significant as its Δd_c 's. Using the *indicator*, the effect of disentangled image manipulation can be quantified.

2.5 Experiments

2.5.1 Implementation Details

To verify the proposed method, we conduct experiments on the challenging face editing task. We compare our method with our strong baseline, i.e., the latent mapper in StyleCLIP [123]. Following StyleCLIP, we use the CelebA-HQ dataset [105, 70], which consists of 30,000 images, 27,176 for train-set and 2,824 for test-set; StyleGAN2 [72] pre-trained on FFHQ [71] is used to generate images; e4e [151] is used to invert images into latent embeddings in the latent space of StyleGAN2. Moreover, we train all models following the original settings as the official StyleCLIP implementation [122]. In other words, for all text commands, we train the corresponding models without tuning the hyper-parameters. We use the same loss-coefficients setting, which is $\lambda_{L2} = 0.8$ and $\lambda_{ID} = 0.1$. For the proposed entanglement loss, $\lambda_E = 100$. The number of predicted entangle attributes N in the entanglement loss (Eq. (2.7)) is set to 10 by default. R in Eq. (2.6)



Figure 2.5: Qualitative comparison with StyleCLIP [123] using different text commands (indicated on the top). Ours achieves more disentangled manipulation results as only the desired attribute is manipulated while others are maintained well.

is set to 40, empirically.

2.5.2 Quantitative Results

We conduct multiple experiments using different text commands, which especially include the ones that are regarded as entangled in previous works [90, 167]. In Table 4.2, we illustrate the quantitative results using the evaluation metric introduced in Sec. 2.4.3. In concrete, *indicator* is the overall metric for disentangled image manipulation. Lower *indicator* means target manipulation is achieved with fewer entanglements, and vice versa. In addition, $\Delta d'_c$ is the normalized Δd_c , and $\Delta d'_e$ is the normalized Δd_e , as in Eq. (2.9), Eq. (2.10) and Eq. (2.12).

According to the results, we draw the following two conclusions: 1) The latent mapper method in StyleCLIP is highly entangled, and our method predicts the entangled attributes well. As can be seen in the results of “StyleCLIP”, the manipulated images are closer to text commands while they are also closer to the text of entangled attributes. For example, for text command “grey hair” (Table 2.2a), when $\Delta d'_c$ reaches 0.4878, $\Delta d'_e$ changes at a comparable scale as “grey eyes” is 0.2433 and “white skin” is 0.2641. More significantly, for text command “blue eyes” (Table 2.2k),

	StyleCLIP	Ours
<i>indicator</i> (↓)	0.3359	0.0071
$\Delta d'_c$	0.4878	0.3519
$\Delta d'_e$	short eyebrows	0.1637
	short hair	0.1945
	with bangs	0.0927
	grey eyes	0.2433
	sideburns	0.1548
	narrow eyes	0.1393
	high cheekbones	0.0873
	white skin	0.2641
	pointy face	0.1362
	with makeup	0.1626

(a) grey hair

	StyleCLIP	Ours
<i>indicator</i> (↓)	0.4022	0.1691
$\Delta d'_c$	0.2877	0.1442
$\Delta d'_e$	blue eyes	0.1249
	long hair	0.1591
	brown hair	0.1349
	with makeup	0.1274
	wide eyes	0.1186
	with earrings	0.0940
	with bangs	0.0815
	pinched nose	0.1027
	with lipstick	0.1136
	closed mouth	0.1004

(c) wavy hair

	StyleCLIP	Ours
<i>indicator</i> (↓)	0.3269	0.1298
$\Delta d'_c$	0.3679	0.1341
$\Delta d'_e$	grey hair	0.2560
	receding hairline	0.0417
	no beard	0.0858
	long eyebrows	0.1132
	long face	0.1351
	male	0.1008
	narrow eyes	0.1096
	big nose	0.0842
	black eyes	0.0937
	closed eyes	0.1829

(e) with wrinkles

	StyleCLIP	Ours
<i>indicator</i> (↓)	0.5553	0.1411
$\Delta d'_c$	0.3628	0.1898
$\Delta d'_e$	short eyebrows	0.2362
	with bangs	0.1583
	short hair	0.1927
	black eyes	0.2555
	narrow eyes	0.2005
	high cheekbones	0.2002
	with lipstick	0.1683
	pointy face	0.2078
	sideburns	0.1694
	with makeup	0.1708

(b) black hair

	StyleCLIP	Ours
<i>indicator</i> (↓)	0.3870	0.1266
$\Delta d'_c$	0.3451	0.2384
$\Delta d'_e$	short hair	0.1568
	with lipstick	0.1249
	smiling	0.1030
	round eyes	0.1418
	with makeup	0.1368
	brown hair	0.1927
	brown eyes	0.1316
	with glasses	0.0640
	thin nose	0.1441
	with earrings	0.1399

(d) with bangs

	StyleCLIP	Ours
<i>indicator</i> (↓)	0.2580	0.0900
$\Delta d'_c$	0.4072	0.3190
$\Delta d'_e$	oval face	0.1498
	small nose	0.1566
	narrow eyes	0.1133
	with lipstick	0.1100
	long eyebrows	0.1245
	short hair	0.1004
	with bangs	0.0594
	receding hairline	0.0268
	sideburns	0.0876
	high cheekbones	0.1225

(f) with glasses

Table 2.2: Quantitative comparisons (1/2) of disentangled text-driven image manipulation with StyleCLIP [123], using the evaluation metrics introduced in Sec. 2.4.3. For the *indicator*, lower is better. The text command is indicated under each sub-table. Specifically, we illustrate each individual item in the changes of predicted entangled attributes $\Delta d'_e$.

		StyleCLIP	Ours			StyleCLIP	Ours
<i>indicator</i> (↓)		0.4401	0.1521	<i>indicator</i> (↓)		0.3800	0.1418
$\Delta d'_c$		0.5371	0.263	$\Delta d'_c$		0.5579	0.2452
$\Delta d'_e$	green eyes	0.1867	0.0270	$\Delta d'_e$	open mouth	0.262	0.0376
	narrow eyes	0.3378	0.0672		oval face	0.2924	0.0442
	dark eyebrows	0.1920	0.0251		round eyebrows	0.1990	0.0563
	with lipstick	0.1914	0.0574		big nose	0.2588	0.0243
	long nose	0.2805	0.0494		big mouth	0.2839	0.0485
	high cheekbones	0.1708	0.0322		with lipstick	0.1156	0.0144
	oval face	0.3418	-0.0489		sideburns	0.1311	-0.0122
	with makeup	0.2076	0.0275		rosy cheeks	0.1566	0.0444
	blond hair	0.1586	0.0181		closed eyes	0.2248	0.0466
	rosy cheeks	0.2968	0.0472		bald	0.1972	0.0218
(g) pale				(h) double chin			
		StyleCLIP	Ours			StyleCLIP	Ours
<i>indicator</i> (↓)		0.3069	0.1491	<i>indicator</i> (↓)		0.4002	0.1881
$\Delta d'_c$		0.4904	0.3149	$\Delta d'_c$		0.3585	0.1425
$\Delta d'_e$	arched eyebrows	0.1077	0.0270	$\Delta d'_e$	with lipstick	0.1783	0.0364
	closed mouth	0.2105	0.0814		round eyes	0.1541	0.0275
	with makeup	0.2154	0.1035		with makeup	0.1909	0.0420
	green eyes	0.0467	0.0134		thick nose	0.1836	0.0418
	high cheekbones	0.1517	0.0526		round face	0.1222	0.0157
	oval face	0.1933	0.0482		rosy cheeks	0.1404	0.0121
	pinched nose	0.1070	-0.0027		with earrings	0.1044	-0.0278
	white skin	0.1668	0.0371		double chin	0.1633	0.0410
	big eyes	0.1157	0.0424		blond hair	0.1112	0.0243
	rosy cheeks	0.1900	0.0611		with bangs	0.0865	0.0175
(i) with lipstick				(j) arched eyebrows			
		StyleCLIP	Ours			StyleCLIP	Ours
<i>indicator</i> (↓)		0.4220	0.2163	<i>indicator</i> (↓)		0.3703	0.2917
$\Delta d'_c$		0.4880	0.2480	$\Delta d'_c$		0.4575	0.0712
$\Delta d'_e$	wide eyes	0.3635	0.1212	$\Delta d'_e$	arched eyebrows	0.1425	0.0193
	with makeup	0.2127	0.0677		short hair	0.1369	0.0183
	bags under eyes	0.2702	0.1005		with makeup	0.2054	0.0370
	with lipstick	0.1587	0.0350		high cheekbones	0.1553	0.0188
	rosy cheeks	0.2317	0.0375		with lipstick	0.1531	0.0210
	blond hair	0.1263	0.0321		green eyes	0.1338	0.0115
	round face	0.1993	-0.0518		round face	0.1965	-0.0233
	pinched nose	0.1786	0.0374		with bangs	0.0992	0.0142
	white skin	0.2236	0.0432		round eyes	0.1899	0.0203
	long hair	0.0949	0.0109		with makeup	0.2035	0.0240
(k) blue eyes				(l) with earrings			

Table 2.2: Quantitative comparisons (2/2) of disentangled text-driven image manipulation with StyleCLIP [123], using the evaluation metrics introduced in Sec. 2.4.3. For the *indicator*, lower is better. The text command is indicated under each sub-table. Specifically, we illustrate each individual item in the changes of predicted entangled attributes $\Delta d'_e$.

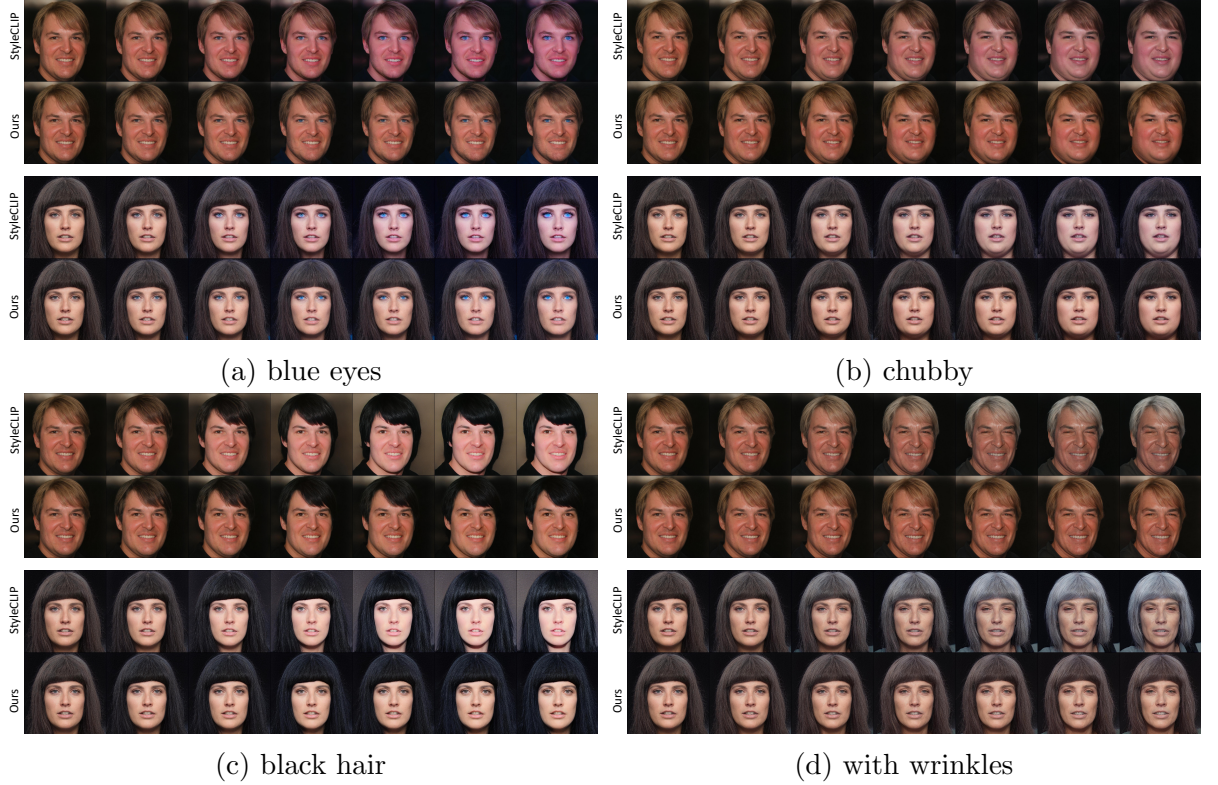


Figure 2.6: Image manipulation results from StyleCLIP [123] and ours, using gradually increasing manipulation strengths. Our method presents better fore-and-aft consistency along the change of manipulation strength.

when $\Delta d'_c$ reaches 0.4880, the $\Delta d'_e$ for “wide eyes” is 0.3635. 2) Our entanglement loss prevents the entanglements in image manipulation effectively. For each text command in the experiment, the *indicator* of “Ours” is significantly lower than that of “StyleCLIP”, indicating that we achieve more disentangled image manipulation. Changes on previously entangled attributes are greatly diminished (as in $\Delta d'_e$) while the manipulation effect is not affected much (according to $\Delta d'_c$). In qualitative results, we illustrate that our method can achieve comparable manipulation effect with StyleCLIP when increases the manipulation strength.

2.5.3 Qualitative Results

Direct Manipulation Outputs. We firstly compare the directly outputted manipulation results from the trained models, without changing the manipulation strengths. In Fig. 2.5, we illustrate the comparing qualitative results on multiple text commands. As can be seen in the manipulation results of “StyleCLIP”, it not only manipulates the required attributes, but also manipulates other attributes. Take text command “*grey hair*” as an example, the manipulated face gets grey hair, while it gets whiter skin and grey eyes simultaneously. Similarly, for the text command “*with wrinkles*”, the manipulated face gets wrinkles, grey hair, and more closed eyes in the meanwhile. Other manipulation results are obtained in similar conditions.

By contrast, “Ours” achieves more ideal manipulation results, where almost only the desired attribute is manipulated while other attributes of are well preserved. For example, for “*wavy hair*”, “Ours” hair becomes wavy while the hair length is close to the original one and the skin color does not become whiter; for “*double chin*”, “Ours” gets double chin while the eye color remains light brown, skin color is kept well, and mouth does not open much. In addition, it is worth mentioning that the qualitative results are quite consistent with the quantitative results, indicating that the proposed evaluation metrics are effective for the disentangled image manipulation task.

Strength-Adjusted Manipulation Outputs. We further compare the manipulation results with gradually increasing manipulation strength. To illustrate, we show four groups of comparing results in Fig. 2.6. In each group, we present the manipulation results for male and female, respectively. We observe that our method learns more disentangled manipulation directions compared to StyleCLIP. For StyleCLIP, when the manipulation strength increases, the desired attribute becomes more and more obvious, as well

as the entangled attributes. As the male-case in Fig. 2.6a, from left to right, the eyes become increasingly blue while they also become wider, the face becomes whiter, and the hair color becomes lighter. Contrarily, our method presents better manipulation consistency. When the manipulation strength increases, the target attribute gradually turns apparent while others remain almost unchanged.

Hyper-Parameters. In the previous sections, we illustrate the ability of our method to achieve disentangled image manipulation without human trials-and-errors. To further study the effects of hyper-parameters, we tune the coefficient of the proposed entanglement loss λ_E in Eq. (2.8) and the number of constraining attributes N in Eq. (2.7). We show comparing results on “blue eyes” and “with earrings”, which are found to be more entangled according to previous experimental results. As in Fig. 2.7a, when λ_E increases, the manipulation effects become less conspicuous while other attributes remain better. However, the manipulation effect can be enlarged by increasing the manipulation strength afterwards. As in Fig. 2.7b, when N varies, there are no obvious differences between the manipulation results. To conclude, our method is not sensitive to hyper-parameters.

2.6 Discussions and Conclusion

2.6.1 Discussions

Generality. We extend our method to the multiple attribute manipulations by handling them as multiple single-command, predicting entangled attributes for each one, and using Entanglement Loss based on the grouped predicted entangled attributes. The experimental results are illustrated in Fig. 2.8. As shown, our method can be generalized to the multiple attribute manipulation setting.

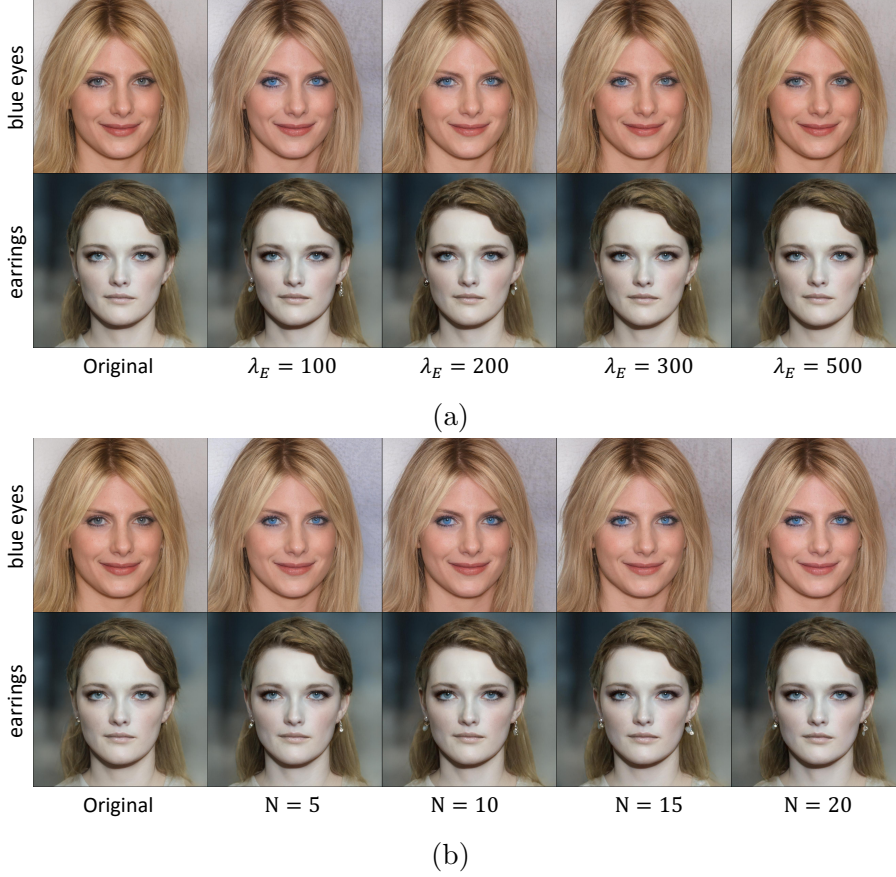


Figure 2.7: Hyper-parameters study of PPE.

