Co-Painter: Fine-Grained Controllable Image Stylization via Implicit Decoupling and Adaptive Injection

Supplementary Material

7. Summary

This paper provides an in-depth analysis of the urgent need for fine-grained image stylization in practical applications. Considering the limitations of existing methods in handling fine-grained image stylization tasks, we propose a novel model, **Co-Painter**. This model can finely decouple the attributes of reference images implicitly and adaptively inject them into the diffusion model. Extensive experiments demonstrate that the proposed model achieves an optimal balance between text alignment and style similarity to reference images in standard and fine-grained settings.

In this material, we provide further elaboration on the key aspects discussed in the paper. We provide additional details on various aspects of our work. First, we elaborate on the implementation details of **Co-Painter** and baselines (Sec.8). Second, we describe the construction process of the quantitative evaluation metrics employed in this study (Sec.9). Third, further information about the dataset construction is provided (Sec.10). Fourth, we present more visual examples to demonstrate the effectiveness of our method (Sec.11). Fifth, we compare various feature injection mechanisms and offer a deeper analysis of **Co-Painter**'s performance (Sec.12). Sixth, in Sec.13, we evaluated the impact of the quantity of attribute terms on the model's generalization. Seventh, we show the results of combining **Co-Painter** with other controllable models (Sec.14). Eighth, we conducted a user study to quantitatively assess user satisfaction with image stylization (see 15). Ninth, we analyze the societal impact of **Co-Painter** and provide a detailed statement on its safety considerations (Sec.16). Finally, we discuss the limitations of our work and outline potential directions for future research (Sec.17).

8. Model Implementation Details

8.1. Preliminary

This paper employs the stable diffusion [6] as the basic image generation model. It leverages a pre-trained perceptual compression model, consisting of an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$, to map the complex sample features from pixel space into a low-dimensional space focused on the essential semantic components of the data. In this low-dimensional space, the model simulates the diffusion process of the data, gradually learning the original data distribution $p(z)$ from random Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. This entire process can be modeled as learning the reverse process of a fixed Markov chain of length T, where the noise at each step is predicted by a denoising encoder $\epsilon_\theta(z_t, t)$, $t = 1, \dots, T$. Here, z_t represents the low-dimensional representation z_0 of the input sample x_0 after adding Gaussian noise at a specified proportion over different time steps t. Additionally, a text encoder τ_Θ and a cross-attention mechanism are employed to incorporate the text y into the attention modules of the UNet denoising model, enabling text-controlled image generation:

$$\begin{cases} \textbf{Attention}(Q, K, V) = \textbf{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V, \\ Q = W_Q^{(i)} \cdot \varphi_i\left(z_t\right), K = W_K^{(i)} \cdot \tau_\theta(y), V = W_V^{(i)} \cdot \tau_\theta(y), \end{cases} \tag{1}$$

where, $\varphi_i\left(\cdot\right)$ represents the mapping of intermediate latent features z_t within the denoising model ϵ_θ. $\tau_\theta(\cdot)$ refers to the text encoder. $W_Q^{(i)}$, $W_K^{(i)}$ and $W_V^{(i)}$ are learnable linear layers. The entire model is optimized under the given input condition c using the Mean Squared Error (MSE) loss function.

$$L_{\text{LDM}} = \mathbb{E}_{\mathcal{E}(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1}), \boldsymbol{c}, t} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \boldsymbol{c}, t\right) \right\|_2^2 \tag{2}$$

Fine-grained Controllable Image Stylization (FCIS) aims to effectively decouple the rich attributes from reference images and then use these decoupled attributes as guided conditions to generate the target image. FCIS differs from the original text-to-image diffusion model in the following three aspects:

First, in the FCIS task, the model inputs are extended to include the text y and multiple reference images $(i_1, i_2, \dots, i_N)$, where N represents the number of image conditions. Second, these image conditions are fine-grained and decoupled into multiple conditional embeddings $(c_i^1, c_i^2, \dots, c_i^N)$. Finally, these conditions are adaptively fused with the text condition c_y, and through cross-attention, they collectively influence the diffusion process of the model.

8.2. Model Encoder Selection

Base Framework. We employed Stable Diffusion v1.5 [6] as the base model for **Co-Painter**. To retain the rich knowledge from the pre-trained model, the UNet and text encoder (the ViT-L/14 from CLIP [5]) were frozen. For the image encoder, the pre-trained IP-Adapter [9] was used to initialize the parameters of the image projection and the linear layers in the 16 cross-attention layers. Notably, the image projection is untrainable. Additionally, the image encoder (the ViT-L/14 from CLIP [5]) was adopted to extract the global information of multiple reference images.

	Model	Type	Content image	Training (fine-tuning)	Inference Memory(MB)	Inference Time(s)
Standard	InST [12]	inversion based	✓	✓	11331	3.29
	CAST [11]	conventional based	✓	✗	10443	0.01
	StyleTr2 [2]	conventional based	✓	✗	3527	0.02
	T2I-Adapter [3]	diffusion based	✗	✓	12119	3.11
	IP-Adapter [9]	diffusion based	✗	✓	6279	1.74
	DEADiff [4]	diffusion based	✗	✓	11967	1.87
	Co-Painter(Ours)	diffusion based	✗	✓	6373	1.83
Fine-Grained	T2I-Adapter [3]	controllable diffusion	✗	✗	10745	3.18
	ControlNet [10]	controllable diffusion	✗	✗	5701	2.22
	IP-Adapter [9]	stylized diffusion	✗	✓	6265	1.68
	DEADiff [4]	stylized diffusion	✗	✓	7895	1.83
	T2I-Adapter(stacked) [3]	stylized diffusion	✗	✓	13555	6.85
	IP-Adapter(stacked) [9]	stylized diffusion	✗	✓	6370	1.70
	Co-Painter(Ours)	stylized diffusion	✗	✓	6475	1.81

Table 1. Model implementation details and inference efficiency on 1 RTX 3090 GPU.

Proposed Modules. For the fine-grained decoupling module, each module consists of a linear layer and a LayerNorm layer. For the gated feature injection module, we constructed a feed-forward network using two linear layers and a GLUE [8] activation function. Furthermore, all gated parameters were initialized to 0 to ensure the stable convergence of the training process.

Benefits. Through these strategies, we can effectively preserve the rich knowledge of the pre-trained Stable Diffusion [6] and IP-Adapter [9] models, facilitating fast transfer and seamless adaptation.

8.3. Computational Resource

During the training phase, we utilized 4 Nvidia RTX 3090[1] GPUs to train the model on the fine-grained style dataset build in this paper. The training lasted for approximately 40 hours. The batch size was set to 4, and the total GPU memory usage was around 56,800 MB. The entire training framework was built upon and further updated from the IP-Adapter[2] [9].

8.4. Hyper-parameters

First, for the division between the coarse and fine layers, we follow DEADiff [4] and number the 16 cross-attention layers of Stable Diffusion [6] from 0 to 15. Among these, layers 4-8 are designated as coarse layers, into which content information is injected. Correspondingly, the remaining layers are defined as fine layers, receiving fine-grained style information. Second, the context tokens for content, brushstroke, and color embeddings are set to 4, and the latent feature dimension of the feedforward layers is set to four times the input dimension. Lastly, for the inference process, we introduce a negative prompt (*text, watermark, low-res, low quality, worst quality, deformed, glitch, low contrast, noisy, saturation, blurry*) to enable classifier-free guidance in image generation. The guidance scale is set to 5.0, and the number of inference steps is 30.

8.5. Efficiency

For testing, we evaluated the inference memory and 30-step inference time of baselines and **Co-Painter** on 1 RTX 3090 GPU. As shown in the Table 1, traditional image stylization methods (CAST [11] and StyleTr2 [2]) have shorter inference times and relatively lower memory consumption. However, their performance in image stylization is suboptimal. In contrast, diffusion-based methods (T2I-Adapter-Style [3], IP-Adapter [9], T2I-Adapter(stacked) [3], IP-Adapter(stacked) [9], DEADiff [4], and ControlNet [10]) and inversion based method (InST [12]) demonstrate stronger stylization capabilities, but due to iterative diffusion process, they tend to have longer inference times. The proposed **Co-Painter** is a fine-grained image stylization approach based on diffusion models. Benefiting from its lightweight image encoder, **Co-Painter** achieves superior results in inference memory and inference time compared to others. Compared to IP-Adapter [9], our model introduces only a minor increase in inference memory and inference time. This slight increase is justified as it provides optimal fine-grained control for image stylization tasks.

8.6. Baselines Implementation Details

In this section, we present the implementation details of the comparative methods under both standard and fine-grained settings (see Table 1).

[1] https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3090-3090ti/

[2] https://github.com/tencent-ailab/IP-Adapter

Standard Image Stylization Settings. For InST [12], we input each style reference image from the test set into the fine-tuning framework to extract the inverted feature embeddings. These style feature embeddings are then fed into the diffusion model, along with the text descriptions, to perform style transfer on the content images. For CAST [11] and StyleTr2 [2], we loaded the pre-trained models provided by the authors for evaluation. For T2I-Adapter [3] (Style) and IP-Adapter [9], we fine-tuned these two models on the dataset proposed in this paper using a reconstruction-based training strategy. For DEADiff [4], we fine-tuned this model using a non-reconstruction-based training strategy. Following the original settings, we injected the content reference image and style reference image into the coarse and fine layers, respectively.

Fine-Grained Image Stylization Settings. For T2I-Adapter [3], we used the composable adapters (Canny & Color) to perform style attribute transfer on the brushstroke image and the color image, respectively. For ControlNet [10], we achieved the same goal using the Color-Canny-ControlNet model[3] fine-tuned by Ghoskno *et al.* on the laion-art-en-colorcanny[4] dataset. Accordingly, we fine-tuned all image stylization methods on the dataset built in this paper. For IP-Adapter [9], a single image encoder processes the three reference images to derive latent representations, which are then concatenated and fed into the diffusion model through the decoupled cross-attention module, as outlined in the original methodology, to guide feature adaptation effectively. In contrast, IP-Adapter (stacked) [9] employs three stacked image encoders, each processing one reference image independently before concatenating the representations and injecting them into the diffusion model via the same cross-attention mechanism. For T2I-Adapter(stacked) [3], three stacked style encoders are employed to decouple multiple fine-grained image attributes separately. These attributes ate then concatenated with the text and fed into the diffusion model to guide image stylization. For DEADiff [4], we control the brushstroke and color reference images through the style Q-former while directing the content reference image into the content Q-former. The extracted latent representations are subsequently disentangled and infused into the diffusion model's coarse and fine layers as per the established method.

9. Evaluation Metrics

In this section, we provide a detailed introduction to the evaluation metrics used in this paper. We employed three metrics proposed by DEADiff [4] to assess the stylization results: Style Similarity (SS), Text Alignment (TA), and Image Quality (IQ). In addition, we introduced three new metrics based on cosine similarity, Content Alignment (CONA), Brushstroke Similarity (BSTS), and Color Similarity (COLS), to evaluate the fine-grained image stylization performance of the model. The details of these metrics are as follows:

9.1. Text Alignment

Text Alignment (TA) aims to assess the consistency between the generated image and the given text description. We use the cosine similarity in the CLIP [5] (ViT-L/14) text-image alignment space to evaluate the degree of alignment between the text prompt and the stylized image.

9.2. Style Similarity

Style Similarity (SS) is designed to evaluate the visual style similarity between the generated image and the reference image. First, we use CLIP Interrogator 2[5] to generate the best text prompt aligned with the reference image. Next, we remove vocabulary related to the content of the reference image to obtain a style-specific prompt. Finally, we use the cosine similarity from CLIP [5] (ViT-L/14) to assess the style similarity between the stylized image and the reference image. It is worth noting that for the fine-grained image stylization experiments, our style prompt consists of brushstroke and color descriptions: "a {color} image in {brushstroke} style."

9.3. Image Quality

Image Quality (IQ) aims to quantitatively assess the visual quality of the images. We utilize the LAION-Aesthetics Predictor V2[6] to evaluate the aesthetic quality of the images generated by different methods.

9.4. Content Alignment

CONtent Alignment (CONA) aims to assess the consistency between the generated image and the given content prompt. First, we extract the content-descriptive terms from the instruction prompt, such as "dog," "cat," etc. Next, we construct a content-specific prompt using a template: "an image of {content}." Finally, we evaluate the content alignment by calculating the cosine similarity in the CLIP [5] (ViT-L/14) alignment space.

9.5. Brushstroke Similarity

BrushSTroke Similarity (BSTS) aims to evaluate the similarity in brushstroke details between the generated image and the reference image. First, we extract brushstroke-related descriptive terms from the reference image, such as "Cartoon" or "Vincent Van Gogh." Next, we construct a brushstroke-specific prompt using the template: "a

[3] https://huggingface.co/ghoskno/Color-Canny-Controlnet-model

[4] https://huggingface.co/datasets/ghoskno/laion-art-en-colorcanny

[5] https://github.com/pharmapsychotic/clip-interrogator

[6] https://github.com/christophschuhmann/improved-aesthetic-predictor

{brushstroke} style image." Finally, we assess the consistency of brushstroke attributes by calculating the cosine similarity in the CLIP [5] (ViT-L/14) alignment space.

9.6. Color Similarity

COLor Similarity (COLS) aims to evaluate the similarity in color details between the generated image and the reference image. First, we extract color-related descriptive terms from the reference image, such as "purple, pink, and gray" or "white, blue, and green." Next, we construct a color-specific prompt using the template: "a {color} image." Finally, we assess the consistency of color attributes by calculating the cosine similarity in the CLIP [5] (ViT-L/14) alignment space.

10. Details in Dataset

Overall. In this section, we provide detailed information on the dataset created in this study. These details include the lexical repository, GPT-Prompts, build costs and time, dataset distribution, as well as data filtering and checking processes. The details are as follows:

10.1. Lexical Repository Building

To construct the text prompts for the generative model, we gathered 8 types of typical style terms, more than 20 color combinations, and about 130 commonly used content terms. Table 2 shows the entire lexical repository. These terms are randomly selected during the data construction to generate detailed text prompts. This randomized combination of terms ensures the diversity of data samples.

10.2. Text Caption Generation via ChatGPT

We used ChatGPT v4 [1] to combine different terms to generate detailed text captions. During this process, we randomly selected 1 instruction from Table 3 to guide GPT [1], enhancing data diversity and preventing the creation of similar or repetitive image samples.

10.3. Image synthesis via Midjourney

After obtaining a detailed description of the image, we used Midjourney v6.0[7] to perform high-quality image synthesis (see Figure 1). First, the text captions generated by ChatGPT [1], combined with brushstroke terms, were input into the Midjourney to synthesize 4 image samples. Next, the samples that align with input instructions were upsampled. Finally, following manual checking, a high-quality image sample set with diverse brushstrokes was constructed.