Ethical Impact. One common issue in the image manipulation model is that it is biased toward the dataset the model was trained on. For example, BlendGAN [99] indicates that the ethical biases of a dataset may transfer to their model, *e.g.*, the model outputs faces with lighter skin while the input faces are darker-skinned. Our work can help reduce this ethical impact, as our method aims at disentangled image manipulation that only changes the desired attribute while letting the others unchanged, *e.g.*, our method can change the eye color while maintaining the skin color well.



Figure 2.8: Multiple attribute manipulations with PPE.

2.6.2 Limitations

The limitations of the proposed PPE method are as follows: 1) Similar to StyleCLIP, the command out of the domain of CLIP and StyleGAN may not obtain ideal manipulation results. 2) The disentanglement extent in the manipulation results depends on the disentanglement extent in the latent space of StyleGAN. Since we study latent space image manipulation, the best our method can do is to find the most disentangled latent path in the latent space of pre-trained generator. If the attributes are originally entangled for the generator, PPE is unable to achieve completely disentangled manipulations.

2.6.3 Conclusion

In this work, we propose Predict, Prevent, and Evaluate (PPE) to achieve disentangled image manipulation with little manual effort by deeply exploiting the powerful large-scale pre-trained vision-language model CLIP. CLIP is leveraged to 1) **Predict** the entangled attributes given textual manipulation command, 2) **Prevent** the model from finding entangled manipulating latent directions through a novel entanglement loss, and 3) establish a new evaluation metric that can simultaneously **Evaluate** the effects of manipulation and entanglement. PPE is tested on the challenging face editing task and is proven effective.

This chapter explored how predictive language modeling can enhance the controllability of image manipulation by anticipating and preventing semantic entanglement. While the focus here was on disentangling attribute-level modifications guided by natural language, the next chapter turns to a different aspect of language-driven control: using language as a means to guide abstract, conceptual style transfer in the latent space of generative models.

Chapter 3

Language-Guided Image Style Transfer in Generative Latent Space¹

3.1 Problem Statement

In the last few years, a lot of work has focused on the style transfer task using a *reference image* as a representative of the target style, where the goal is to transfer the style of the reference to a content image [52, 63, 120, 17, 96, 32, 31, 162]. Recent improvements in this field include: reducing the artifacts [15, 104, 23, 162], modeling the style-content relationship [120, 177, 101], increasing the generation diversity [93, 152, 183] and many others. However, art styles are usually abstract concepts, e.g., pop art, fauvism, and the style of Van Gogh. To transfer these abstract art styles to a content image, the low-level features (e.g., textures and colors), which are commonly extracted from a single reference image, are not enough. A possible solution is to collect a set of reference images that can be used, e.g., to train a Generative Adversarial Network (GAN) for artist-specific style transfer [140, 186, 78]. The disadvantage of this *set-based representation* is the effort required to collect sufficiently

¹This chapter is based on our published work: Xu et al., "StylerDALLE: Language-Guided Style Transfer Using a Vector-Quantized Tokenizer of a Large-Scale Generative Model", ICCV 2023.

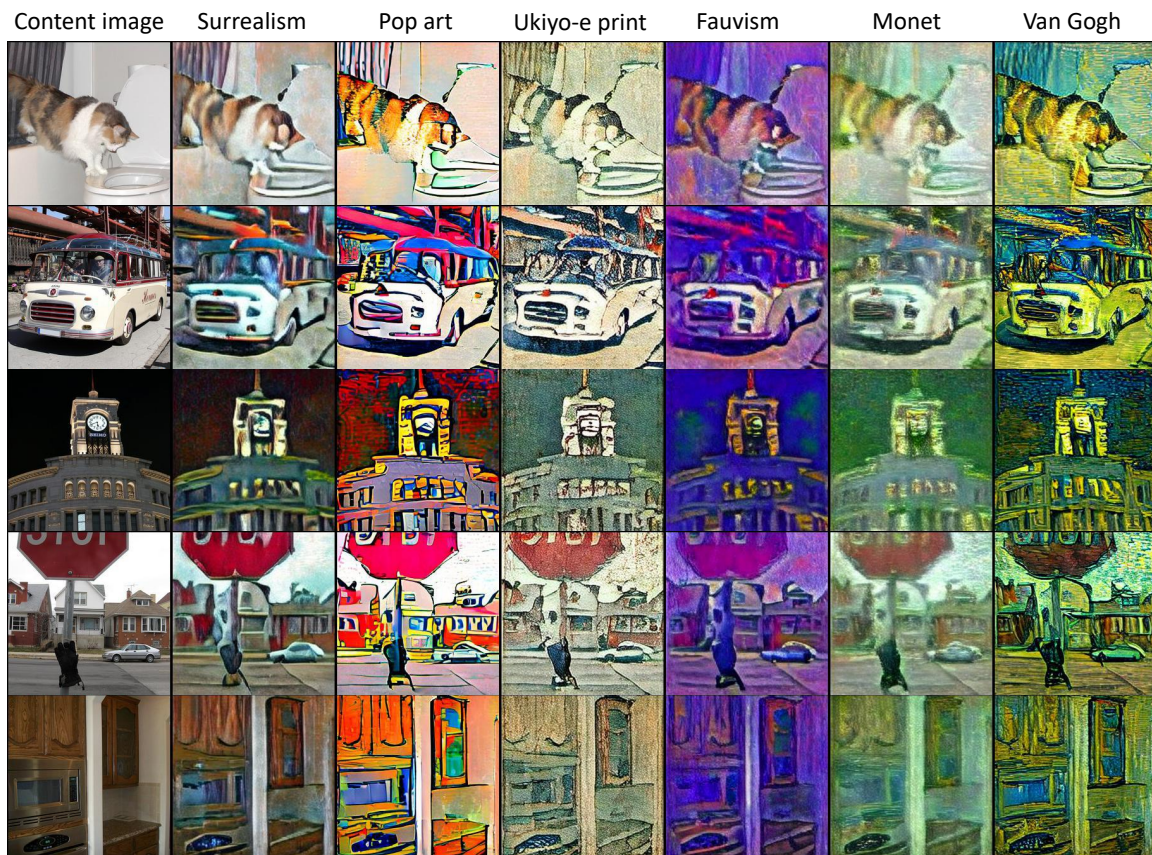


Figure 3.1: With a large-scale pretrained vector-quantized tokenizer (e.g., the dVAE of DALL-E) and CLIP, *StylerDALLE* can transfer various types of styles (indicated on the top), from abstract art styles to specific artist styles and more.

large style-specific data for training.

Recently, large-scale image generative models [131, 130, 139, 179] have shown their power to generate high-quality images of various types, e.g., realistic photo, cartoon, ukiyo-e print, or painting of a specific artist. Moreover, CLIP [128], which is trained with 400 million text-image pairs, learns good joint knowledge between language and vision. Can we leverage the power of these large-scale models for style transfer? In this work, we propose **StylerDALLE**, a language-guided style transfer method that uses both the vector-quantized tokenizer of large-scale image generative methods and CLIP. There are three advantages of our method. Firstly, as compared to images, language is a more natural and efficient way to describe abstract art styles and enables more flexibility. Language can directly indicate art styles at different levels, e.g., “*Van Gogh style oil painting*” and “*Van Gogh starry night style oil painting*”. Secondly, using CLIP and the language description of style saves the effort of collecting style images, as CLIP already learns style-related knowledge and thus can be used to provide supervision. Thirdly, using a large-scale pretrained vector-quantized tokenizer potentially enables style transfer results to be closer to real artworks, as the model is trained from an enormous number of real-world images. In concrete, we propose a Non-Autoregressive Transformer (NAT) model [56], which translates tokens of a content image to tokens of a stylized image in the discrete latent space of a pretrained visual tokenizer, and a two-stage training paradigm.

First of all, since both the input content image and output stylized image can be represented by a sequence of tokens via the large-scale pretrained vector-quantized tokenizer, we formulate the language-guided style transfer task as a token sequence translation. Specifically, we design a NAT model that learns to translate a token-based representation of a low-resolution image into a full-resolution representation, where the final token

sequence contains appearance details specialized for the target style. The most important advantage of using NAT in image generation, with respect to the more common autoregressive transformer (AT) based generation, is that NAT is much faster than AT at inference time, as it allows a *parallel* token generation while AT generates in a token-by-token way.

Secondly, we propose to use CLIP-based language supervision to ensure stylization and content preservation simultaneously, saving the effort to collect data and design dedicated losses. A similar solution that uses CLIP to do style transfer is CLIPStyler [79]. Since merely maximizing the CLIP similarity with respect to a textual style description is not enough for a style transfer task, which should also *preserve* the content of the source image, CLIPstyler introduces a hybrid of losses, including a content loss measured in an external pre-trained VGG network. Conversely, we propose an alternative direction, which avoids the need to tune the coefficients of different loss functions so as to keep the balance between style and content. Specifically, we introduce a two-stage training paradigm: 1) a self-supervised pretraining stage, where the model learns to add semantically coherent image details from a low-resolution image to a high-resolution image; and 2) a style-specific fine-tuning stage, where the model learns to incorporate style into the high-resolution image. Since the fine-tuning phase is built on top of the first stage, the translator is able to keep the semantic consistency with respect to the input image as learned during pretraining. Moreover, we create a textual prompt by concatenating both the style and the textual description of the image content (i.e., its caption). This way, the prompt simultaneously models both the target appearance (i.e., the style description) and the image content which should be preserved by the translation process. Finally, since our translator’s output is discrete and there is no ground-truth tokens for a stylized image, we introduce a Reinforcement Learning (RL) approach to fine-tune the translator using

a reward based on the CLIP similarity between the stylized image and textual prompt, enabling the model to explore the answers in the latent space of a pretrained vector-quantized tokenizer.

We call our network **StylerDALLE** and we show that it can generate stylized images driven by different types of language guidance. Compared with previous language-guided and reference image-based transfer methods, our generated images are less inclined to produce artifacts or semantic errors. Moreover, they can capture abstract concepts related to the target style (e.g., the typical brushstrokes of the artist) besides low-level features like texture and colors. We illustrate the effectiveness of our method through qualitative results, quantitative results, and a user study.

To conclude, our main contributions are:

- We propose **StylerDALLE**, a language-guided style transfer method that manipulates the discrete latent space of a pretrained vector-quantized tokenizer using a token sequence translation approach.
- We propose a non-autoregressive translation network that translates a low-resolution content image into a full-resolution image with style-specific details.
- We propose a two-stage training procedure, including an RL strategy to ensure stylization and content preservation using CLIP-based language supervision.
- Experimental results show that **StylerDALLE** can effectively transfer abstract style concepts, going beyond simple texture and color features while simultaneously preserving the semantic content of the translated scene.

3.2 Related Work

Reference Image-Based Style Transfer. Gatys et al. [52] propose a neural style transfer method in which a pre-trained CNN is used to extract content and style information from images, and to transfer the latter from an image to another. Following this pioneering work, a lot of interests have been attracted, with different methods focusing on different aspects of the topic, such as, e.g., diversified style transfer [152, 163], or attention mechanisms to fuse style and content [176, 120, 101]. A specific line of work focuses on artistic style transfer. For instance, Chen et al. [17] propose to use internal-external learning and contrastive learning with GANs to bridge the gap between human artworks and AI-created artworks. Wang et al. [162] introduce an aesthetic discriminator trained with a large corpus of human-created artworks. Other works train GANs using an artist-specific collection of images [140, 78, 16]. In contrast, we use the generic visual-language semantics embedded in the large-scale pretrained vector-quantized tokenizer and CLIP to avoid collecting style or artist-specific datasets.

Language-Guided Style Transfer. Very recently a few works have proposed transferring methods conditioned on a textual description of the style. For instance, Fu et al. [46] use contrastive learning to train a GAN for artistic style transfer, but they adopt descriptive language instructions rather than more abstract style concepts. Gal et al. [50] use the CLIP space for a domain adaptation of a pre-trained StyleGAN [73]. The method closest to our approach is CLIPStyler [79], where a patch-wise CLIP loss is used to train a U-Net [138]. However, to condition the style change while preserving the image content, CLIPStyler uses hybrid losses and a rejection threshold, introducing many hyperparameters. In our method, we only use

CLIP-based language supervision to ensure the style and content, saving the efforts of designing losses and tuning hyperparameters.

Large-scale Text-to-Image Generation Models. Recently, text-to-image models trained with large or very large scale datasets [131, 130, 139, 179, 14] have attracted tremendous attention because of their excellent performance in generating high-quality images starting from a textual query. Inspired by their strong ability to synthesize various types of images, we study how to use large-scale text-to-image models for style transfer. Specifically, we focus on transformer-based methods, which are potentially less time-consuming as compared to diffusion model-based methods [108]. In addition, we propose to use a non-autoregressive transformer that can generate tokens in parallel.

3.3 Background

Vector-Quantized Image Tokenizer. Transformer-based text-to-image generative models [131, 34, 35, 179, 14] rely on a vector-quantized image tokenizer to produce a discrete representation of the images, e.g., DALL-E [131] has a dVAE [132] and PARTI [179] has a ViT-VQGAN. Despite the differences in implementing the tokenizers, the effects are the same. In more detail, an image I is transformed into a $k \times k$ grid of tokens $X = \{x_{i,j}\}_{i,j=1,\dots,k}$, using an encoder $E(\cdot)$. Each token $x_{i,j} \in X = E(I)$ is an index of a codebook of embeddings ($C = \{\mathbf{e}_1, \dots, \mathbf{e}_M\}$, $1 \leq x_{i,j} \leq M$), built during the training, and corresponds to a patch in I . A decoder $G(\cdot)$ takes as input a grid of embeddings and reconstructs the original image: $\hat{I} = G(\{\mathbf{e}_{x_{i,j}}\}_{i,j=1,\dots,k})$. Training in the transformer-based text-to-image generative models consists of two stages. The first stage is dedicated to training the image tokenizer, while in the second phase, a transformer is

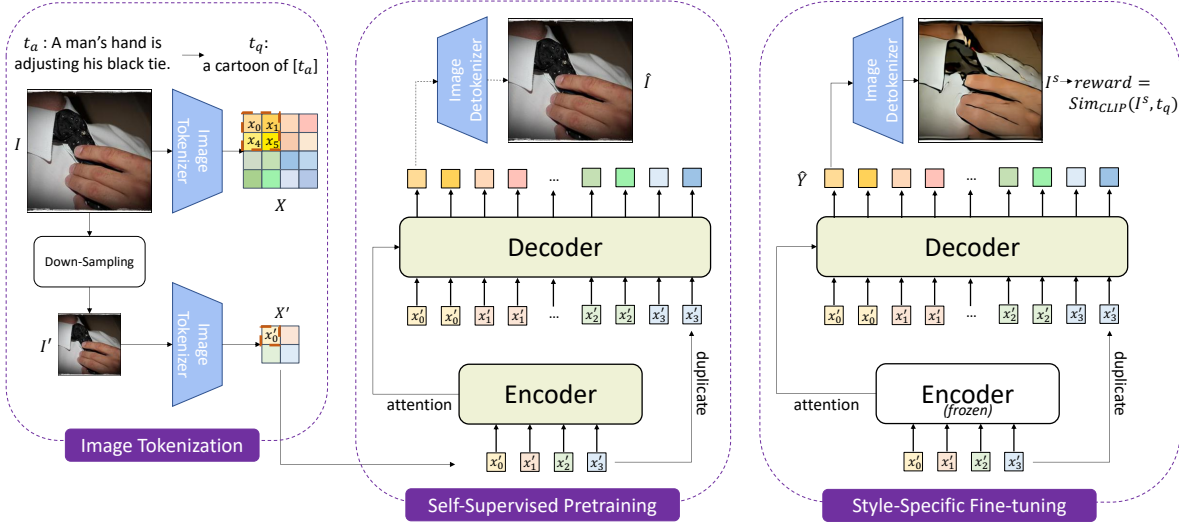


Figure 3.2: A schematic illustration of **StylerDALLE**. We propose a NAT to translate tokens of a half-resolution *content* image to tokens of a full-resolution *stylized* image. **StylerDALLE** consists of three parts: 1) image tokenization preprocessing, to obtain the discrete visual tokens from a pretrained image tokenizer; 2) self-supervised pretraining, to train the NAT to predict full-resolution content images from half-resolution content images; and 3) style-specific fine-tuning, to let the model adding style details in the up-scaling prediction via Reinforcement Learning with language-prompted CLIP guidance. Specifically, in our NAT, the decoder input is duplicated from the tokens of the half-resolution image, following the natural positional correspondence (as highlighted by orange rectangles in the left part) with the full-resolution image.

used to learn a prior distribution over the text and the image tokens. In **StylerDALLE** (Sec. 3.4) we only use the pretrained image tokenizer.

Non-Autoregressive Transformer (NAT). Gu et al. [56] propose a NAT for natural language translation, which consists of an encoder and a decoder. The encoder takes as input a source sentence $X = \{x_1, \dots, x_{N'}\}$ of N' tokens and outputs a distribution over possible output sentences $Y = \{y_1, \dots, y_N\}$, where Y is the translation of X in the target language and, usually, $N \neq N'$. The main novelty of NAT with respect to AT is that, during training, NAT uses a *conditional independent* factorization for the target sentence

and the following log-likelihood:

$$\mathcal{L}(\theta) = \sum_{n=1}^N \log p(y_n | x_{1:N'}; \theta), \quad (3.1)$$

which differs from the common AT factorization in which the prediction of the n -th token (y_n) depends on the previously predicted tokens ($p(y_n | y_{0:n-1}, x_{1:N'}; \theta)$). This conditional independence assumption makes it possible a *parallel* generation of Y at inference time, largely accelerating the translation time with respect to AT models. Importantly, to make parallel generation possible, the encoder input (X) is provided as input to the decoder as well, and individual tokens ($x_n \in X$) can be copied zero or more times, with the number of times each input is copied depending on a specific predicted “fertility” value. As we will see in Sec. 3.4.1, we do not need to predict fertilities because, in our case, the cardinality of the input copies is fixed and determined by the up-scaling task we use for the image translation.

3.4 Method

The language-guided style transfer task can be described as follows. Given an image I , we want to generate a new image I^s which preserves the semantic content of I but changes its appearance according to a textual style description t_s (e.g., “cartoon”). In StylerDALLE, we formulate this task as a *visual-token based translation*, from tokens of a content image to tokens of a stylized image. Specifically, given a content image I we first downsample I to get a half-resolution image I' . Then, I' is fed to the tokenization encoder (Sec. 3.3) which extracts a discrete grid of $k \times k$ source tokens $X' = E(I')$. X' can now be “translated” into a target (discrete) representation $\hat{Y}$, where $\hat{Y}$ is a grid at the original resolution ($2k \times 2k$)

and $\hat{Y} = f(X')$, being $f(\cdot)$ the translation function. Finally, $\hat{Y}$ is fed into the tokenization decoder obtaining the stylized image $I^s = G(\hat{Y})$. In the following subsections, we describe the architecture of $f(\cdot)$ and the way in which it is trained.

3.4.1 Architecture and self-supervised pre-training

For our translation network $f(\cdot)$ we use a NAT architecture [56] (Sec. 3.3) which we train from scratch using a self-supervised learning pretext task consisting in predicting the image at full resolution. Specifically, given the downsampled image I' and its corresponding grid of tokens $X' = E(I')$, we use the indexes in X' to extract the corresponding embeddings from C (Sec. 3.3). For each $x_{i,j} \in X'$, let $\mathbf{e}_{x_{i,j}}$ be the corresponding embedding in C and let $N' = k^2$. The so obtained set of embeddings is flattened into a sequence, and, for each element $\mathbf{e}_n$ of the sequence ($1 \leq n \leq N'$), we add an absolute positional embedding [153] $\mathbf{p}_n$, where $\mathbf{p}_n$ has the same dimension as $\mathbf{e}_n$: $\mathbf{v}_n = \mathbf{e}_n + \mathbf{p}_n$. The final sequence $V' = \{\mathbf{v}_1, \dots, \mathbf{v}_n, \dots, \mathbf{v}_{N'}\}$ is input to the encoder of $f(\cdot)$. Note that an alternative solution is to directly feed $f(\cdot)$ with (a flattened version of) X' and let $f(\cdot)$ learn its own initial token embedding. However, using the embeddings in C has the advantage of exploiting the image representation of the pre-trained image tokenizer. Moreover, from the original image I we extract the ground truth $X = E(I)$, which is flattened in a sequence of $N = 4k^2$ tokens.

Finally, following [56], we build a second sequence of input embeddings V , with cardinality N , which is fed to the decoder of $f(\cdot)$. As mentioned in Sec. 3.3, differently from NAT, we do not predict fertility values. Instead, we get the input of the decoder by simply replicating each element $x_{i,j} \in X'$ according to the positional correspondences between the low-resolution image and high-resolution image (as in Fig. 3.2). The rationale behind this choice is that $f(\cdot)$ is trained to predict the full-resolution image,

and each encoder input ($\mathbf{e}_{x_{i,j}}$) corresponds to a patch in the subsampled image I' and to 4 patches in the full-resolution image I . Thus, initializing the decoder with 4 replicas of each source-image patch initial embedding provides a coarse-grained signal for the upsampling task. Similar to before, the embeddings extracted from C are then flattened and added with a new positional encoding.

Both the encoder and the decoder have self-attention layers and no causal masking is used. However, following [56], in the decoder, we mask out each query position (n) only from attending to itself. Using V' and V , $f(\cdot)$ generates N *parallel* posterior distributions over the visual vocabulary ($\{1, \dots, M\}$): $P = f_\theta(V', V)$, where P is a $N \times M$ matrix, $P_n \in [0, 1]^M$ and $P_n[y] = p_\theta(Y_n = y|V', V)$. Using $Y = \{y_1, \dots, y_n, \dots, y_N\}$, $f(\cdot)$ is trained to maximize:

$$\mathcal{L}_{pre-train}(\theta) = \sum_{n=1}^N \log P_n[y_n]. \quad (3.2)$$

This pre-training stage is independent of the target style and it can be shared over different styles. After this stage, $f(\cdot)$ is able to generate realistic low-level details (which are missing in I'). In the next, we describe how a specific style is incorporated in $f(\cdot)$ using a fine-tuning phase.

3.4.2 Style-specific fine-tuning

Given a style description provided with a textual sentence t_s , the goal is to fine-tune the pre-trained translator $f(\cdot)$ (Sec. 3.4.1) to make it generate image details in the style of t_s . We fine-tune only the decoder of $f(\cdot)$, keeping frozen the encoder. To ensure both stylization and content preservation, we design a prompt that consists of two parts, i.e., a style description t_s and an image caption t_a , which describe the image *content* and can be obtained from a generic image captioning dataset. For instance, given $t_a =$

“A man’s hand is adjusting his black tie” and $t_s = \text{“cartoon”}$, we obtain $t_q = \text{“a cartoon of a man’s hand is adjusting his black tie.”}$ On the other hand, in order to represent the image generated by $f(\cdot)$, we first need to sample the distributions in P (Sec. 3.4.1), and we do so using multinomial sampling:

$$\hat{Y}_n = \text{Sampling}(P_n[y]), \quad \forall n \in \{1, \dots, N\}. \quad (3.3)$$

The sampled sequence $\hat{Y}$ is reshaped to a $2k \times 2k$ grid and fed to the image detokenizer to get the final image $I^s = G(\hat{Y})$. Finally, using the CLIP visual and textual encoders we compute the cosine similarity on the CLIP space:

$$r = \text{Sim}_{CLIP}(I^s, t_q). \quad (3.4)$$

However, directly using Eq. 3.4 as the fine-tuning objective function is not possible because Eq. 3.3 is not differentiable. In addition, since there is no ground truth for the tokens of stylized images we use RL to encourage the model to explore the answers in the latent space of a pretrained vector-quantized model. We use the REINFORCE algorithm [164] that updates the parameters of $f_\theta(\cdot)$ using the CLIP-based reward r , which could keep awarding the model for achieving better-stylized results until the model reaches a limit. This leads to the gradient estimate:

$$\nabla_{\theta|_d} \mathcal{L}_{fine-tune}(\theta|_d) = \sum_{n=1}^N r \nabla_{\theta|_d} \log P_n[y_n], \quad (3.5)$$

where $\theta|_d$ indicates the parameters of the decoder only (we found fine-tuning both could lead to content loss). By maximizing Eq. 3.5 we encourage $f(\cdot)$ to generate images having both the content and the style of the prompt t_q .



Figure 3.3: Qualitative results of **StylerDALLE-1** on various styles: (a) Monet Sunrise, (b) Picasso cubism, (c) Van Gogh blue color, (d) Van Gogh purple color, (e) warm and relaxing, (f) ukiyo-e print, (g) fauvism, (h) pixel art illustration. The style references at the top are for illustration only (not used as input to the model).

With respect to the whole method, the reason we use down-sampled versions of the content image is that “style” is commonly assumed to be involved in the low-level visual details, such as colors, texture, painting strokes, etc. X' , which in our formulation represents I at a lower resolution, presumably keeps most of the content in I discarding some details, this way facilitating the style translation process. Preliminary experiments (in Appendix 3.8) in which we fed the encoder with tokenized full-resolution images led to poor results, demonstrating that the different cardinality between the source and the target sequence is an essential component in this translation process.

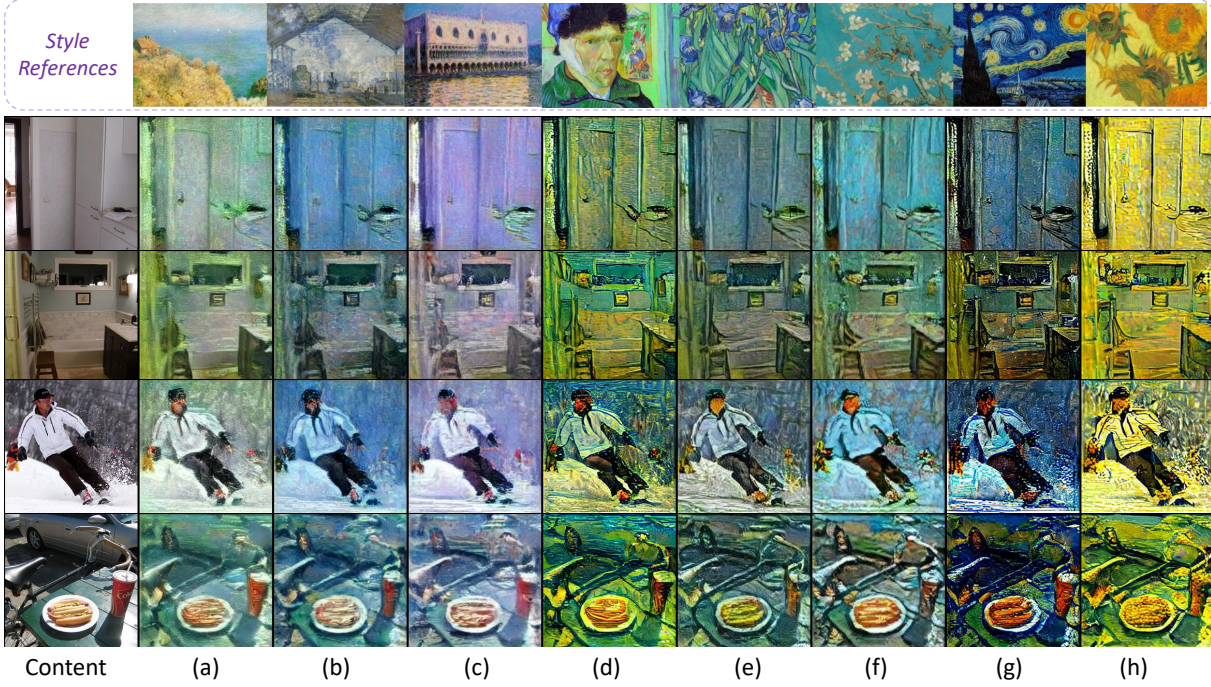


Figure 3.4: Qualitative results of **StylerDALLE-Ru** on various styles: (a) Monet, (b) Monet Paris, (c) Monet Venice, (d) Van Gogh, (e) Van Gogh Irises, (f) Van Gogh Almond, (g) Van Gogh Starry Night, (h) Van Gogh Sunflowers. The style references at the top are for illustration only (not used as input to the model).

3.5 Experiments

3.5.1 Training Details

We implement our method based on two types of pretrained vector-quantized tokenizer: 1) the officially released dVAE of DALL-E 1 [116] and 2) the VQGAN of Ru-DALLE [158], and consequently obtain two groups of results, noting as **StylerDALLE-1** and **StylerDALLE-Ru**, respectively. We use the two models in our experiment because they are open-sourced while our method is applicable to any large-scale pretrained vector-quantized tokenizer.

To train the model, we use the MS-COCO [97] train-set, which contains 83k images of common objects in daily scenes. In the self-supervised pre-

training stage, we only use the images while in the style-specific fine-tuning stage we use both images and captions. The **StylerDALLE** NAT model consists of a 4-layer encoder and an 8-layer decoder while the attention head number is 8 and the hidden dimension is 512. We use Pytorch [121] to implement our method. We train the NAT model for 25 epochs with a learning rate of 1e-4. We use Adam [77] optimizer. In the style-specific fine-tuning stage, we use both the images and the captions. In particular, we utilize all the caption annotations to enhance the model robustness, as usually human-beings annotate different captions of a single image. Notably, the caption annotations are only used at the fine-tuning stage. In other words, **StylerDALLE** does not need to use image caption as input at inference time. We only fine-tune the decoder of the NAT model, and keep the encoder frozen. We use Adam optimizer with a learning rate of 1e-6. For CLIP model, we use the CLIP ViT-B/32 model. For both training stages, the model is trained on a single RTX-A6000 GPU for 24 hours.

For pretrained vector-quantized image tokenizer, we use the officially released dVAE of DALL-E² or the VQGAN of Ru-DALLE³. To compare with CLIPStyler, we use the official implementation.⁴ For all reference image-based comparing methods, we use the officially released trained models.⁵

3.5.2 Experimental Results

In the following, we present qualitative results, quantitative results, and user study results of our method and comparative methods, as well as comparisons with reference image-based methods. We also include ablation studies and inference time comparisons.

²DALL-E: <https://github.com/openai/dall-e>

³Ru-DALLE: <https://github.com/ai-forever/ru-dalle>

⁴CLIPStyler: <https://github.com/cyclomon/CLIPstyler>

⁵AesUST: <https://github.com/EndyWon/AesUST>, StyTr2: <https://github.com/diyiiyiii/StyTr-2>.

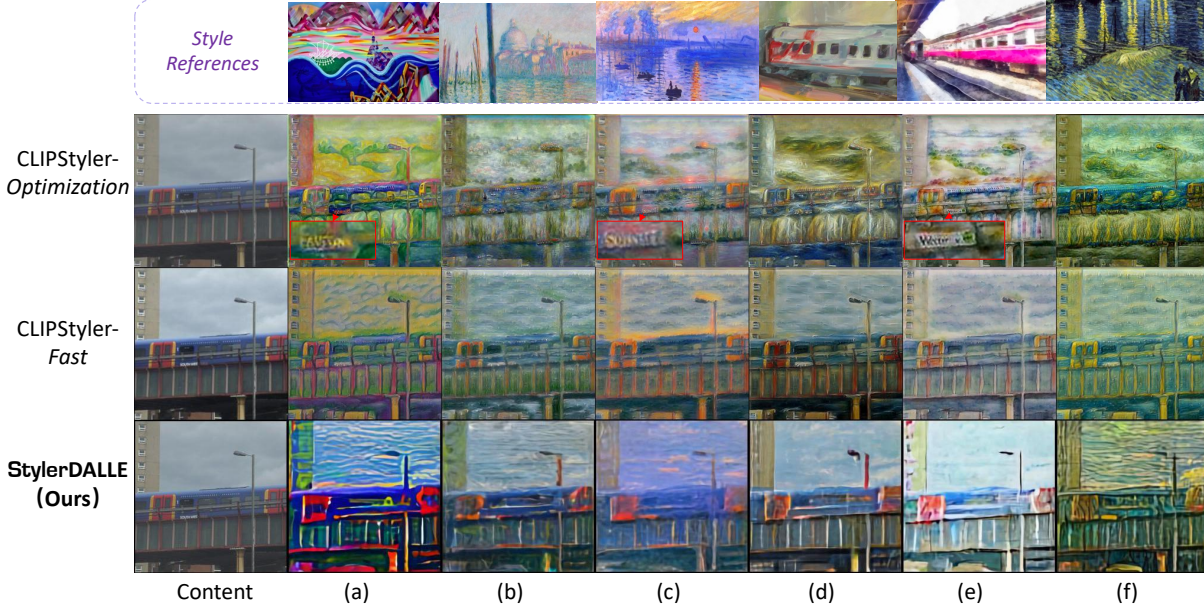


Figure 3.5: Comparisons with language-guided methods. The styles are (a) fauvism, (b) Monet, (c) Monet Impression Sunrise, (d) oil painting, (e) watercolor painting, (f) Van Gogh, (g) Van Gogh blue color. The style references at the top are for illustration only (not used as input to the model).

Qualitative Results. In Fig. 3.1, Fig. 3.3 and Fig. 3.4, we show that our method can effectively transfer various types of styles, i.e., a) abstract art styles, e.g., “fauvism” and “pop art”; b) artist-specific styles, e.g., “Monet” and “Van Gogh”; c) artist-specific styles with additional descriptions, e.g., “Monet Paris” and “Van Gogh Sunflowers”; d) artistic painting types, e.g., “pixel art illustration”; and e) emotional effects, e.g., “warm and relaxing”. According to the qualitative results, we draw the following conclusions: 1) **StylerDALLE** can transfer abstract style concepts which go beyond the texture and color features and are similar to the typical trait of the artist/artistic target style; 2) each style corresponds to generated images that are different from those of other styles; 3) the image content is well preserved; and 4) **StylerDALLE** can be applied to open-domain content images (i.e., the image content can contain animals, human beings, daily objects, buildings, etc.).

Dataset	Style Method	Fauvism	Monet	Monet Sunrise	Monet Sunset	Monet Paris	Van Gogh Irises	Van Gogh Almond Blossoms
COCO	CLIPstyler-Fast	26.11	27.46	24.78	26.70	27.60	26.23	27.05
	Ours	30.61	27.72	28.81	30.68	29.63	28.14	31.64
AFHQ	CLIPstyler-Fast	26.52	26.08	23.27	25.05	25.42	23.28	24.31
	Ours	29.96	26.55	26.58	29.11	26.10	28.04	31.92
ImageNet-100	CLIPstyler-Fast	26.63	27.16	24.33	26.12	27.28	26.32	26.23
	Ours	30.22	27.53	28.29	30.28	27.67	28.26	31.60

Table 3.1: Quantitative comparisons with CLIPstyler [79] on different types of styles and datasets.