10.4. Image synthesis via ControlNet

To construct samples with varying colors, we employed ControlNet [10] for color transformation. First, different color combinations, image brushstroke terms, and the image caption were combined to form new prompt instructions (e.g., "a {colors} image in {brushstroke} style, {Caption}, high-quality, extremely detailed, 4K"). Second, the canny edge map of the original image was used as an additional prompt to retain the structural information. Finally, image samples with district color levels and Various brushstrokes were obtained. Sample examples from the dataset are shown in Figure 2.

10.5. Data Filtering & Checking

In the data construction process, we combined CLIP [5] with manual supervision to filter and check the Synthesized images, ensuring data accuracy and diversity. On the one hand, the 6 evaluation metrics established in Sec.9 are used to assess various aspects of the generated paired sample data, which ultimately led to the filtering of approximately 20% of anomalous data. On the other hand, we conducted manual supervision and checks at each stage. This process primarily involved data summarization, correction of textual errors, and removal of anomalous data.

10.6. Build Cost and Time

We listed the cost and time required to construct the dataset (see Table 4). A data processing team of 5 members was assembled to organize and review all procedures. Most steps were automated, and each team member contributed approximately 30 hours, processing over 50,000 data pairs. The entire dataset creation process took a total execution time of around 104 hours.

10.7. Data Distribution

To demonstrate the diversity and correlation structure within our dataset, PCA dimensional reduction was employed to conduct a statistical analysis of the data distribution. Figures 3a and 3b illustrate the distribution differences across varied brushstrokes and colors, respectively. These findings indicate that our constructed dataset encompasses a rich variety of samples and displays a significant correlation between brushstroke and color dimensions. This structure provides a solid foundation for subsequent model training and style transfer tasks, ensuring that these complex structural features are fully accounted for during image stylization. Furthermore, the distributional variations within the dataset offer a richer style space, thereby enhancing the model's capacity for fine-grained control in image generation.

11. Visualization

To demonstrate the effectiveness of **CO-PAINTER**, we present more visual results in standard and fine-grained image stylization settings (see Table 5 Figure 4, Figure 5, Figure 6, and Figure 7).

[7] https://www.midjourney.com

Brushstrokes #8				
cartoon	children illustration	ink wash painting	miyazaki hayao	oil painting
photo	pixel	vincent van gogh		
Colors #23				
black, white	blue	blue, white, orange	brown	gray, orange, green
green	orange	pink	pink, blue, green	pink, blue, yellow
pink, white, green	purple	purple, cyan, yellow	purple, pink, gray	red
red, green, blue	red, pink, green	red, yellow, blue	red, yellow, green	white
white, blue, green	yellow	yellow, green, blue		
Contents #133				
airplane	ant	apple	backpack	baseball
basketball	beach	bear	bee	bench
bicycle	bird	boat	bookshelf	bottle
boy	bridge	building	bus	butterfly
camera	canoe	car	cat	cave
chair	chicken	child	classroom	cliff
clouds	computer	coral reef	cow	crab
cucumber	cup	deer	desert	dog
dragon	duck	elephant	farm	fence
fish	fisherman	fishing boat	flower	flowers
forest	fox	frog	fruits	garden
giraffe	girl	glacier	grass	hedgehog
helicopter	hill	horse	house	island
kangaroo	kayak	lake	lamp	lantern
leaf	library	lion	lychee	man
mango	monkey	moon	moon and stars	motorcycle
mountain	oasis	ocean	office	old man
palace	panda	panda	panda	pig
plate	playground	potato	rabbit	rain
river	road	sailboat	seashell	sheep
shell	shrimp	snake	snow	soccerball
sofa	spider	squirrel	star	starfish
stars	street	sun	table	telescope
tennis	tent	tiger	tomato	train
tree	truck	turtle	umbrella	umbrella
vegetable	villa	waterfall	wave	window
woman	yacht	zebra		

Table 2. Lexical repository. It includes 8 typical style terms, 23 color combinations, and 133 commonly used content terms.

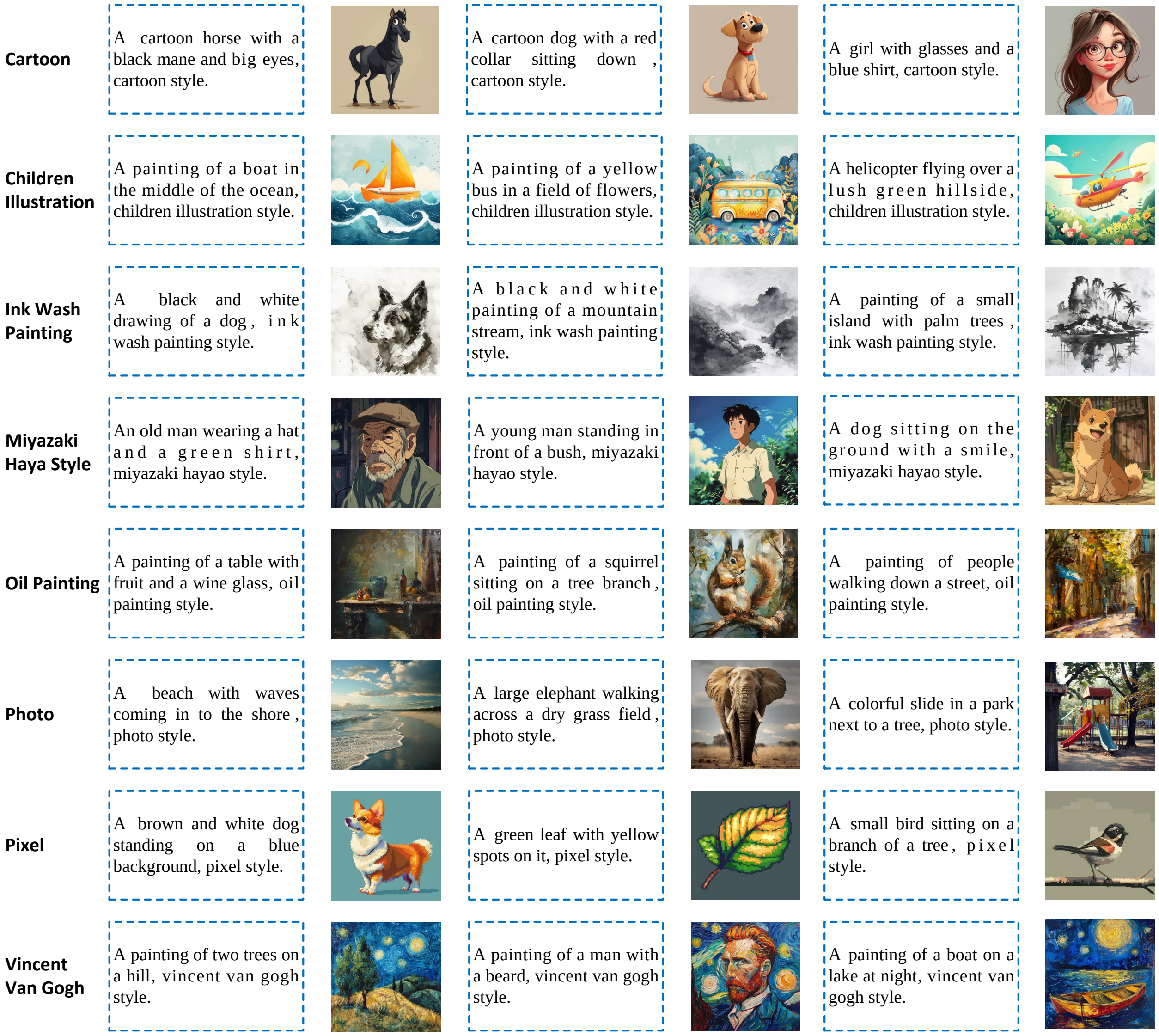
conducted 2 aspects of evaluation: **1)** We present 2 groups of results where the same prompt is used for all 9 generated images(**(a) and (b) in the above Figure 8**). Comparing (a) and (b) shows the model's performance under different prompts with consistent brushstroke and color. It can be observed that **CO-PAINTER** effectively captures variations in conditions. **2)** To evaluate the controllability of each branch/attribute (i.e., color or brushstroke) that operates independently, we use images only to guide 1 condition while varying the other based on prompts(**(c), (d) in above Figure 8**). And, the result generated solely by text prompts was present (**(e) in above Figure 8**). Both of the above results validate the sufficient disentanglement of multiple attributes.