Further, we compare **StylerDALLE** with the recent language-guided style transfer method CLIPStyler [79]. CLIPstyler proposes two methods: 1) CLIPStyler-Optimization, which optimizes a style transfer network *for each content image*, thus being time-consuming; and 2) CLIPStyler-Fast, which is the most comparable method to ours as it trains a network *for each style*, and then it can be used with any content images. As shown in Fig. 3.5, CLIPStyler-Optimization generates diverse stylized results but it suffers from inharmonious artifacts. For instance, in the column “Monet”, there are many plants on the train. In addition, CLIPStyler-Optimization has the problems of writing the style text in the results, as in the results of “fauvism” and so on. On the other hand, the images generated by CLIPStyler-Fast do not contain artifacts but there is less variation among different styles. Importantly, it is hard to recognize the typical trait of each artistic style, and the main differences among the styles are the colors. In contrast, the results of **StylerDALLE** are much closer to the artworks of the specific artistic style, they show distinct differences among different styles, and they have no artifact issues.

Quantitative Results. To do quantitative analysis, we use the CLIP similarity score, formalized as $score = Sim_{CLIP}(I, t_s)$, which is computed between generated stylized images and textual description of the target style. In Tab. 3.1, we present the results of **StylerDALLE**-1 and CLIPstyler-Fast.

Since both methods are applicable for arbitrary content images, we use the MS-COCO val-set, AFHQ val-set [25], and ImageNet-100 val-set [30] for evaluation. According to the results on multiple datasets, although the way we use CLIP is to compute rewards in Reinforcement Learning, instead of directly using CLIP-scores to optimize the network as CLIPstyler, we achieve comparable and even better quantitative results, indicating the effectiveness of our method.

User Study. We conduct a user study to see human opinions towards the stylized images coming from different methods: 1) CLIPstyler-Fast, 2) StylerDALLE-1, and 3) StylerDALLE-Ru. In specific, we collect opinions from 35 human subjects with a 30-question questionnaire. In each question, we ask them to select one stylized result that is closest to a target style. As in Tab. 3.2, among the three methods, StylerDALLE-Ru achieves the highest preference score of 68.76%, indicating the superiority of our method. We also find that humans prefer StylerDALLE-Ru much more than StylerDALLE-1. We infer this could be because the results from StylerDALLE-1 are blurry, as they are based on the dVAE of DALLE-1, and humans dislike blurry images.

Other than the quantitative analysis and qualitative analysis, as in Tab. 3.2, we further involve human subjects to evaluate the style transfer results of StylerDALLE and the comparing method CLIPstyler. To help the participants know the styles, at the beginning of evaluating each style, we incorporate several illustrations of the style (Tab. 3.6a). We show part of the questionnaire in Tab. 3.6b. We use Google Forms to collect user opinions.

Comparisons with Reference Image-Based Methods. We compare StylerDALLE with state-of-the-art reference image-based methods:

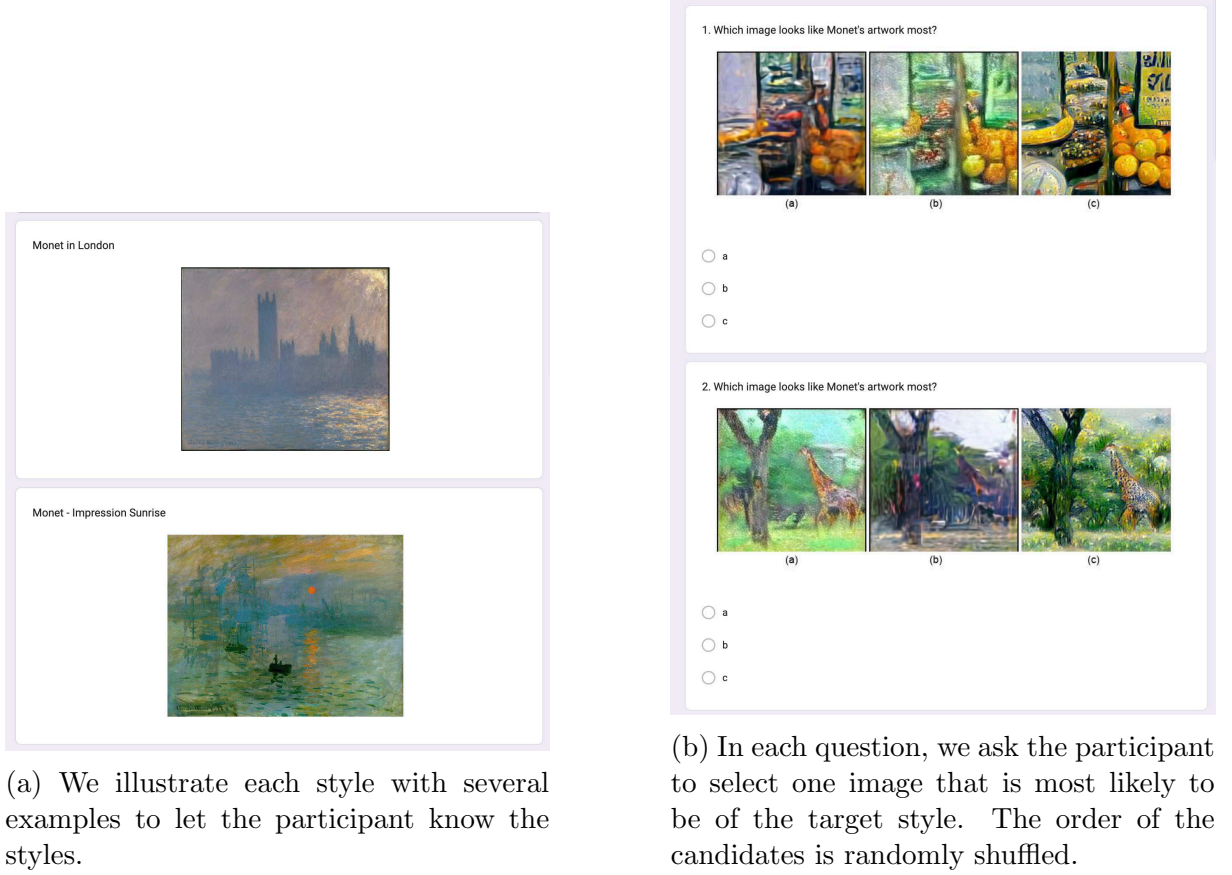


Figure 3.6: Illustrations of the *StylerDALLE* user study details.

1) AesUST [162], an arbitrary style transfer method which enhances the aesthetic reality using a GAN trained with a collection of artworks; and 2) StyTr2 [31], an arbitrary style transfer method which uses a transformer to eliminate the biased content representation issues of CNN-based methods. To make the comparison feasible, we show the results of AesUST and StyleTr2 using two Van Gogh paintings as the reference images, and the results of *StylerDALLE* trained using the corresponding language description.

As shown in Fig. 3.7, the results of *StylerDALLE* are distinctive from the results of reference image-based methods. In concrete, the results of both AesUST and StyleTr2 are mostly affected by the colors and the textures of the reference images, and to some extent in an unnatural way. For

(%)	CLIPstyler-Fast	Styler DALLE-1	Styler DALLE-Ru
Preference	15.90	15.33	68.76

Table 3.2: Preference scores of StylerDALLE user study.

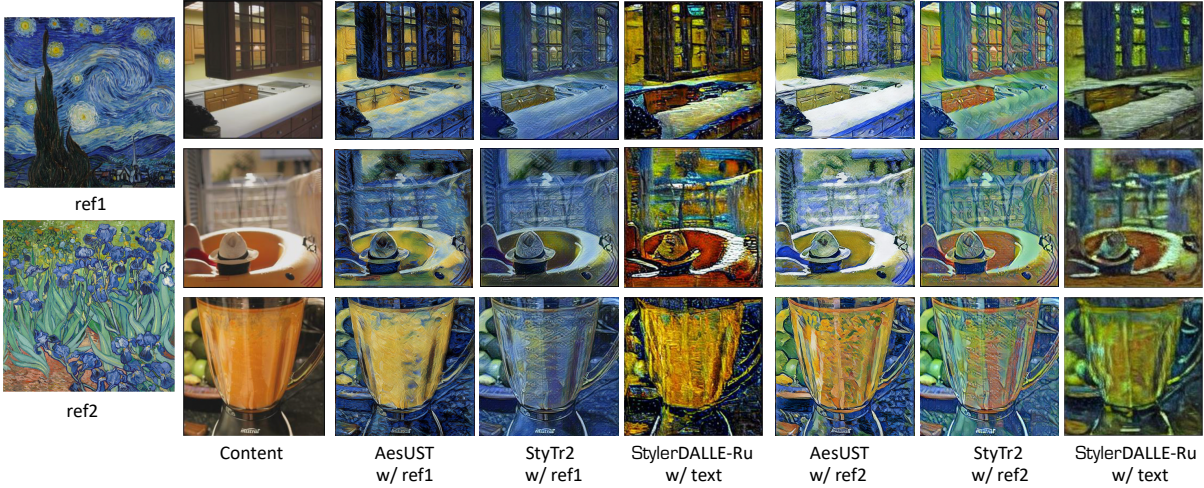


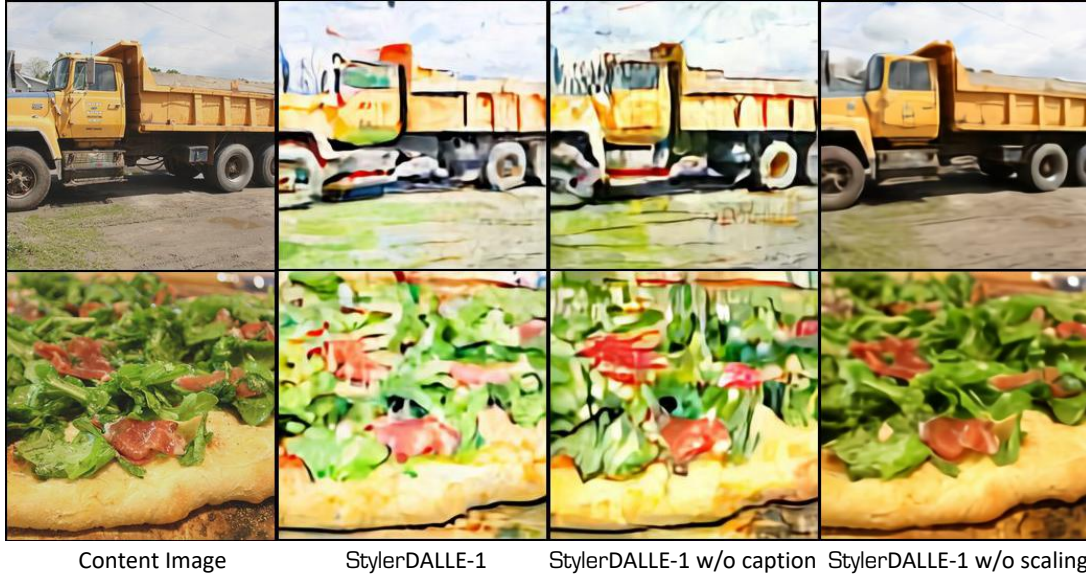
Figure 3.7: Comparisons with reference image-based methods. **StylerDALLE** is able to transfer more abstract concepts, e.g., specific painting strokes, and is less likely to produce semantic errors.

instance, on the bottom line, in the “Van Gogh Irises” stylized results of AesUST and StyleTr2, the textures of the irises get transferred to the orange cup. In transferring the style of “Van Gogh Starry Night”, the objects in the results of AesUST and StyleTr2 are mostly in the same dark-sky blue as the reference image, making the scenes a bit in-realistic. By contrast, the colors and textures used in the reference style are well transferred while being appropriately applied to the contents, without changing the original semantics. Moreover, **StylerDALLE** achieves to transfer higher-level style features, e.g., the brushstroke, rather than merely colors and textures, leading to the results of a similar style to the target one. For example, the strokes in *Starry Night* are sharper as compared to the ones in *Irises*, and the differences are also presented in the stylized results of **StylerDALLE**.

Ablation Study. We study two ablations of *StylerDALLE*: (1) without captions, and (2) without scaling.

Firstly, we ablate the usage of captions in formulating the prompt-based reward during the style-specific fine-tuning stage (Sec. 3.4.2). In more detail, instead of using the CLIP similarity between the stylized image I^s and the prompt t_q (which combines the style description t_s and the image caption t_a) as the reward, we discard t_a and we compute the CLIP similarity between the stylized image I^s and the style description t_s as the reward. As shown in Fig. 3.8, the two models *StylerDALLE*-1 and *StylerDALLE*-Ru show different results on the ablation “w/o captions”. For *StylerDALLE*-1 (Fig. 3.8a), we see that the results of the full model are slightly better. In the results of *StylerDALLE*-1, the details are preserved better, and the colors are closer to the light and muted colors used in watercolor painting. Moreover, the results are overall harmonious as there are few abrupt brushstrokes. Meanwhile, “*StylerDALLE*-1 w/o captions” also presents a satisfying style transfer quality, as the results keep a good balance between the stylization and content maintainness. This indicates our method can also work for the dVAE of DALL-E when no caption is provided, thus being less annotation-dependent. Nevertheless, “*StylerDALLE*-Ru w/o captions” (Fig. 3.8b) fails to keep content consistency, emphasizing the significance of using captions as part of the language supervision in the Reinforcement Learning process for maintaining the content.

Secondly, we ablate the operation of down-sampling as introduced in Sec. 3.4. Specifically, we directly input the discrete tokens of the full-resolution image to the NAT model while conducting the same self-supervised pre-training and style-specific fine-tuning. As shown in the results of “*StylerDALLE*-1 w/o scaling” (Fig. 3.8a) and “*StylerDALLE*-Ru w/o scaling” (Fig. 3.8b), scaling is an important procedure in *StylerDALLE*: when the NAT model is input with the discrete tokens



(a) Ablation study results on **StylerDALLE-1**, which is implemented based on the officially released dVAE of DALL-E [116].



(b) Ablation study results on **StylerDALLE-Ru**, which is implemented based on the VQGAN of Ru-DALLE [158].

Figure 3.8: Ablation study on **StylerDALLE**.

of the full-resolution image, the style cannot be incorporated effectively through the Reinforcement Learning fine-tuning stage.

Inference Time. To generate a single 256×256 stylized image (including the time to down-sample, encode, translate through the NAT, and decode), **StylerDALLE** needs 0.076s, which is the average time computed over the COCO val-set using an RTX-A6000 GPU. The main idea of our work is to use large-scale pretrained image generative models for style transfer and we focus on using vector-quantization-based methods. Therefore, we conclude that as compared to style transfer methods that are based on large-scale diffusion models, **StylerDALLE** has the advantage of having less inference time. For instance, as reported in a recent paper [14], Imagen [139] takes 9.1s to generate a 256×256 image on TPUv4 accelerators.

3.6 Conclusion

In this work, we present **StylerDALLE**, a language-guided style transfer method that leverages the power of the large-scale pretrained vector-quantized image tokenizer and CLIP. Specifically, inspired by natural language translation, we propose a non-autoregressive sequence translation approach to manipulate the discrete visual tokens, from the content image to the stylized image. We use Reinforcement Learning to include CLIP-based language supervision on the style and content. Differently from previous work, **StylerDALLE** can transfer abstract style concepts that are implicitly represented in the pretrained image tokenizer and CLIP and which cannot be easily obtained using reference images. Moreover, using the large-scale pretrained latent space as the basic image representation makes it possible to reduce the artifacts and the semantic incoherence better than the previous work that operates at the pixel level.

Having examined how language can guide abstract style transfer by operating in the discrete latent space of large-scale generative models, we now shift our focus to a more structural challenge in language-guided genera-

tion. The next chapter investigates that artifacts emerge in CLIP-guided image generation and proposes a spectral approach to mitigate these undesired effects, aiming to improve the semantic robustness of generation.

Chapter 4

Spectral Filtering for CLIP-Guided Image Generation¹

4.1 Problem Statement

Style transfer is about transforming the overall appearance of a given *content image* to adhere to a specific style while preserving its content. Starting from the pioneering paper of Gatys *et al.* [52], this task has attracted a growing interest in the scientific community because of its large application interest (*e.g.*, in the e-commerce or the entertainment industry, etc.). While most of the methods proposed so far extract the style information from a *reference image* [52, 63, 120, 17, 96, 32, 31, 162, 15, 104, 23, 176, 101, 152, 163] or a set of reference images [140, 186, 78], very recently, with the emergence of vision-language foundation models, a few approaches have started investigating the use of a *textual description* of the target style [50, 79, 172]. The main idea is to describe the target style with a natural language sentence (*e.g.*, “pop art”) which is used to condition the content image transformation. The main advantage of this approach is that natural language sentences can describe more general and more abstract

¹This chapter is based on our published work: Xu et al., “SpectralCLIP: Preventing Artifacts in Text-Guided Style Transfer from a Spectral Perspective”, WACV 2024.

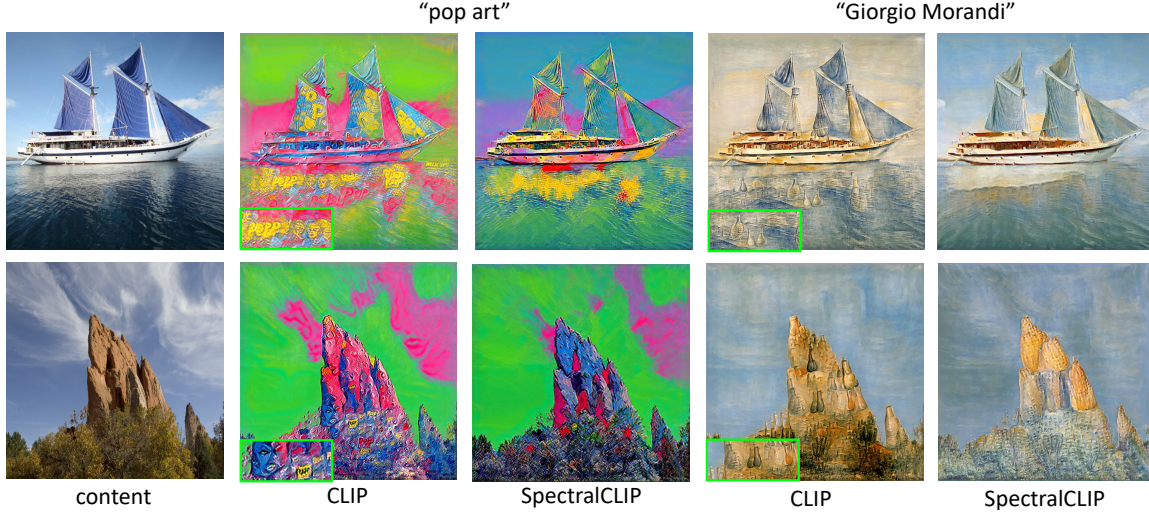


Figure 4.1: Text-guided style transfer results using either a CLIP-based image-text similarity or SpectralCLIP. While using the standard CLIP leads to the generation of undesirable artifacts (written words and unrelated visual entities as *highlighted*), the proposed SpectralCLIP prevents the artifacts while maintaining other style features.

style characteristics that can hardly be extracted from a single reference image. Moreover, this way, it is possible to indirectly exploit the knowledge contained in the vision-language foundation model, which is usually pre-trained using hundreds of millions of image-text pairs.

Kwon *et al.* proposed CLIPstyler [79], which utilises the power of CLIP for text-guided style transfer for arbitrary images, demonstrating a broader range of styles and higher transfer quality than previous work based on reference images. However, as pointed out in [79], this method tends to generate images with over-specific artifacts. To distinguish, we define two types of artifacts: textual and visual artifacts. Visual artifacts are over-specific entities drawn on the generated image. In the example of fig. 4.1, when the style is “pop art”, CLIPstyler adds red lips and faces onto the image. Textual artifacts are written words, typically from the textual prompt describing the desired style, that appear on the generated image in an unwanted manner. Examples can also be seen in fig. 4.1, where the word “pop” is spread over the generated image when CLIPstyler transfers

the image with the style of “pop art”. The presence of this type of artifact is largely due to the entanglement of visual concepts and written texts inherent in CLIP [107]. This entanglement issue in CLIP has been shown to be problematic and prevalent in a variety of CLIP application scenarios, including zero-shot classification and text-guided generation. The study and alleviation of effects resulting from such undesirable entanglement is a significant direction that has attracted increasing research attention [54, 81, 107].

In this paper, we propose to prevent artifact generation in text-guided style transfer using a spectral approach. Spectral analysis has been used to analyze temporal or spatial variations of signals [55]. Recently, researchers in the field of natural language processing (NLP) have proven its effectiveness in capturing linguistic information at different granular levels [147, 109]. Tamkin *et al.* [147] show that the sequence of textual tokens input to a Transformer network [153] contains structures at different scales: *e.g.*, the word scale, the sentence scale, the document scale, etc. These scales correspond to different frequencies in the changes of the values of the neurons’ activations and can be isolated using the coefficients obtained by applying a DCT to the neuron activation sequence [147]. Intuitively, analogously to a sequence of linguistic tokens containing a hierarchy of semantic levels (*i.e.* from word level up to the document level), the constituent patches of an image also contain information at different levels. Similarly to [147], we sort the frequency components into several continuous bands. After analysing the patterns of artifacts present in CLIPstyler-generated stylised images, we find that these artifacts are highly related to certain frequency bands, and that by masking out these frequency components, we can remove the artifacts effectively without hurting the quality of stylised images. Hence, we propose SpectralCLIP to mask out those frequency bands, which implements a spectral filtering layer on top of the

last layer of the CLIP vision encoder. We conduct experiments that verify the following points: (i) we experiment with many types of styles and find SpectralCLIP can effectively reduce both visual and textual artifacts while maintaining the target style well; (ii) we conduct a user study of 30 participants to compare the visual quality in terms of the overall style and the artifact-free performance of generated images, and find that our generated images are preferred by **55.28%** and **74.44%** of the users in terms of overall quality and artifact-free performance, respectively (Sec. 4.4.2); and (iii) we also leverage the ‘learn-to-spell’ CLIP (the CLIP subspace that focuses on written texts in the image) [107] to quantitatively validate that SpectralCLIP efficiently reduces textual artifacts (Sec. 4.4.1) as the score w.r.t. written texts is notably reduced. In addition, we employ SpectralCLIP for text-guided image generation (Sec. 4.4.4) and show it effectively prevents written words on the generated images.

To conclude, the contributions of this paper are:

- We propose SpectralCLIP to prevent both textual and visual artifacts in CLIP-guided style transfer. The effectiveness of SpectralCLIP has been verified on multiple styles through qualitative results, quantitative results, and a user study.
- SpectralCLIP is the first work to use spectral filtering in vision-language models. Other than solving the artifacts issues in CLIP-guided image style transfer, it also gives a new perspective on the disentangling of written texts and visual concepts in the CLIP space.
- To emphasize the generality of SpectralCLIP, we show that it can reduce the textual artifact generation also when used in a non style-transfer task and jointly with a completely different generator based on VQGAN.

4.2 Related Work

Reference Image-Based Style Transfer. After a few initial works on style transfer [41, 60], Gatys et al. [52] propose to use the Gram matrix of a convolutional network to represent the target style extracted from a single reference image. Following this paradigm, multiple aspects have been successively explored, ranging from *e.g.*, arbitrary style transfer [63, 57, 120], diversified style transfer [152, 163], attention mechanisms to fuse style and content [176, 120, 101], reducing artifacts [15, 104, 23, 162], increasing the content persistence [92, 166], and many others. However, it is tricky to *generalize* the description of an abstract style (*e.g.*, “pop art” or “warm and calm”) from a single reference image. For this reason, a line of work is based on using a (large) *set* of reference images of the target style [186, 140, 78, 16]. In contrast, humans can usually understand a style using one or a few words, thanks to their knowledge and the relation they learned between words and visual appearance. Massively trained vision-language models like CLIP [128] now make it possible to emulate this human process, and text-guided style transfer approaches can avoid collecting a dataset for each target style, with a simple textual query.

CLIP-Guided Image Synthesis. The CLIP space has been largely used for image synthesis (*e.g.*, image generation [28, 125, 141] and image manipulation [123, 7, 171, 24]). However, when using CLIP for a text-guided style transfer task, a challenging aspect is to preserve the content while changing the style, since the style textual description usually does not contain any reference to this content. To solve this problem, Gal *et al.* [50] propose a directional loss and fine-tune a StyleGAN [71, 72] pre-trained using images of a specific domain. CLIPstyler [79] extends this approach to an open-domain scenario and uses multiple patches. However, the multi-patch directional loss leads to the generation of textual artifacts

and “over-specification”, where the latter refers to over-specific visual artifacts which locally remind of the textual description of the style [79] (section 4.1). Most of the experiments shown in this paper are based on a CLIPstyler baseline, and we show that using SpectralCLIP for the directional loss computation, we can largely alleviate both the visual and textual artifact problem. Finally, Materzynska *et al.* [107] analyse the text-image entanglement problem in the CLIP space, and learn orthogonal projections (“forget-to-spell” and “learn-to-spell”) of this space to disentangle the two modalities. We empirically show that using the “forget-to-spell” projection on a CLIPstyler baseline, we can indeed reduce the textual artifacts. In contrast, the proposed SpectralCLIP can reduce the generation of both the visual and the textual artifacts.

4.3 Method

SpectralCLIP is based on computing a text-image similarity using a frequency filter of the CLIP representations. In this section, we first describe how this filtering is obtained (Sec. 4.3.1), and then we show how it can be plugged into existing text-based generative approaches (Sec. 4.3.2), and finally how the band filters are selected (Sec. 4.3.3).

4.3.1 Spectral based filtering

Given an image I , we use the CLIP vision encoder $E_v(\cdot)$ to represent I with a grid of $k \times k$ vectors which either are extracted from the last convolutional layer of a ResNet [58] or correspond to the final embeddings of a Vision Transformer [38]. Our method is independent of the specific encoder architecture and can be applied to both types of networks. In the experiments of this paper, following [79], we used a ViT-B/32 [38], pre-trained by the authors of CLIP and then frozen. Since a ViT-based encoder

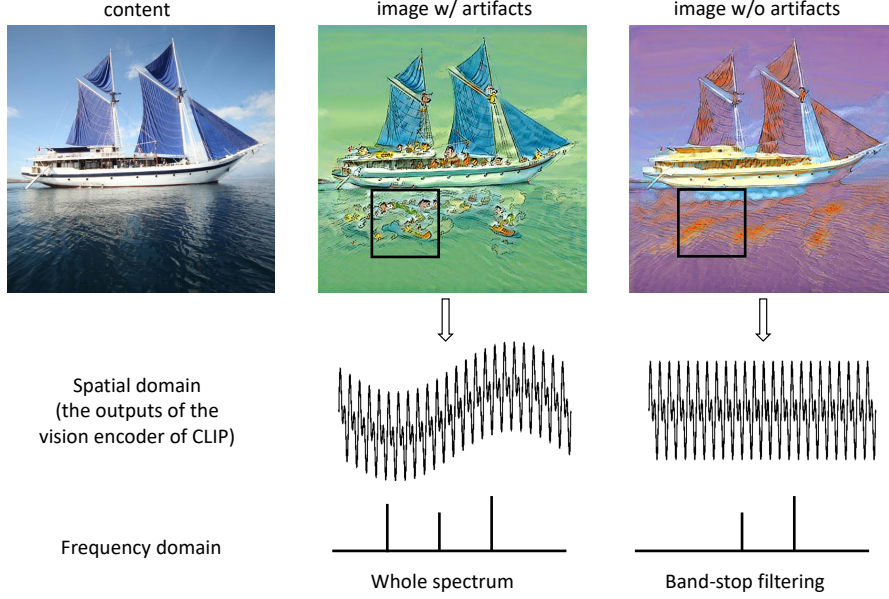


Figure 4.2: An illustration of SpectralCLIP. To transfer a “cartoon” style to the leftmost content image, CLIPstyler generates many cartoon-like artifacts, spreading over the whole image (central figure). The corresponding spectral representation is a composition of frequencies with different periods. Removing the frequencies corresponding to the artifact scales (SpectralCLIP) prevents the generation of these unwanted artifacts (right figure).

also includes a class token, we get $n = 1 + k^2$ vectors, which we flatten into a sequence: $V = E_v(I)$, where $V = \{\mathbf{v}_0, \dots, \mathbf{v}_i, \dots, \mathbf{v}_{n-1}\}$. Despite each $\mathbf{v}_i \in \mathbb{R}^d$ being a d -dimensional vector, following [147], a spectral representation of V can be obtained separately considering each dimension j ($0 \leq j \leq d-1$), which, in our case, corresponds to the j -th channel of the CLIP embedding. For a given j , if $x_i^{(j)} = \mathbf{v}_i[j]$ is the j -th component of $\mathbf{v}_i$, the corresponding (scalar-valued) sequence $X^{(j)} = \{x_0^{(j)}, \dots, x_{n-1}^{(j)}\}$ can be represented in the frequency domain using, *e.g.*, the DCT-II variant of DCT:

$$f_m^{(j)} = \sum_{i=0}^{n-1} x_i^{(j)} \cos \left[\frac{\pi m}{n} (i + 1/2) \right], \quad (4.1)$$

where $m = 0, \dots, n-1$, and $f_m^{(j)}$ is the coefficient of the m -th frequency and represents the contribution of the corresponding cosine wave in the

“signal” discretely sampled by the sequence $X^{(j)}$. Different frequencies describe different cosine waves, and each frequency period (i.e., the number of elements of $X^{(j)}$ it takes to complete a full cycle) corresponds to the scale of the change in the activation of the j -th neuron. Tamkin *et al.*[147] observe that, in the natural language processing domain, these scale changes usually correspond to different structures contained in the input document: *e.g.*, the single word structure corresponds to the highest frequencies, while medium-level frequencies correspond to sentences, etc. Analogously, since, in our domain, artifacts are relatively large-scale visual structures appearing repeatedly throughout the image (section 4.1), we separate those frequencies most likely corresponding to the artifacts from the other frequencies containing useful style information (texture, color, etc.). To do so, we use a *band-stop* filter (see fig. 4.2), inspired by periodic noise removal techniques in spectral-based image processing [55].

Concretely, we stack all the frequencies of all the d channels in a single $d \times n$ matrix F , where the j -th row F_j contains the n DCT coefficients in eq. (4.1). Then, for each target style, we define a binary filter $\mathbf{b} \in \{0, 1\}^n$, which contains zero elements only in specific bands (see section 4.3.3 for details). We use $\mathbf{b}$ to zero out those columns in F which should be filtered:

$$S = F \odot M[\mathbf{b}], \quad (4.2)$$

where $\odot$ is the Hadamard product, and $M[\mathbf{b}]$ is a $d \times n$ matrix in which all the elements are ones except those corresponding to the columns in $\mathbf{b}$, which are zeros.

S is the spectral representation of V , in which frequencies $\mathbf{b}$ are *ignored*. Note that, differently from [147], where the spectrum of each neuron is individually filtered, in our case $\mathbf{b}$ is uniformly used for all the d dimensions. This is because an artifact is a complex visual structure, most likely simultaneously involving different dimensions of the CLIP space. S is fi-

nally back-projected into the original CLIP space using the inverse DCT (IDCT), obtaining $\hat{V} = \{\hat{\mathbf{v}}_0, \dots, \hat{\mathbf{v}}_{n-1}\}$. $\hat{V}$ is a representation of I which can be used jointly with different metrics (*e.g.*, the Euclidean metric or a cosine similarity, etc.) to compute a CLIP-based similarity between images or between images and text *which is not influenced by the frequencies in $\mathbf{b}$* . This way, we can condition the generation process using a textual sentence (section 4.3.2) while simultaneously ignoring those frequencies corresponding to the artifact generation.

4.3.2 Computing an image-text similarity

In this section, we show how the proposed spectral-based filtering of an image representation (section 4.3.1) can be used to condition a generative process and plugged into existing text-conditioned generative frameworks with negligible modifications of the original approaches. In our experiments, we use both CLIPstyler [79] (a state-of-the-art text-guided style transfer method, see sections 4.1 and 4.2) and the VQGAN+CLIP method [28] adopted in [107].

VQGAN+CLIP [28] is a text-to-image generation approach based on VQGAN discrete latent codes [42]. The latter are randomly sampled and then optimized using the cosine similarity between the CLIP embedding of the generated image ($\mathbf{z}_v$) and the CLIP embedding of a textual prompt ($\mathbf{z}_t$). The only thing we need to change to use SpectralCLIP in this framework is the image representation. To do so, we use $\mathbf{z}_v = \hat{\mathbf{v}}_0$, where $\hat{\mathbf{v}}_0$ is the representation of the class token in $\hat{V}$ (section 4.3.1). Note that $\hat{\mathbf{v}}_0 \neq \mathbf{v}_0$ because the frequencies in $\mathbf{b}$ have been removed. Other possible choices can be, *e.g.*, using an average pooling of $\hat{V}$ or a linear projection of the concatenation of all the elements of $\hat{V}$ into a vector of the same dimensions as $\mathbf{z}_t$. Following [28] we use the class token which is a simple and effective solution.

band	Frequency index	Period (tokens)
b_1	0-1	25- ∞
b_2	2-3	7-25
b_3	4-7	4-7
b_4	8-15	2-4
b_5	16-49	1-2

Table 4.1: Correspondences between the bands, frequency indexes and period (tokens). The corresponding periods are approximate numbers of tokens that are needed to complete a cosine wave cycle.

CLIPstyler [79] is based on a U-Net generator [138] which, given a content image I_c as input, generates a style-transferred image I_s . To condition the generation process on a style textual description s (where s is a natural language sentence), the embedding of s , obtained using the textual CLIP encoder, is compared with the textual embedding of a fixed sentence (“Photo”). The difference between these two textual embeddings should have the same direction of the difference between the visual embedding of I_c and I_s . This *directional CLIP loss*, initially proposed in [50], is further developed in CLIPstyler by introducing patch-level comparisons. We adopt exactly the same framework, and the only necessary change to use SpectralCLIP in CLIPstyler is to replace the standard CLIP visual embedding of an image (or an image patch) with our filtered representation. Since in CLIPstyler an image/image patch is represented using the class token, we analogously use the class token extracted from $\hat{V}$ (i.e., $\mathbf{z}_v = \hat{\mathbf{v}}_0$).