12. Comparison of Different Feature Injection Mechanisms

In this section, we compared the impact of different feature injection mechanisms on image stylization. Specifically, DCM refers to the disentangled conditioning mechanism proposed in DEADiff [4], DCA refers to the decoupled cross-attention feature injection mechanism introduced in IP-Adapter, and GFI represents the gated feature injection

GPT Prompts #10
1. The content and brushstroke of the image are: {content}, {brushstroke}, generate a detailed text prompt.
2. Based on the image's content {content} and its brushstroke {brushstroke}, create a detailed prompt.
3. Describe the image by emphasizing the target content {content} and distinctive brushstroke style {brushstroke}.
4. Construct a detailed image prompt that highlights both the content {content} and the {brushstroke} brushstroke.
5. Using the target {content} and brushstroke {brushstroke}, write a descriptive prompt for the image.
6. Generate a prompt focusing on the {content} and stylistic {brushstroke} elements in the image.
7. Create a descriptive prompt that captures the specific {content} and {brushstroke} brushstroke of the image.
8. Write a text prompt emphasizing the image's target {content} along with its unique {brushstroke} style.
9. Generate a prompt that elaborates on both the desired {content} and artistic {brushstroke} features of the image.
10. Write a detailed prompt centered around the {content} and its {brushstroke} brushstroke.

Table 3. GPT prompt templates. We randomly select 1 template to prompt GPT to generate detailed image captions.



Cartoon	A cartoon horse with a black mane and big eyes, cartoon style.	A cartoon dog with a red collar sitting down , cartoon style.	A girl with glasses and a blue shirt, cartoon style.
Children Illustration	A painting of a boat in the middle of the ocean, children illustration style.	A painting of a yellow bus in a field of flowers, children illustration style.	A helicopter flying over a lush green hillside, children illustration style.
Ink Wash Painting	A black and white drawing of a dog , ink wash painting style.	A black and white painting of a mountain stream, ink wash painting style.	A painting of a small island with palm trees , ink wash painting style.
Miyazaki Haya Style	An old man wearing a hat and a green shirt, miyazaki hayao style.	A young man standing in front of a bush, miyazaki hayao style.	A dog sitting on the ground with a smile, miyazaki hayao style.
Oil Painting	A painting of a table with fruit and a wine glass, oil painting style.	A painting of a squirrel sitting on a tree branch , oil painting style.	A painting of people walking down a street, oil painting style.
Photo	A beach with waves coming in to the shore , photo style.	A large elephant walking across a dry grass field , photo style.	A colorful slide in a park next to a tree, photo style.
Pixel	A brown and white dog standing on a blue background, pixel style.	A green leaf with yellow spots on it, pixel style.	A small bird sitting on a branch of a tree , pixel style.
Vincent Van Gogh	A painting of two trees on a hill, vincent van gogh style.	A painting of a man with a beard, vincent van gogh style.	A painting of a boat on a lake at night, vincent van gogh style.

Figure 1. Image Examples with various brushstrokes.

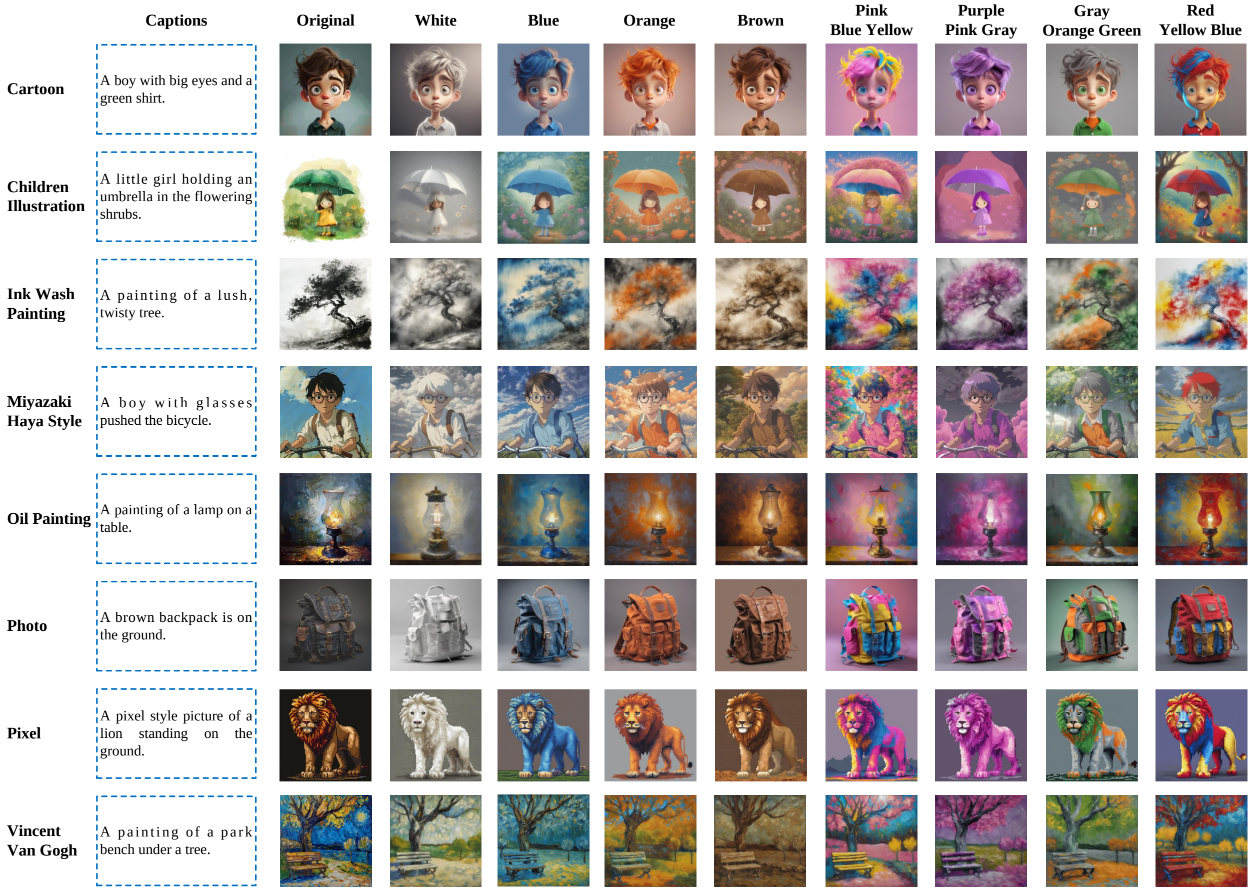

Captions
Original
White
Blue
Orange
Brown
Pink Blue Yellow
Purple Pink Gray
Gray Orange Green
Red Yellow Blue
Cartoon
A boy with big eyes and a green shirt.
Children Illustration
A little girl holding an umbrella in the flowering shrubs.
Ink Wash Painting
A painting of a lush, twisty tree.
Miyazaki Haya Style
A boy with glasses pushed the bicycle.
Oil Painting
A painting of a lamp on a table.
Photo
A brown backpack is on the ground.
Pixel
A pixel style picture of a lion standing on the ground.
Vincent Van Gogh
A painting of a park bench under a tree.