4.3.3 Band selection

So far we have assumed that we can associate a frequency filter $\mathbf{b}$ to a given textual description of a style (in CLIPstyler) or to a textual prompt (in VQGAN+CLIP). However, since selecting the best $\mathbf{b} \in \{0, 1\}^n$ would be intractable, we adopt a simpler solution, inspired by [147], where the whole spectrum of frequencies $(0, \dots, n - 1)$ is split in fixed bands, each

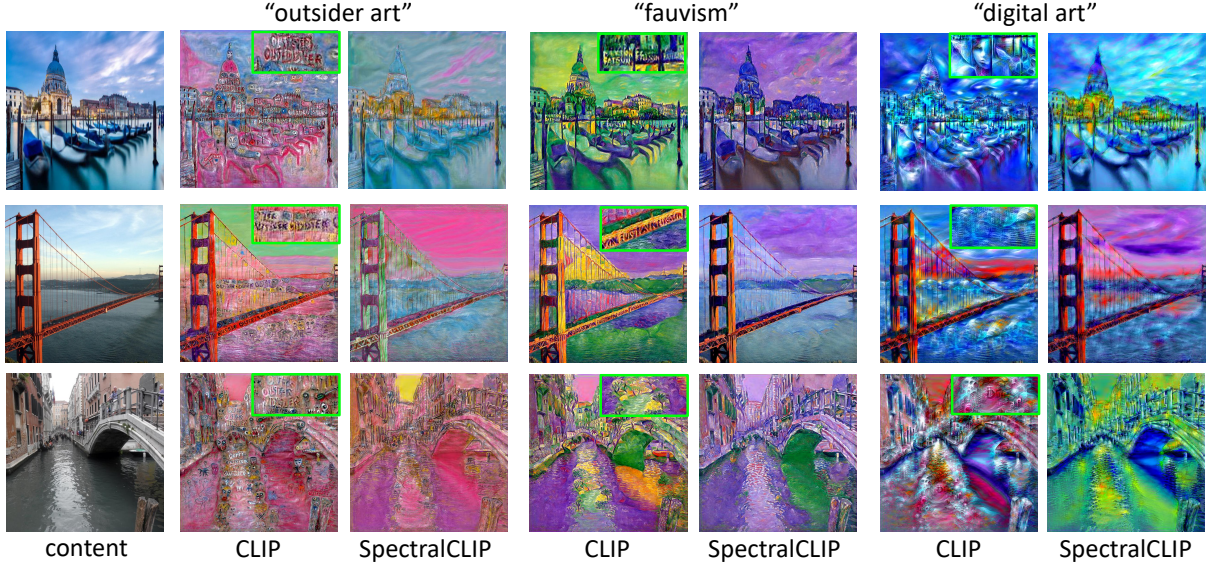


Figure 4.3: Style transfer results using either the CLIP or SpectralCLIP to condition the generation. SpectralCLIP effectively prevents artifact generation while achieving similar style features, e.g., color, paint stroke (artifacts are *highlighted*).

being a contiguous interval of frequencies. Note that each index m of the DCT has a frequency of $2m$ (see eq. (4.1)), and that given a sequence of length N , the period is $N/2m$ (it takes $N/2m$ tokens to complete a cycle). For example, in our case where ViT-B/32 is used as the vision encoder, the sequence length is $N = (224/32)^2 + 1 = 50$. Therefore, index 1 of the DCT corresponds to the period of 25, and index 5 to the period of 5, etc. Specifically, we define the following 5 bands: $b_1 = [0, 1]$, $b_2 = [2, 3]$, $b_3 = [4, 7]$, $b_4 = [8, 15]$, $b_5 = [16, 49]$. The correspondence between these frequency bands and periods is presented in table 4.1. In ViT-B/32, the input image is first resized to the resolution of 224^2 , and then divided into 7×7 image patches of 32^2 . In this sense, it can be estimated that b_1 relates to artifacts that roughly span more than 3 lines of patches, b_2 to those spanning 2 lines, and $b_3 - b_5$ to small artifacts spanning within 1 line.

Another problem is that, given a style description, the artifact condition is unpredictable. Specifically, it is difficult to judge if the stylized image contains artifacts or not, as well as to detect the artifact appearance. Nev-

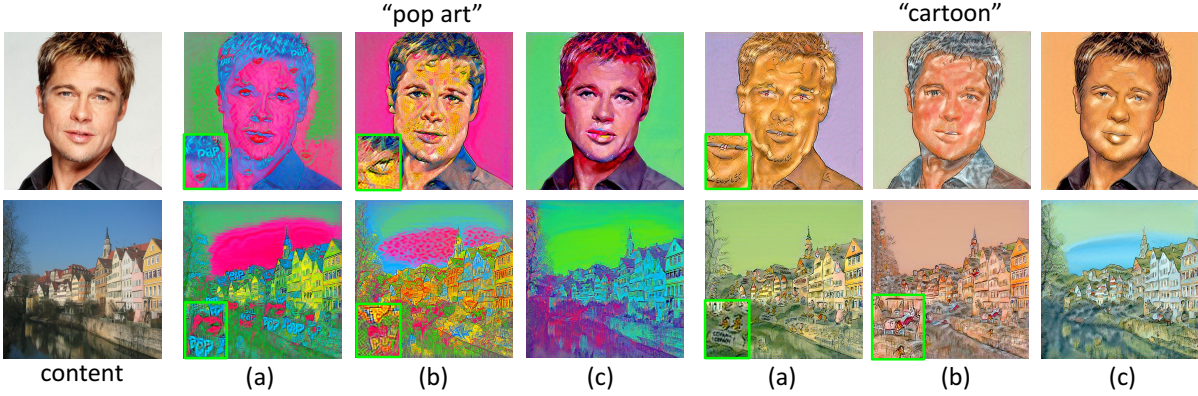


Figure 4.4: Comparisons among text-guided style transfer results, generated using CLIP-styler with (a) CLIP, (b) forget-to-spell CLIP [107], and (c) SpectralCLIP, respectively (artifacts are *highlighted*).

ertheless, through experiments on various styles, we find the artifacts are usually at three scales. Therefore, we propose a simple yet effective method based on empirical studies. Through experimenting on multiple band combinations, we find three filtering strategies ($c_1 = \{b_1, b_2, b_4\}$, $c_2 = \{b_1, b_2\}$, $c_3 = \{b_1\}$) that are effective for preventing the artifacts at the corresponding three scales, respectively. For instance, using c_1 , the associated filter $\mathbf{b}$ contains ones in the intervals b_1, b_2, b_4 , and it is used in eq. (4.2) to zero out the corresponding bands. We use visual inspection to select the band combination that leads to the best result, then it is used in all image stylisation conditioned on s . This selection step is done *only once per given style s* using a single image content. More discussions are in Sec. section 4.5.1.

4.4 Experiments

4.4.1 Style Transfer Results

In this section, we evaluate SpectralCLIP in a text-guided style transfer task. For a fair comparison with CLIPstyler[79], we use its same network, loss functions, training protocols, hyperparameters, etc., changing *only* the

basic image-text similarity as described in section 4.3.2.

Qualitative results. In fig. 4.1 and fig. 4.3, we show multiple text-guided style transfer results generated using either the standard CLIP space (i.e., the original CLIPstyler method) or our SpectralCLIP. These images show that CLIPstyler frequently generates visual and textual artifacts. By contrast, the results generated using SpectralCLIP do not have the issues of both visual artifacts and textual artifacts while the styles are presented well. Take our results of “outsider art” (fig. 4.3) for example, the style shown in the images is similar to the style in the results of CLIPstyler, but with artifacts excluded.

Quantitative results. Quantitatively evaluating a style-transfer approach is difficult because of the lack of a universally accepted metric that can assess the reflection of a target style in a generated image. On the other hand, using a cosine similarity between the CLIP-based representations of the target style and the generated image may favour those methods (such as CLIPstyler) which use the same similarity in the optimization stage. To partially solve this problem, we use an additional metric, based on the “learn-to-spell” projection of the CLIP space proposed (and trained) in [107] (section 4.2). The idea behind this metric is that since part of the artifacts have a textual nature (i.e., strings drawn on the generated images, sections 4.1 and 4.2), then a learn-to-spell based similarity between a generated image and the corresponding textual description of the style should be *higher* for those images containing *more* textual artifacts.

To explain further, Materzynska *et al.* [107] analyse the entanglement problem of written texts and visual concepts in the CLIP space, and learn two orthogonal projections, i.e., “forget-to-spell” and “learn-to-spell”, the former is for recognizing visual concepts while the latter is for recognizing written text. Specifically, in this paper, we use the “learn-to-spell” projection to measure the textual artifact condition in the generated stylized

		CLIPstyler w. CLIP	CLIPstyler w. SpectralCLIP
cartoon	<i>CLIP</i>	0.269 ± 0.014	0.256 ± 0.014
	<i>CLIP-Spell</i>	0.482 ± 0.056	0.441 ± 0.066
pop art	<i>CLIP</i>	0.315 ± 0.023	0.287 ± 0.018
	<i>CLIP-Spell</i>	0.419 ± 0.137	0.353 ± 0.121
visionary art	<i>CLIP</i>	0.322 ± 0.016	0.278 ± 0.018
	<i>CLIP-Spell</i>	0.527 ± 0.086	0.397 ± 0.057
outsider art	<i>CLIP</i>	0.314 ± 0.018	0.255 ± 0.019
	<i>CLIP-Spell</i>	0.571 ± 0.108	0.360 ± 0.095

Table 4.2: Average cosine similarity between the stylized images and the textual description of the style, measured both on the original CLIP space ($\uparrow$) and on the learn-to-spell CLIP (in short referred to as CLIP-Spell) ($\downarrow$).

image, as the more the image contains textual artifacts, the higher the similarity between the image and the textual description of style using the “learn-to-spell” projection (as in Tab. 4.2 and Fig. 4.12).

Concretely, we sample 100 images from the COCO [97] val-set and use them as content images. Then, for each style, we generate the style transfer results using either SpectralCLIP or CLIPstyler. Finally, we use both the original CLIP space and learn-to-spell CLIP projection [107] to evaluate the similarity between the textual style description and the generated images. table 4.2 shows that using SpectralCLIP, the learn-to-spell CLIP score is significantly reduced, indicating that SpectralCLIP effectively prevents textual artifact generation. On the other hand, the CLIP similarity is also reduced; however, as aforementioned, this metric is biased towards CLIPstyler, where the whole, non-filtered CLIP image representation is used for optimization.

4.4.2 Comparison with Forget-to-Spell CLIP

In this section, we compare SpectralCLIP with “forget-to-spell” CLIP [107], which is a learned subspace of CLIP semantic space that alle-

(%)	w. CLIP	w. forget-to-spell	w. SpectralCLIP
Overall	34.44	10.28	55.28
Artifact-Free	19.45	6.11	74.44

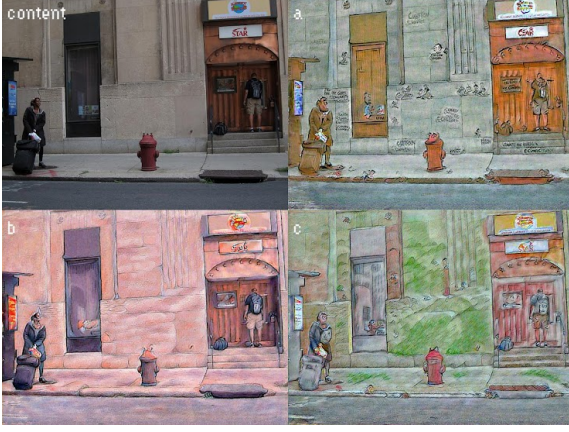
Table 4.3: User preference of the three style transfer methods with respect to the overall quality of the generated images ($\uparrow$) and the presence of artifacts ($\uparrow$).

viates the text-image entanglement problem (section 4.1). Specifically, we again use CLIPstyler as the baseline and we replace its (standard) CLIP space with the forget-to-spell projection proposed and trained in [107]. Hence, we compare three methods: (a) CLIPstyler with CLIP (i.e., the original CLIPstyler), (b) CLIPstyler with the forget-to-spell CLIP, and (c) CLIPstyler with SpectralCLIP.

Qualitative results. From the qualitative results shown in fig. 4.4, we draw three conclusions: 1) the images generated by the original CLIPstyler contain both visual and textual artifacts; 2) using the forget-to-spell CLIP alleviates the textual artifact issue, but it still generates visual artifacts, which makes the results unlike human created artworks; and 3) similar to the analysis in section 4.4.1, using SpectralCLIP, no visual nor textual artifacts have been generated, improving the overall quality of the results.

User Study. We further compare the three methods through a user study. Specifically, we use 10 styles jointly with two different tasks which respectively analyse: 1) the overall quality of the generated images, asking the users to assess whether the stylized results are consistent with the target style, the content is well preserved, and no inharmonious artifacts are generated; and 2) the specific artifact issue, asking the users to assess the generated images regarding the possible presence of visual/textual artifacts. We randomly sampled 100 content images from the COCO val-set and then created a questionnaire with 24 questions. We recruited 30 users, who were asked to select one out of the three methods for each content image. The

Please select the best stylized image considering: (i) how much it reflects the cartoon style, (ii) how much it preserves the content of the content image, (iii) how much it is clean (i.e., it does not contain textual or visual artefacts)



- ☐ a
☐ b
☐ c

(a)

Please select the best stylized image considering: (i) how much it preserves content and (ii) how much it is clean (i.e. it does not contain textual or visual artefacts).



- ☐ a
☐ b
☐ c

(b)

Figure 4.5: The SpectralCLIP user study questions used to evaluate two tasks: (a) the overall image quality, and (b) the possible appearance of artifacts.

user preference results, as reported in table 4.3, show that SpectralCLIP gets the best scores on both tasks. Specifically, SpectralCLIP achieves a significantly higher preference score (77.44%) in the artifact-free evaluation, indicating the effectiveness of our proposal in preventing artifacts.

In details, we asked 30 participants to answer 24 questions, split in two tasks (12 questions per task), respectively evaluating the overall quality of the generated images and the possible appearance of artifacts. fig. 4.5 shows an example of the both tasks. Since evaluating the overall quality of a style-transfer task requires the participants to consider whether the stylized results reflect target style, for the first task we selected 4 better-known styles (i.e., “pop art”, “cartoon”, “fauvism” and “Giorgio Morandi”). For the second task (artifact evaluation), we used the other 6 styles (“low-brow”, “outsider art”, “visionary art”, “harlem renaissance”, “neon art” and “digital art”). For each style, we randomly sampled 10 images from the

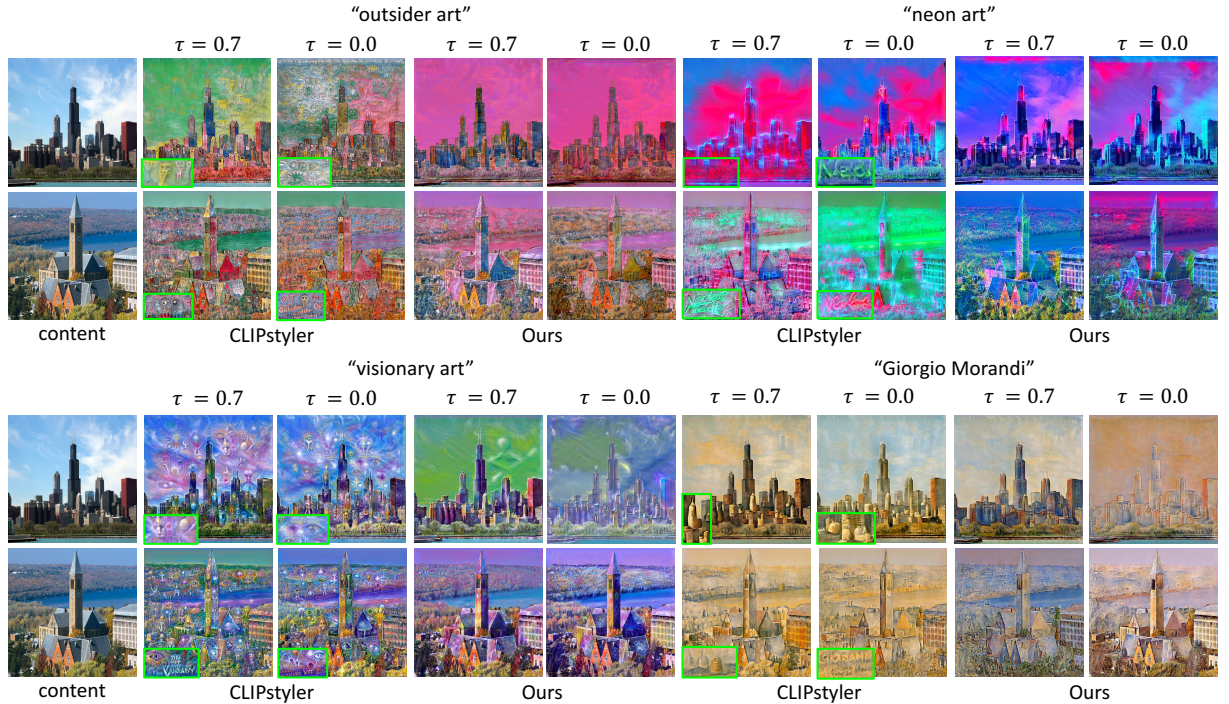


Figure 4.6: The effect of the patch-rejection threshold τ in the CLIPstyler patch-loss (artifacts are *highlighted*, zoom in to see details).

COCO val-set and used them as content images to generate the stylized images with the three methods. Thus, we obtain 100 groups of stylized results in total (each composed of 3 images generated by the 3 compared methods, e.g., see fig. 4.5). For the first task, we equally sampled 3 groups of stylized results for each style, obtaining 12 questions. For the second task, we randomly sampled 12 groups of images without considering the style. We used Google Forms as the platform.

4.4.3 Hyperparameter Study

Threshold in the patch loss. CLIPstyler uses a patch-rejection threshold τ in its patch loss to alleviate the over-specification problem (section 4.2). The value of this threshold is a (manually selected) hyperparameter, which is fixed to $\tau = 0.7$ in [79]. In fig. 4.6 we compare the use of this threshold ($\tau = 0.7$) with a non-thresholding variant ($\tau = 0.0$). The results

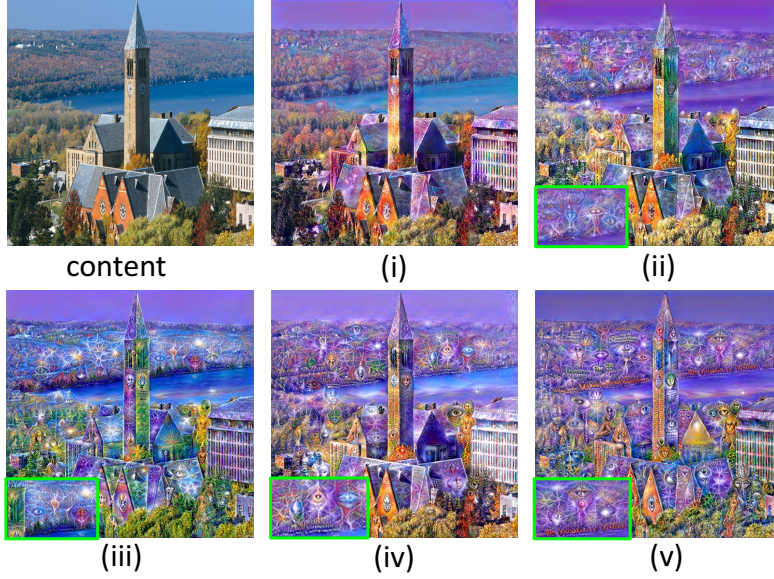


Figure 4.7: The effects of different band-stop filters (artifacts are *highlighted*).

show that the original CLIPstyler is heavily influenced by the thresholding, since its removal ($\tau = 0.0$) leads to the generation of many more artifacts, independently of the target style. By contrast, SpectralCLIP is much less sensitive to this thresholding, since the results using $\tau = 0.0$ are consistent with the images generated with $\tau = 0.7$, showing that our method does not rely on this thresholding step to avoid over-specification.

Band selection.

We study the effect of the band selection (section 4.3.3) using the style “visionary art” (fig. 4.7), jointly with five different filters: (i) masking bands 1, 2 and 4 (corresponding to c_1 in section 4.3.3); (ii) masking bands 1, 2 and 5; (iii) masking bands 1 and 2 (c_2 in section 4.3.3); (iv) masking bands 1 (c_3 in section 4.3.3); and (v) only masking the lowest frequency (i.e., the frequency index $m = 0$). fig. 4.7 shows that the visual appearance change caused by higher frequencies tends to be more local, as can be observed by comparing (i) with (ii), and (ii) with (iii). For instance, comparing (i) with (ii), the former filter leads to greater visual appearance changes within a larger region. Moreover, the differences between (ii)

and (iii) are marginal, indicating that band 5 (which includes the highest frequencies) is related to changes in smaller areas. Furthermore, lower frequencies result in the generation of larger artifacts, as shown by the comparison between (iii) and (iv), and between (iv) and (v). For example, not masking band 2 leads to the generation of larger and more obvious artifacts on the eyes. A similar phenomenon can be observed when we additionally do not mask the frequency with index 1 (see (iv) and (v)). These results confirm our assumption (section 4.1) that most artifacts are visual structures with a specific period, whose generation can be prevented by adopting the proposed spectral representation and the corresponding frequency filters.

4.4.4 Text-to-Image Generation

To test the generality of SpectralCLIP, we consider a different task, adopting the text-to-image generation framework used in [107] to evaluate the word-image disentanglement of forget-to-spell and learn-to-spell (sections 4.1 and 4.4.2). Specifically, following [107], we use VQGAN+CLIP [28] and we replace its text-image similarity computed on the original CLIP space with SpectralCLIP. fig. 4.8 compares the results obtained with VQGAN+CLIP and VQGAN+SpectralCLIP, and confirms the observations of Materzynska *et al.* [107], who highlight that VQGAN+CLIP frequently generates inappropriate textual strings (textual artifacts) mixed with visual content. By contrast, this problem is largely alleviated with SpectralCLIP. Meanwhile, the generated image content in VQGAN+SpectralCLIP is still consistent with the given text prompt (except for the nonsense text input “irmin”). In all the VQGAN+SpectralCLIP results shown in fig. 4.8, we use the same filtering strategy (masking only band 4, i.e., b_4 , see section 4.3.3). Note that the scale of a textual/visual artifact depends on the CLIP encoder input, which is the full image in



Figure 4.8: Text-to-image generation results using (a) VQGAN+CLIP, and (b) VQGAN+SpectralCLIP.

the case of VQGAN, and this results in a shorter period with respect to CLIPstyler.

4.5 Discussions and Conclusion

4.5.1 Discussions

As mentioned in section 4.3.3, for each target style we select one of the three band combinations c_1, c_2 or c_3 , and then we use the corresponding filter

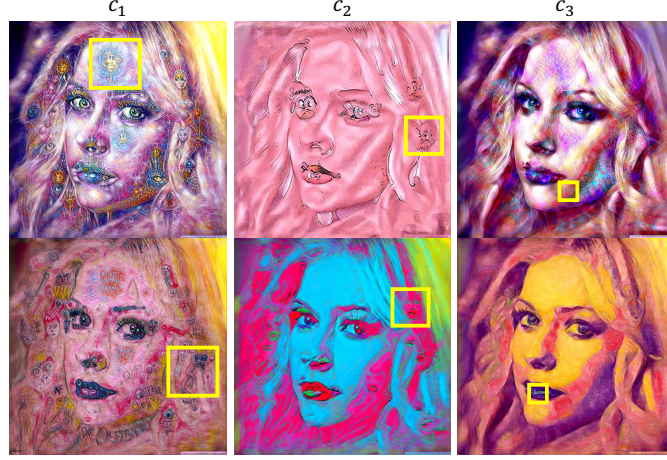


Figure 4.9: Different styles result in artifacts at different scales which are tricky to predict.

for all the images translation of that style. Note that this is a negligible operation if compared, *e.g.*, to the collection of a set of reference images used to train a style-specific generator [140, 186, 78]. Moreover, there are two underlying reasons: 1) given a style description, the artifact condition is unpredictable. To be specific, it is difficult to judge if the stylized image contains artifact or not, as well as to detect the artifact appearance; 2) the artifact scales are of limited variable ranges.

Hence, we propose SpectralCLIP, a simple yet effective method based on empirical studies. Through experimenting on multiple band combinations, we find three filtering strategies ($c_1 = \{b_1, b_2, b_4\}$, $c_2 = \{b_1, b_2\}$, $c_3 = \{b_1\}$) that are effective for preventing the artifacts at the corresponding three scales (as shown in Fig. 4.9) (and through experiments on various styles, we find the artifacts are usually of one of the three scales). To prevent artifacts of larger scales, more frequency bands should be masked. The bands are defined according to the scales of the artifacts, i.e., the length of the dependant visual tokens in the CLIP-ViT scenario.

In the leftmost column of fig. 4.11, we show the stylized results of CLIPstyler corresponding to three different style descriptions (“outsider art”, “cartoon” and “digital art”). In all cases, CLIPstyler generates a lot of ar-

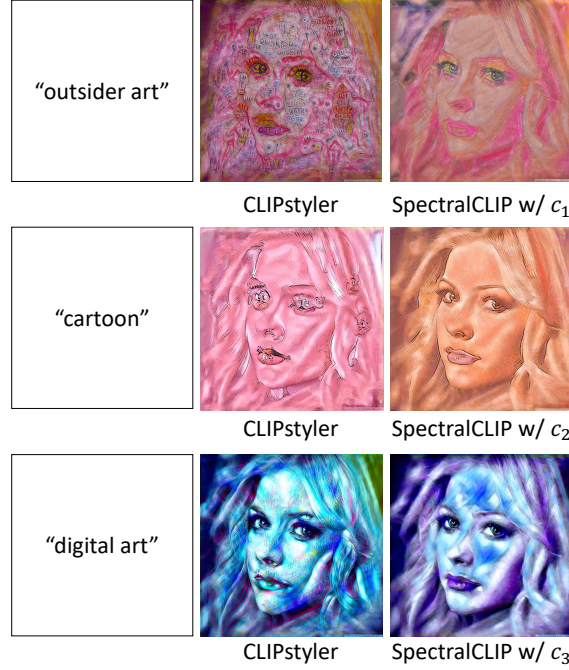


Figure 4.10: By CLIPstyler, different styles lead to different types of artifacts. Therefore, in SpectralCLIP, we filter different band combinations (indicated on the bottom).

tifacts. Additionally, the corresponding scales vary from style to style. A simple comparison can be seen in Fig. 4.9. As mentioned in section 4.3.3, in order to select the most suitable filter for each style, we use a single content image per style and we generate stylized images using our SpectralCLIP and one among c_1 , c_2 and c_3 . Correspondingly, we give an illustration in fig. 4.11, where we show the results obtained using SpectralCLIP with one filtering band combination among c_1 , c_2 or c_3 . For “outsider art”, c_1 is the best band combination, while c_2 is the best for “cartoon”, and c_3 for “digital art”.

In addition, since we can use the CLIP/forget-to-spell CLIP to generate stylized images with/without text artifacts and use learn-to-spell CLIP to measure the text artifact presence, we did a spectral analysis to find further support for our work. Specifically, We first collect 100 images with/without artifacts using CLIP/forget-to-spell CLIP, respectively. Then, we individ-

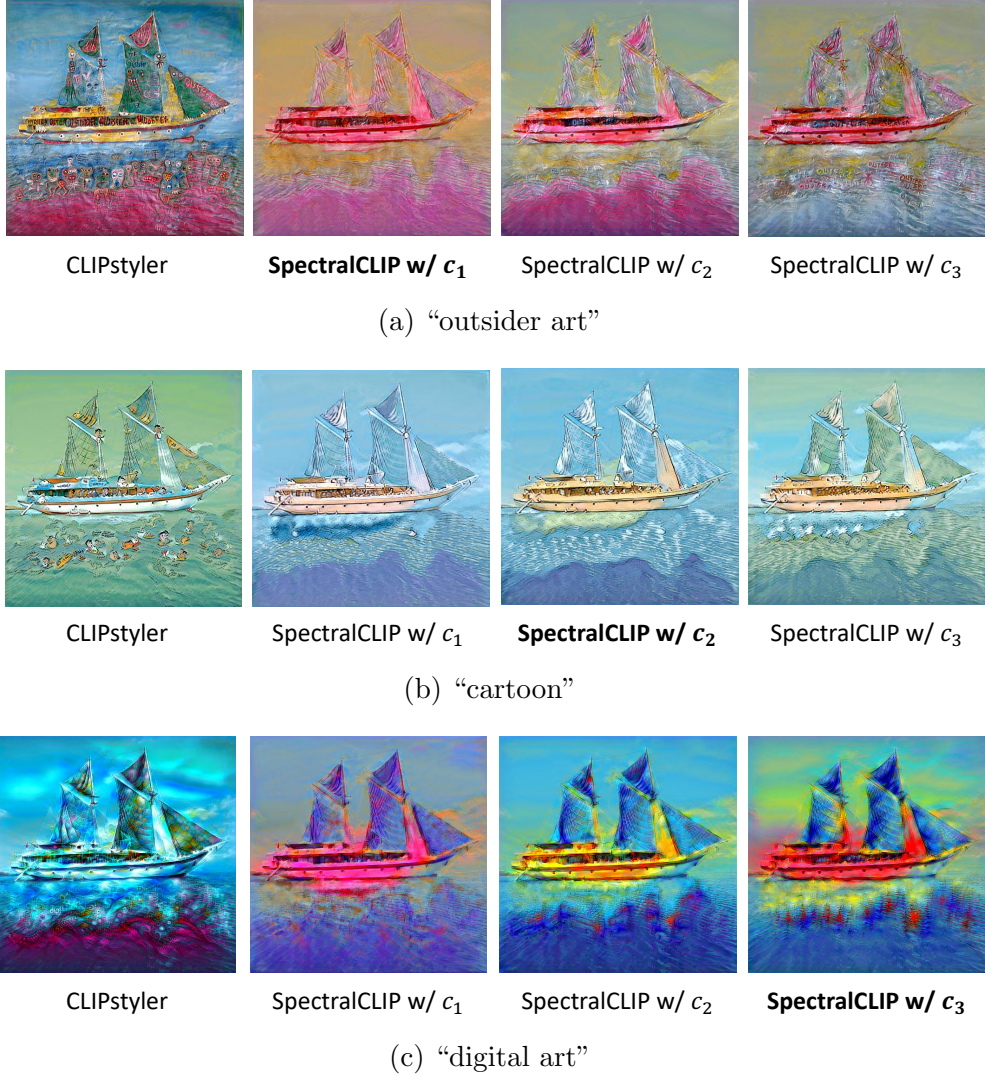


Figure 4.11: Leftmost column: images generated using CLIPstyler and three different styles. All the other columns show the results obtained using SpectralCLIP and one among c_1, c_2 or c_3 as the filter. In case of "outsider art", c_1 is the best band combination, while c_2 is the best for "cartoon", and c_3 for "digital art".

ually mask each frequency at one time and use the filtered CLIP representation to compute the cosine similarity using learn-to-spell CLIP (the computations are based on patches as in CLIPstyler).

As shown in Fig. 4.12, masking frequency 0 and 1 significantly reduces the learn-to-spell CLIP similarity scores of the images containing textual artifacts, indicating that those frequencies are related to text artifacts (e.g.,

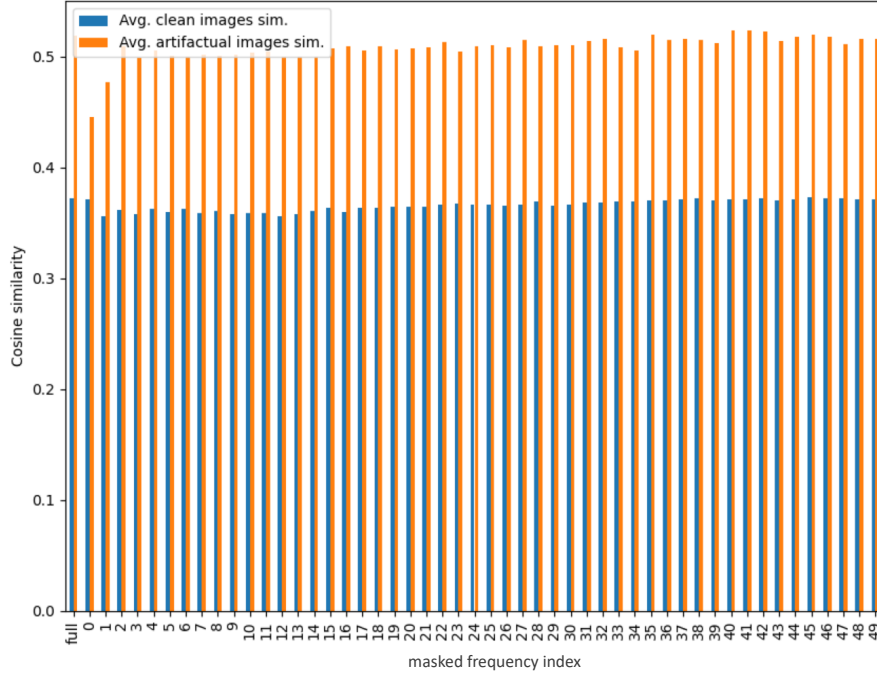


Figure 4.12: Spectral analysis w.r.t. textual artifacts.

Band combination	Style
c_1	Lowbrow
	Outsider art
	Visionary art
	Rosy-color oil painting
	Makoto Shinkai
c_2	Pop art
	Cartoon
	Giorgio Morandi
	Harlem renaissance
	Neon art
	Contemporary art
c_3	Francoise Nielly
	Fauvism
	Digital art

Table 4.4: Band combination used for each style.

in Fig. 4.9, masking c_3 is useful to prevent the generation of text artifacts).

In table 4.4, we provide the band combination we used for each of the styles presented in this paper.

4.5.2 Limitations

Despite the promising results of SpectralCLIP, there are still some limitations. Firstly, we empirically analyse the artifact patterns present in a range of artistic styles, and mask out certain bands using one of the three general filters. The reason why a certain target style tends to produce different scales of artifacts is still unclear. This may require a deeper understanding of how CLIP captures these artistic concepts when it was pre-trained. Secondly, this work defines three general band combinations that effectively produce cleaner stylised images. A more promising alternative for future work is to automatically select frequency bands that cater to a target style. Recently, in the language domain, Müller-Eberstein *et al.* [109] promote [147] and develop learnable filters rather than handcrafted ones, offering an intriguing direction to follow. To this end, for image style transfer, a widely recognised metric to measure the presence of artifacts is still missing.

4.5.3 Conclusion

Despite the wide success of vision-language foundation models like CLIP in different vision-language tasks, directly using CLIP for style transfer suffers from the generation of visual and textual artifacts. To resolve this problem, we propose SpectralCLIP, which transforms the CLIP embedding sequence into the frequency domain and filters those frequencies whose period corresponds to the artifact scales. Experimental style transfer results show that SpectralCLIP significantly mitigates artifact generation, thus improving the realistic degree and the quality of the generated images.

This chapter addressed the emergence of artifacts in CLIP-guided generation by introducing a spectral filtering approach to disentangle visual semantics from textual noise. Together with the preceding chapters, it

highlights how language, while powerful, requires careful modeling and interpretation across multiple aspects of the generative process—from controllability and abstraction to robustness. In the next, we conclude the thesis by summarizing these insights and outlining promising directions for future work.

Chapter 5

Conclusion and Future Work

This thesis has investigated the use of natural language as a high-level interface for controllable image generation, focusing on key challenges in precision, abstraction, and robustness. While recent vision-language foundation models such as CLIP and DALL·E have greatly expanded the possibilities of multimodal synthesis, treating language as a precise and interpretable control mechanism remains an open problem.