Figure 2. Some image examples with various brushstrokes and colors.

	Lexical Repository	Text Caption	Brushstroke Image	Color Image	Filtering & Checking
Total Time	≈ 8h	≈ 6h	≈ 24h	≈ 36h(8 GPUs)	≈ 30h
Response Time	/	10s/iteration	30s/iteration	20s/iteration	/
Resource	ChatGPT[1] & Labor	ChatGPT[1]	Midjourney	ControlNet[10]	CLIP[5] & Labor

Table 4. Dataset building cost and time.” The ”Total time” represents the overall time cost for each stage, while ”Response time” indicates the average response time across different models. ”Resource” refers to the resources required for each stage. The total time spent on building the dataset is approximately 104 hours.

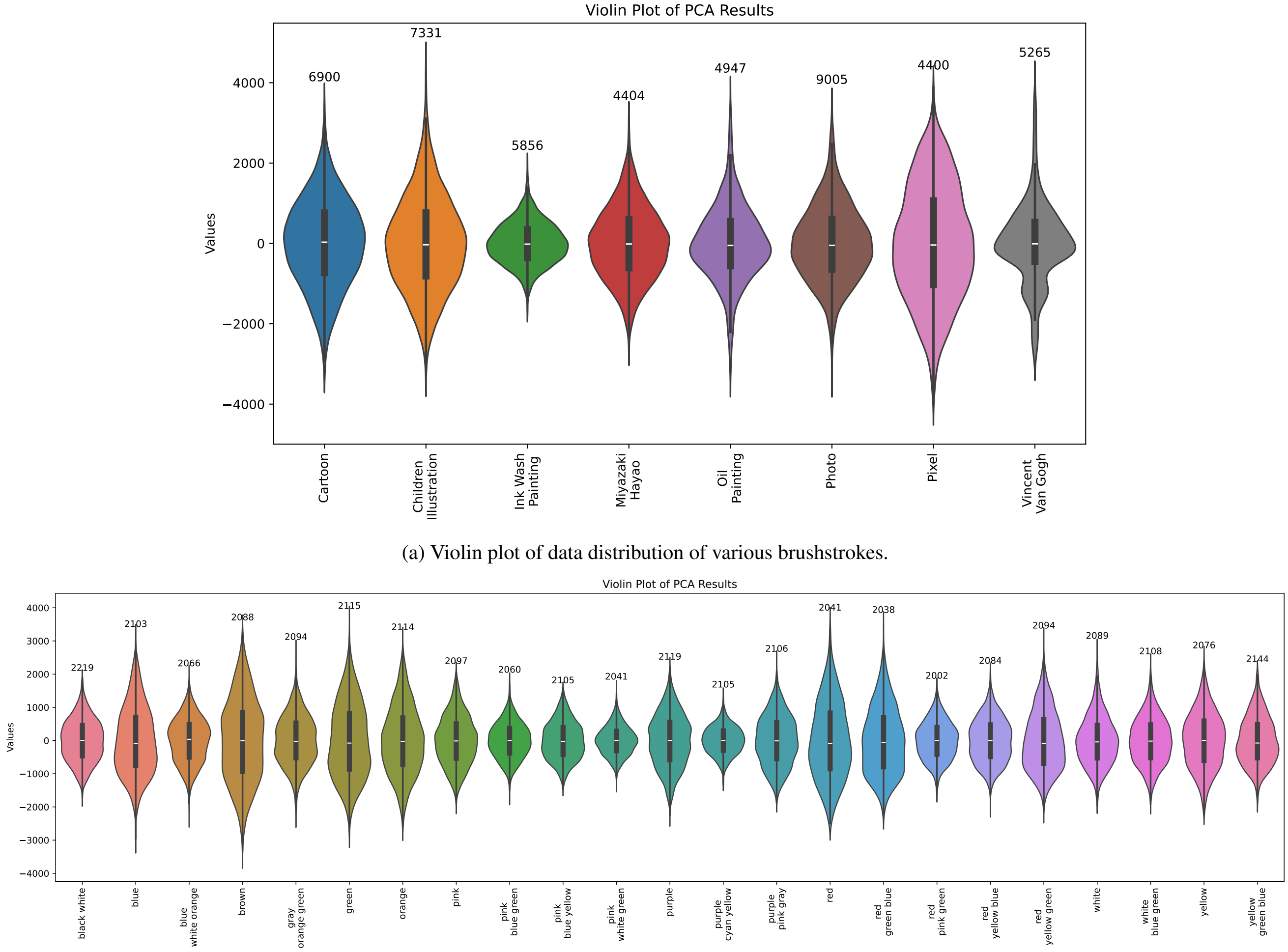
Methods	IQ↑	SS↑	TA↑	LPIPS↓	Chamfer
StyleDrop	5.86	27.4	23.3	0.797	0.074
StyleStudio	5.84	26.2	24.2	0.786	0.068
StyleAlign	6.06	27.5	23.7	0.785	0.074
CSGO	5.95	27.3	24.3	0.789	0.073
InstantStyle	5.96	31.4	20.4	0.806	0.078
Rb-modulation	6.05	27.5	23.9	0.788	0.069
Co-Painter(Ours)	6.14	27.8	24.5	0.785	0.065

Table 5. he quantitative evaluation of more baselines.

Method	IQ↑	SS↑	TA↑
Baseline	6.07	**36.9**	14.3
+DCA	5.93	32.5	23.0
+DCM	5.94	31.9	23.2
+GFI	5.96	31.6	23.3
+FGD & GFI	**6.14**	27.8	**24.5**

Table 6. The quantitative evaluation of different feature injection mechanisms.

mechanism proposed in this work. To ensure a fair comparison, we injected the image embeddings extracted by the image encoder into the fine layers of the diffusion model using these three feature injection mechanisms, rather than



(a) Violin plot of data distribution of various brushstrokes.

(b) Violin plot of data distribution of various colors.

Figure 3. We use PCA dimensional reduction to measure the feature distribution of different datasets, where there are significant differences in the feature distribution.

Method	IQ↑	SS↑	TA↑	CONS↑	BSTS↑	COLS↑
3-BST&133CON	5.94	19.8	20.1	25.4	20.5	17.3
5-BST&133CON	6.01	<u>20.6</u>	20.2	25.4	21.3	<u>17.6</u>
8-BST&80-CON	6.01	<u>20.6</u>	20.2	<u>25.5</u>	**21.6**	17.5
8-BST&100-CON	<u>6.04</u>	20.2	<u>20.3</u>	25.3	20.8	<u>17.6</u>
8-BST&133CON	**6.06**	**20.9**	**20.4**	**25.6**	<u>21.4</u>	**17.9**

Table 7. Quantities evaluation of the effect of different numbers of brushstrokes (BST) and contents (CON) on model generalization.

injecting them into all layers. The selection strategy for the fine layers is identical to that used in DCM.

The experimental results reveal several interesting conclusions (refer to columns 3, 4, 5, and 6 of Figure 9 and rows 1, 2, 3, and 4 of Table 6). First, injecting the reference image features only into the fine layers can reduce the model's preference toward the reference images to some extent, thereby mitigating the issue of content leakage. All three feature injection strategies demonstrate varying degrees of performance improvement compared to the Baseline. Second, the DCA injection strategy, which simply adds text and image features, somewhat undermines the guiding capability of the text, leading to content information leakage from the reference images. Third, DCM enhances the importance of text prompts by adopting a feature concatenation strategy. However, the content attribute from

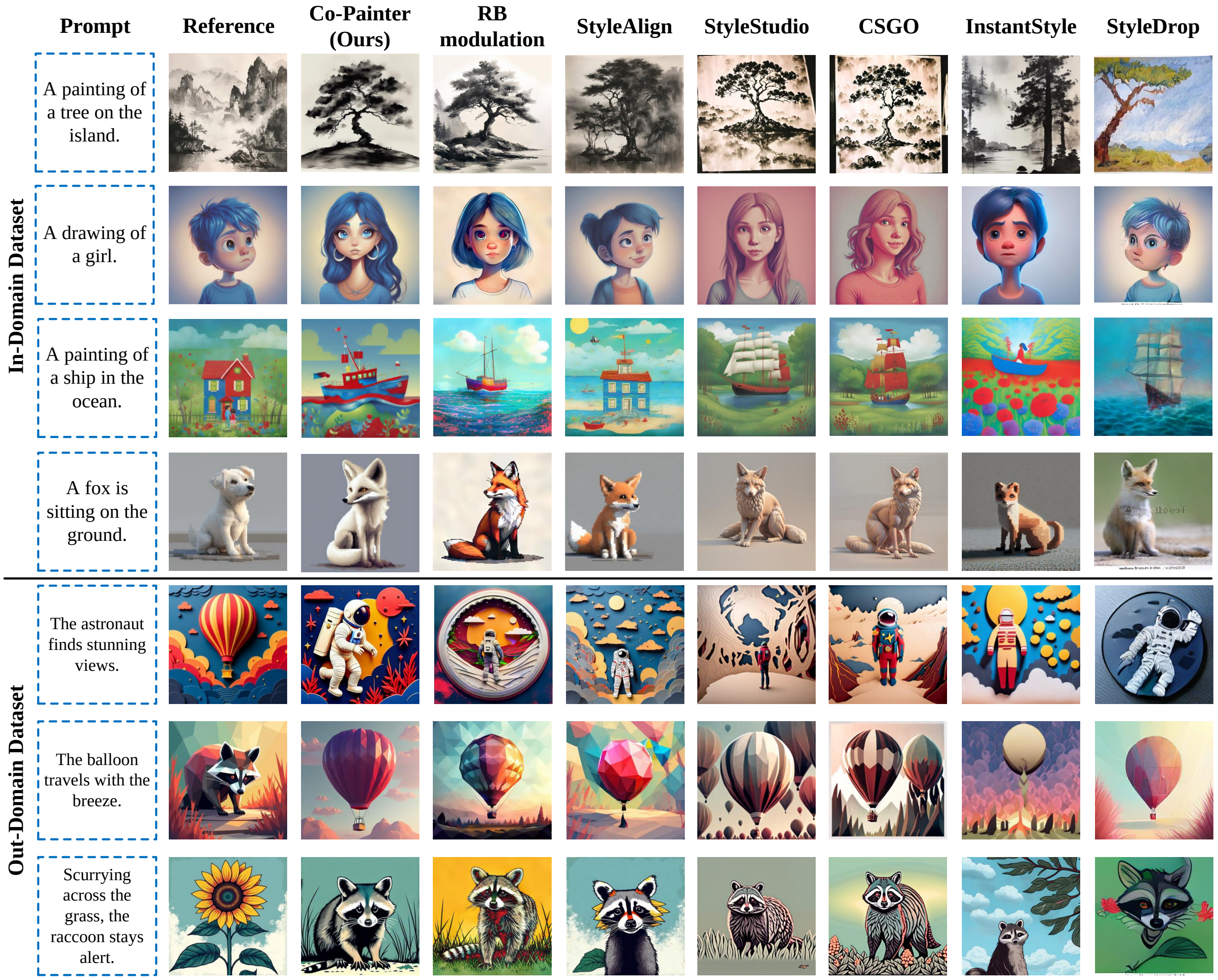

Prompt
Reference
Co-Painter (Ours)
RB modulation
StyleAlign
StyleStudio
CSGO
InstantStyle
StyleDrop
In-Domain Dataset
A painting of a tree on the island.
A drawing of a girl.
A painting of a ship in the ocean.
A fox is sitting on the ground.
Out-Domain Dataset
The astronaut finds stunning views.
The balloon travels with the breeze.
Scurrying across the grass, the raccoon stays alert.

Figure 4. Qualitative evaluation results compared with more baselines in fine-grained image stylization settings.

the reference image still conflicts with the text. Finally, GFI utilizes a gated mechanism to adaptively integrate text and image features, effectively addressing the issue of image content leakage. Additionally, its flexible feature injection strategy effectively adapted to the differences across various cross-attention layers, resulting in improvements in both image quality and text alignment. However, since the reference image attributes are not finely decoupled, its performance remains limited.

From Figure 9 (col 5) and Table 6 (row 3), we can observe that fine decoupling of reference image attributes significantly enhances image stylization performance. The FGD module provides the GFI with a clearer decoupled representation of image attributes, which effectively guides the GFI module to capture detailed information about different conditions and their collaborative relationships.

13. Impact of Attribute Term Quantity on Model Generalization

To assess the impact of attribute term quantities (brushstroke (BST) and content (CON)) on model generalization, we pre-trained **CO-PAINTER** on various training subsets and conducted few-shot fine-tuning and evaluation with OTD data. The results in Table 7 indicate that an increased quantity of both brushstrokes and content is beneficial for the model's generalization performance on OTD data.

14. Combine with Other Controllable Models

In this subsection, we present the combination results of **CO-PAINTER** with ControlNet [10] and Dreambooth [7] (see Figures 10, 11, and 12). It is observed that **CO-PAINTER** seamlessly integrates with other conditional dif-