To address this, we introduced three complementary approaches. First, we proposed *Predict, Prevent, and Evaluate (PPE)*, a framework for disentangled image manipulation that anticipates and regularizes semantic entanglements based on CLIP-informed attribute co-occurrence. Second, we presented *StylerDALL·E*, a language-supervised style transfer model that operates in the discrete latent space of a generative tokenizer, enabling abstract style control while preserving semantic content. Third, we developed *SpectralCLIP*, a frequency-aware filtering method that mitigates visual and textual artifacts in CLIP-guided generation by attenuating disruptive patterns in the embedding space. Together, these contributions demonstrate that language, when appropriately modeled and guided, can offer fine-grained, flexible control over the generative process.

Despite these advances, several limitations remain. Our methods rely

heavily on CLIP as a semantic backbone, inheriting its biases and limitations in attribute grounding and linguistic granularity. The discrete token-based generation in StylerDALLE, while efficient and flexible, may limit fine-detail control compared to pixel-space or diffusion-based methods. Moreover, the spectral properties of vision-language models are still underexplored, and the interpretation of frequency components remains largely empirical.

These observations open several directions for future work. One promising path is to generalize our frameworks to richer domains such as video generation [61, 9, 119], 3D synthesis [100, 126, 95, 20] or world model [11, 115]. Another opportunity lies in extending our methods beyond their current scope—for example, while StylerDALLE is currently designed for vector-quantized image generation models, it could be adapted to operate on continuous latent spaces, thereby broadening its applicability to diffusion-based architectures [89]. Another is to deepen the theoretical understanding of multimodal embedding spaces through spectral or geometric analysis. Extending these methods beyond CLIP—towards open-vocabulary vision-language models or instruction-tuned transformers—could also enhance generalization and reduce reliance on specific pre-trained backbones. Finally, as language becomes a more powerful generative interface, issues of safety, misuse, and social impact must be addressed, particularly in applications involving identity, realism, or misinformation [10, 13, 111].

In summary, this thesis highlights both the promise and complexity of using language as a true interface for visual generation. By bridging linguistic intent and visual synthesis through structured, interpretable mechanisms, we hope to contribute to a broader vision of human-centric, controllable, and trustworthy generative systems.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Nicu Sebe, for his continuous guidance, support, and encouragement throughout my doctoral journey. I am also grateful to Enver Sangineto at the University of Modena and Reggio Emilia for his generous mentorship and collaboration, and to Prof. Philip Torr at the University of Oxford, who kindly hosted me during my research visit.

I would like to thank all the mentors and collaborators I have had the privilege of working with during my internships. At Baidu, I benefited immensely from working with Tianwei Lin, Fu Li, and Dongliang He. At Naver Labs Europe, I am thankful to Riccardo Volpi, Diane Larlus, and Gabriela Csurka for their guidance and collaboration. During my visit to Oxford, I also had the pleasure of working closely with Srikumar Ramalingam, Sadeep Jayasumana, and Jindong Gu from Google—thank you for the stimulating discussions and insightful guidance.

Special thanks go to my former master’s advisor, Prof. Xiaojie Wang, and to Fandong Meng, who supported and encouraged me during my internship at Tencent. Their mentorship played a pivotal role in inspiring my passion, confidence, and love for research.

I am also thankful to the many friends and colleagues I met during my PhD. I truly appreciate the time we shared and the friendships we built along the way.

To my husband, Xubo Liu, who is not only a brilliant researcher in a close field but also my most honest and insightful critic—thank you for helping me see the parts of the world I might otherwise have missed.

I would like to thank my family for their unwavering love and support. In particular, I am deeply grateful to my mother, Yan Wang, whose strength and unconditional encouragement have always been a source of inspiration.

I would not be where I am today without them.

Lastly, I would like to thank myself—for the perseverance, the longing for truth, the curiosity about the world, and the constant pursuit of something better that have carried me this far. Today, I see the mountain as a mountain, and the river as a river, yet I understand this is only one step in the journey of life. May I continue to see a broader world, to encounter the unknown within myself, and to cross paths with those who pursue their own truths with courage and curiosity.

Bibliography

- [1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Style-flow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (TOG)*, 40(3):1–21, 2021.
- [2] Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khadelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick Von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, and Sophia Yang. Pixtral 12b, 2024.
- [3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a matter of style: Age transformation using a style-based regression model. *ACM Trans. Graph.*, 40(4), 2021.
- [4] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Mil-

- lican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.
- [5] Anthropic. Claude: An ai assistant by anthropic. <https://www.anthropic.com/index/introducing-claude>, 2023.
- [6] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022.
- [7] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *arXiv preprint arXiv:2103.10951*, 2021.
- [8] Fengxiang Bie, Yibo Yang, Zhongzhu Zhou, Adam Ghanem, Minjia Zhang, Zhewei Yao, Xiaoxia Wu, Connor Holmes, Pareesa Golnari, David A Clifton, et al. Renaissance: A survey into ai text-to-image generation in the era of large model. *IEEE transactions on pattern analysis and machine intelligence*, 2024.
- [9] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendele-
vitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scal-
ing latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
- [10] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shya-
mal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, An-
nie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky,

Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022.

- [11] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter,

- Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.
- [12] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instruct-pix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.
- [13] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwal, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models, 2023.
- [14] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.
- [15] Haibo Chen, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, Dongming Lu, et al. Artistic style transfer with internal-external learning and contrastive learning. *Advances in Neural Information Processing Systems*, 34:26561–26573, 2021.

- [16] Haibo Chen, Lei Zhao, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Dualast: Dual style-learning networks for artistic style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 872–881, 2021.
- [17] Haibo Chen, Lei Zhao, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Dualast: Dual style-learning networks for artistic style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 872–881, June 2021.
- [18] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pages 74–91. Springer, 2024.
- [19] Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [20] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22246–22256, October 2023.

- [21] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016.
- [22] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.
- [23] Jiaxin Cheng, Ayush Jaiswal, Yue Wu, Pradeep Natarajan, and Prem Natarajan. Style-aware normalized loss for improving arbitrary style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–143, 2021.
- [24] Myungsub Choi. Referring object manipulation of natural images with conditional classifier-free guidance. In *European Conference on Computer Vision*, pages 627–643. Springer, 2022.
- [25] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [26] Noam Chomsky. *Syntactic structures*. Mouton de Gruyter, 2002.
- [27] Herbert H Clark. *Using language*. Cambridge university press, 1996.
- [28] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural lan-

- guage guidance. In *European Conference on Computer Vision*, pages 88–105. Springer, 2022.
- [29] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.
- [30] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [31] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11326–11336, 2022.
- [32] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11326–11336, June 2022.
- [33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

- [34] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.
- [35] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *arXiv preprint arXiv:2204.14217*, 2022.
- [36] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5706–5714, 2017.
- [37] Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. Prompt tuning inversion for text-driven image editing using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7430–7440, 2023.
- [38] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [39] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. 2023.
- [40] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schel-

- ten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.
- [41] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.
- [42] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.
- [43] Michael C Frank and Noah D Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998, 2012.
- [44] Kevin Frans, Lisa B. Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *CoRR*, abs/2106.14843, 2021.
- [45] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023.
- [46] Tsu-Jui Fu, Xin Eric Wang, and William Yang Wang. Language-driven artistic style transfer. In *European Conference on Computer Vision (ECCV)*, 2022.
- [47] Aviv Gabbay, Niv Cohen, and Yedid Hoshen. An image is worth more than a thousand words: Towards disentanglement in the wild. In *Neural Information Processing Systems (NeurIPS)*, 2021.
- [48] Aviv Gabbay and Yedid Hoshen. Demystifying inter-class disentanglement. In *International Conference on Learning Representations (ICLR)*, 2020.

- [49] Aviv Gabbay and Yedid Hoshen. Scaling-up disentanglement for image translation. In *International Conference on Computer Vision (ICCV)*, 2021.
- [50] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.
- [51] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2021.
- [52] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.
- [53] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 12709–12720, 2024.
- [54] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021.
- [55] R.C. Gonzalez and R.E. Woods. *Digital Image Processing, fourth edition*. Pearson, 2018.

- [56] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *ICLR*, 2018.
- [57] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8222–8231, 2018.
- [58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [59] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.
- [60] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, 2001.
- [61] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.
- [62] Nisha Huang, Yuxin Zhang, Fan Tang, Chongyang Ma, Haibin Huang, Weiming Dong, and Changsheng Xu. Diffstyler: Controllable dual diffusion for text-driven image stylization. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
- [63] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

- [64] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371, 2024.
- [65] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Proc. NeurIPS*, 2020.
- [66] Ray Jackendoff. Précis of foundations of language: Brain, meaning, grammar, evolution. *Behavioral and brain sciences*, 26(6):651–665, 2003.
- [67] Ali Jahanian, Lucy Chai, and Phillip Isola. On the ”steerability” of generative adversarial networks. In *International Conference on Learning Representations*, 2020.
- [68] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.
- [69] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10124–10134, 2023.
- [70] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

- [71] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.
- [72] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.
- [73] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.
- [74] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6007–6017, 2023.
- [75] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2426–2435, 2022.
- [76] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.
- [77] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

- [78] Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4422–4431, 2019.
- [79] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18062–18071, 2022.
- [80] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025.
- [81] Yoann Lemesle, Masataka Sawayama, Guillermo Valle Pérez, Maxime Adolphe, Hélène Sauzéon, and Pierre-Yves Oudeyer. Language-biased image classification: evaluation based on semantic representations. In *ICLR*, 2022.
- [82] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. *Advances in neural information processing systems*, 32, 2019.
- [83] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7880–7889, 2020.

- [84] Bowen Li, Xiaojuan Qi, Philip Torr, and Thomas Lukasiewicz. Lightweight generative adversarial networks for text-guided image manipulation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22020–22031. Curran Associates, Inc., 2020.
- [85] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, Jianfeng Gao, et al. Multimodal foundation models: From specialists to general-purpose assistants. *Foundations and Trends® in Computer Graphics and Vision*, 16(1-2):1–214, 2024.
- [86] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.
- [87] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.
- [88] Shikai Li, Jianglin Fu, Kaiyuan Liu, Wentao Wang, Kwan-Yee Lin, and Wayne Wu. Cosmicman: A text-to-image foundation model for humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6955–6965, 2024.
- [89] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

- [90] Xinyang Li, Shengchuan Zhang, Jie Hu, Liujuan Cao, Xiaopeng Hong, Xudong Mao, Feiyue Huang, Yongjian Wu, and Rongrong Ji. Image-to-image translation via hierarchical style disentanglement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8639–8648, 2021.
- [91] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020.
- [92] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3809–3817, 2019.
- [93] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Diversified texture synthesis with feed-forward networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3920–3928, 2017.
- [94] Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, et al. Rich human feedback for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19401–19411, 2024.
- [95] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 300–309, June 2023.

- [96] Tianwei Lin, Zhuoqi Ma, Fu Li, Dongliang He, Xin Li, Errui Ding, Nannan Wang, Jie Li, and Xinbo Gao. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5141–5150, June 2021.
- [97] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [98] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.
- [99] Mingcong Liu, Qiang Li, Zekui Qin, Guoxin Zhang, Pengfei Wan, and Wen Zheng. Blendgan: Implicitly gan blending for arbitrary stylized face generation. In *Advances in Neural Information Processing Systems*, 2021.
- [100] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.
- [101] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6649–6658, 2021.
- [102] Yahui Liu, Marco De Nadai, Deng Cai, Huayang Li, Xavier Alameda-Pineda, Nicu Sebe, and Bruno Lepri. Describe what to change: A

- text-guided unsupervised image-to-image translation approach. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1357–1365, 2020.
- [103] Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. Smoothing the disentangled latent style space for unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10785–10794, 2021.
- [104] Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. Smoothing the disentangled latent style space for unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [105] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [106] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [107] Joanna Materzynska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in CLIP. In *CVPR*, 2022.

- [108] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.
- [109] Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. Spectral probing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7730–7741, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.
- [110] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. *arXiv preprint arXiv:1810.11919*, 2018.
- [111] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models, 2023.
- [112] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.
- [113] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit Patel, and Animashree Anandkumar. Semi-supervised stylegan for disentanglement learning. In *International Conference on Machine Learning*, pages 7360–7369. PMLR, 2020.
- [114] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. Face identity disentanglement via latent space mapping. *ACM Trans. Graph.*, 39(6), November 2020.

- [115] NVIDIA, :, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai, 2025.
- [116] OpenAI. <https://github.com/openai/dall-e>.
- [117] OpenAI. Dall-e 3. <https://openai.com/dall-e>, 2023. Accessed: YYYY-MM-DD.
- [118] OpenAI. Gpt-4 technical report. <https://openai.com/research/gpt-4>, 2023.
- [119] OpenAI. Introducing sora. <https://openai.com/sora>, 2024. Accessed: 2025-07-22.

- [120] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5880–5888, 2019.
- [121] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [122] Or Patashnik and Zongze Wu. Official implementation of StyleCLIP. <https://github.com/orpatashnik/StyleCLIP.git>, 2021.
- [123] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, October 2021.
- [124] Pablo Pernias, Dominic Rampas, Mats L Richter, Christopher J Pal, and Marc Aubreville. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. *arXiv preprint arXiv:2306.00637*, 2023.
- [125] Justin NM Pinkney and Chuan Li. clip2latent: Text driven sampling of a pre-trained stylegan using denoising diffusion and clip. *arXiv preprint arXiv:2210.02347*, 2022.

- [126] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022.
- [127] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirror-gan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1505–1514, 2019.
- [128] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.
- [129] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.
- [130] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [131] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.
- [132] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, 2019.

- [133] Scott Reed, Zeynep Akata, Xincheng Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. Pmlr, 2016.
- [134] Facebook Research. LAMA. <https://github.com/facebookresearch/LAMA.git>, 2021.
- [135] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a style-gan encoder for image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.
- [136] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021.
- [137] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [138] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, 2015.
- [139] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

- [140] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Björn Ommer. A style-aware content loss for real-time hd style transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 698–714, 10 2018.
- [141] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. Styleclipdraw: Coupling content and style in text-to-drawing translation. *arXiv preprint arXiv:2202.12362*, 2022.
- [142] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020.
- [143] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1532–1540, 2021.
- [144] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879, 2024.
- [145] Kihyuk Sohn, Nataniel Ruiz, Kimin Lee, Daniel Castro Chin, Irina Blok, Huiwen Chang, Jarred Barber, Lu Jiang, Glenn Entis, Yuanzhen Li, et al. Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983*, 2023.
- [146] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020.

- [147] Alex Tamkin, Dan Jurafsky, and Noah D. Goodman. Language through a prism: A spectral approach for multiscale language representations. In *NeurIPS*, 2020.
- [148] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [149] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- [150] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151, 2020.
- [151] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Trans. Graph.*, 40(4), July 2021.
- [152] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6924–6932, 2017.

- [153] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [154] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020.
- [155] Binxu Wang and Carlos R Ponce. A geometric analysis of deep generative image models and its applications. In *International Conference on Learning Representations*, 2021.
- [156] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.
- [157] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.
- [158] Phil Wang. <https://github.com/ai-forever/ru-dalle>.
- [159] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18359–18369, 2023.
- [160] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, et al.

- Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499, 2024.
- [161] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yuezhe Wang, Zhen Li, Qiyang Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.
- [162] Zhizhong Wang, Zhanjie Zhang, Lei Zhao, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Aesust: Towards aesthetic-enhanced universal style transfer. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*, 2022.
- [163] Zhizhong Wang, Lei Zhao, Haibo Chen, Lihong Qiu, Qihang Mo, Sihuan Lin, Wei Xing, and Dongming Lu. Diversified arbitrary style transfer via deep feature perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7789–7798, 2020.
- [164] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, may 1992.
- [165] Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.
- [166] Zijie Wu, Zhen Zhu, Junping Du, and Xiang Bai. Ccpl: Contrastive coherence preserving loss for versatile style transfer. In *European Conference on Computer Vision*, pages 189–206. Springer, 2022.
- [167] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceed-*

- ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021.
- [168] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedi-gan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2256–2265, 2021.
- [169] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Towards open-world text-guided face image generation and manipulation. *arxiv preprint arxiv: 2104.08910*, 2021.
- [170] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.
- [171] Zipeng Xu, Tianwei Lin, Hao Tang, Fu Li, Dongliang He, Nicu Sebe, Radu Timofte, Luc Van Gool, and Errui Ding. Predict, prevent, and evaluate: Disentangled text-driven image manipulation empowered by pre-trained vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18229–18238, 2022.
- [172] Zipeng Xu, Enver Sangineto, and Nicu Sebe. Stylerdalle: Language-guided style transfer using a vector-quantized tokenizer of a large-scale generative model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7601–7611, October 2023.
- [173] Zipeng Xu, Songlong Xing, Enver Sangineto, and Nicu Sebe. Spectralclip: preventing artifacts in text-guided style transfer from a spec-

- tral perspective. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5121–5130, 2024.
- [174] Fei Yang, Shiqi Yang, Muhammad Atif Butt, Joost van de Weijer, et al. Dynamic prompt learning: Addressing cross-attention leakage for text-based image editing. *Advances in Neural Information Processing Systems*, 36:26291–26303, 2023.
- [175] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13789–13798, 2021.
- [176] Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang. Attention-aware multi-stroke style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1467–1475, 2019.
- [177] Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang. Attention-aware multi-stroke style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [178] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.
- [179] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

- [180] Kai Zhang, Lingbo Mo, Wenhui Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.
- [181] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. *CVPR 2021*, 2021.
- [182] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9026–9036, 2024.
- [183] Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, and Jimei Yang. Multimodal style transfer via graph cuts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5943–5951, 2019.
- [184] Zhongping Zhang, Jian Zheng, Zhiyuan Fang, and Bryan A Plummer. Text-to-image editing by image information removal. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5232–5241, 2024.
- [185] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020.
- [186] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.