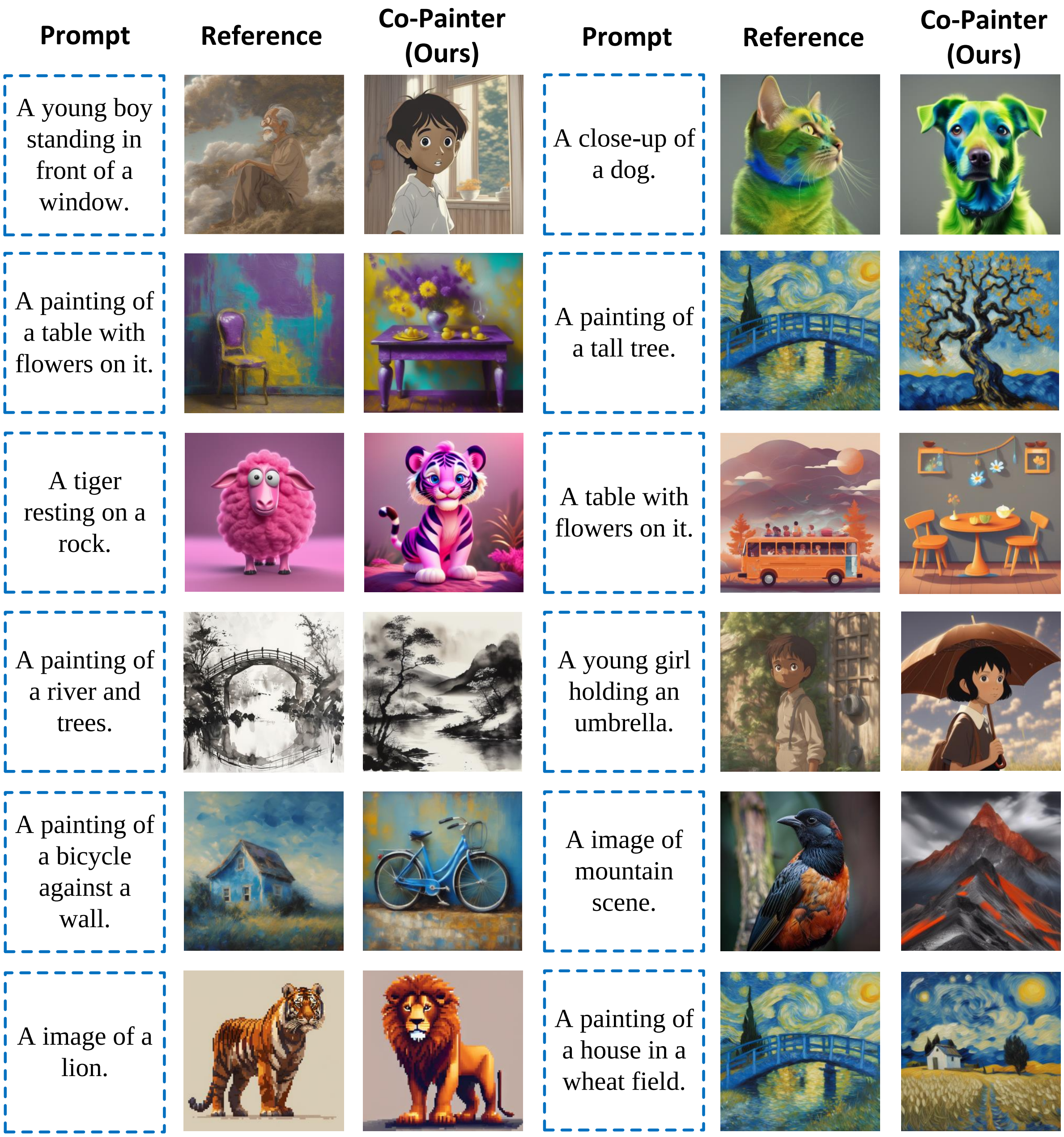

Prompt
Reference
Co-Painter (Ours)
Prompt
Reference
Co-Painter (Ours)
A young boy standing in front of a window.
A close-up of a dog.
A painting of a table with flowers on it.
A painting of a tall tree.
A tiger resting on a rock.
A table with flowers on it.
A painting of a river and trees.
A young girl holding an umbrella.
A painting of a bicycle against a wall.
A image of mountain scene.
A image of a lion.
A painting of a house in a wheat field.

Figure 5. More qualitative evaluation results in standard image stylization setting.

fusion models, achieving outstanding visual effects. For the combination with ControlNet [10], we can see that our model enables fine-grained control over the diffusion process regarding abstract style attributes, while ControlNet [10] provides precise structural and layout information. In the case of the integration with Dreambooth [7], the proposed model demonstrates its ability to perform fine-grained style transformations on customized appearance representations. This validates the strong adaptability of **CO-PAINTER** in image generation. The model not only maintains high-quality visual output but also flexibly supports personalized needs, advancing the progress of artistic creation.

15. User Study

To evaluate the subjective stylization effects of different methods and understand user satisfaction, we conducted a user study under both standard and fine-grained image stylization settings. First, we randomly selected 40 sample groups from the test set and applied various algorithms

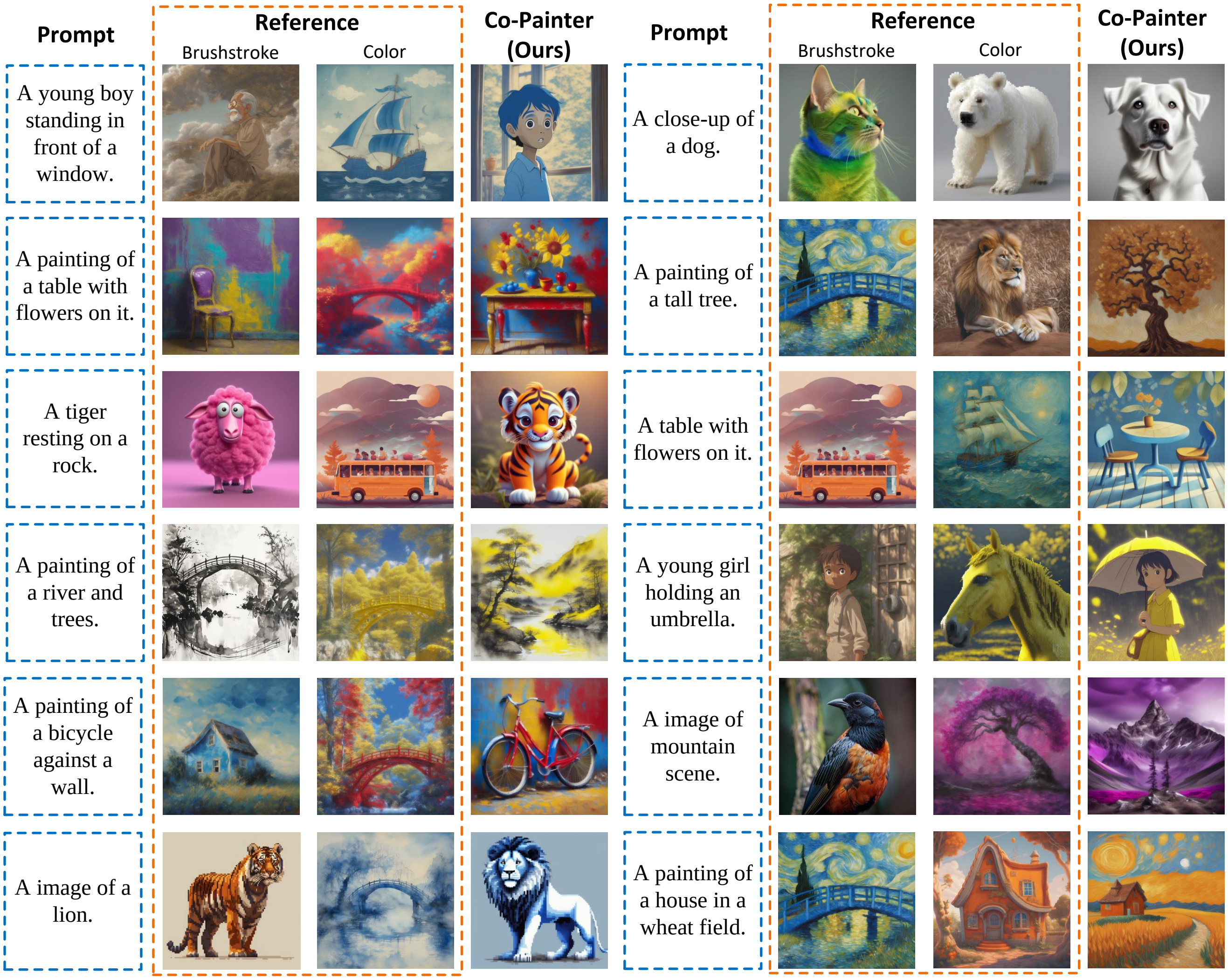

Prompt
Reference
Brushstroke
Color
Co-Painter (Ours)
Prompt
Reference
Brushstroke
Color
Co-Painter (Ours)
A young boy standing in front of a window.
A close-up of a dog.
A painting of a table with flowers on it.
A painting of a tall tree.
A tiger resting on a rock.
A table with flowers on it.
A painting of a river and trees.
A young girl holding an umbrella.
A painting of a bicycle against a wall.
A image of a mountain scene.
A image of a lion.
A painting of a house in a wheat field.

Figure 6. More qualitative evaluation results in fine-grained image stylization settings.

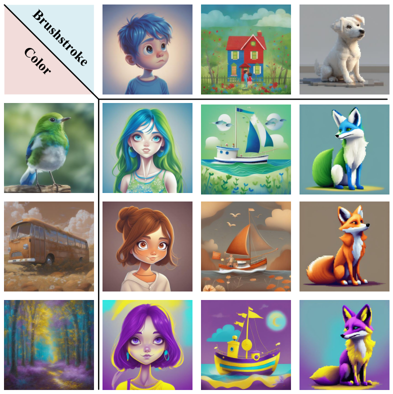
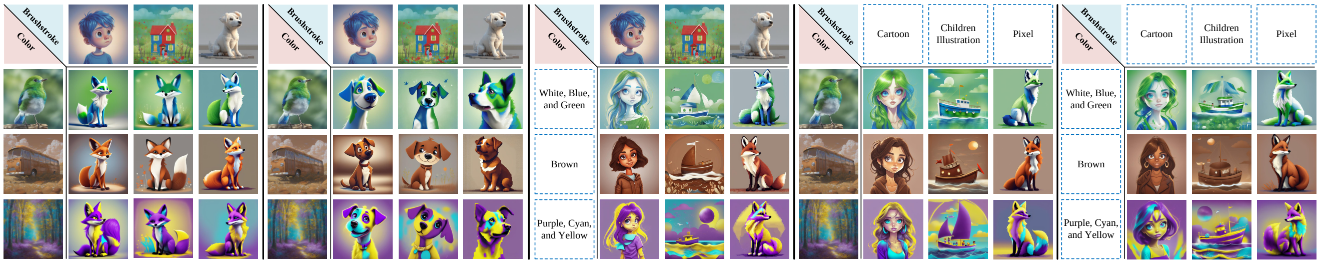


Figure 7. Visual comparative results for fine-grained image stylization.

Method	IQ↑	SS↑	TA↑	Overall↑
InST[12]	73.8	55.3	82.6	72.7
CAST[11]	70.3	71.4	**90.3**	81.8
StyTR-2[2]	72.2	69.5	88.9	83.3
T2I-Adapter[3]	83.2	92.6	40.3	56.8
IP-Adapter[9]	83.4	**93.7**	37.6	57.7
DEADiff[4]	85.2	75.2	88.7	83.0
CO-PAINTER(Ours)	**89.3**	90.3	88.4	**87.9**

Table 8. The user study results of the standard image stylization.

for image stylization. Next, the generated results were randomly shuffled, and two sets of evaluation forms were created. Finally, we recruited 18 volunteers with diverse backgrounds to rate the stylization outcomes. It is important to note that a percentage-based scoring system was used to



(a) Visual comparative results with same prompt.

(b) Visual comparative results with same prompt.

(c) Image reference are used for brushstroke and text prompt for color.

(d) Image reference are used for color and text prompt for brushstroke.

(e) Text prompts are used for both color and brushstroke.

Figure 8. Visual comparative results for more interesting applications.

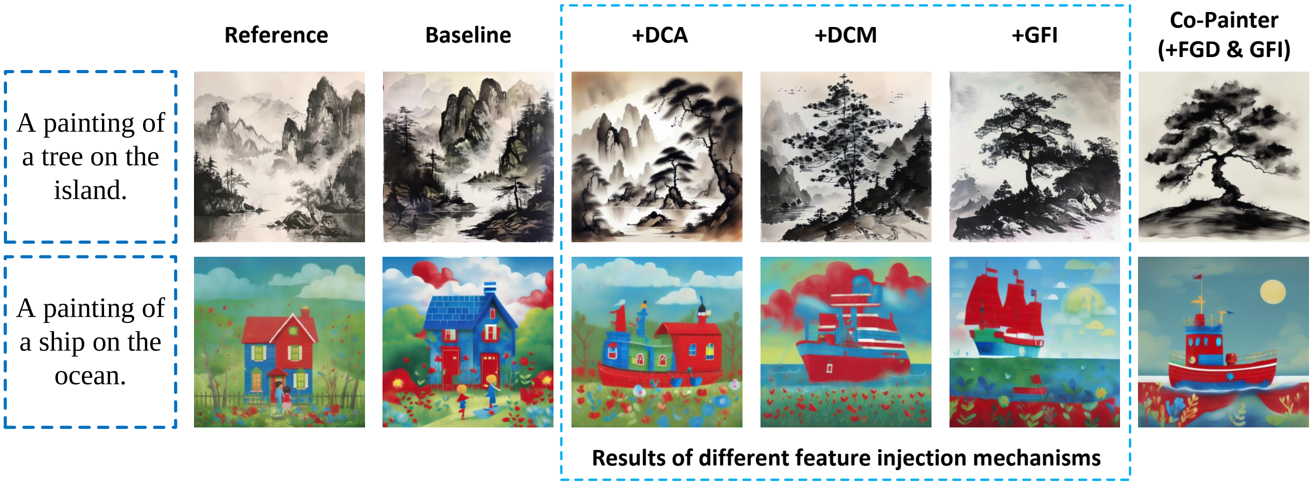

Reference
Baseline
+DCA
+DCM
+GFI
Co-Painter (+FGD & GFI)
A painting of a tree on the island.
A painting of a ship on the ocean.
Results of different feature injection mechanisms

Figure 9. Qualitative evaluation of different feature injection mechanisms.

evaluate the stylization results. Volunteers provided subjective scores based on several aspects, such as style similarity, text alignment, image quality, and overall evaluation.

Table 8 presents the results of the user study conducted under the standard image stylization setting. From the table, we can observe that the proposed **Co-Painter** achieved the highest overall user satisfaction (87.9) in terms of visual results. It also obtained competitive quantitative scores in style similarity, text alignment, and image quality. This indicates that, despite being a fine-grained controllable image stylization model, **Co-Painter** still delivered the best performance under the standard setting. In contrast, other methods exhibited a noticeable decline in user satisfaction to varying degrees.

Table 9 presents the evaluation results under the fine-grained image stylization setting. It can be observed that our model achieved leading performance across all metrics, with an overall user satisfaction score of 89.2. In contrast, other methods showed a significant decline in performance across multiple aspects, due to issues such as severe content leakage, brushstroke leakage, or color leakage.

Overall, these results indicate that our method provides superior image stylization performance from the users' perspective. It achieved satisfactory results in both standard and fine-grained image stylization tasks.

16. Social Compact

Positive Societal Impact. **Co-Painter** introduces innovative opportunities to the fields of art creation and design, enabling artists, designers, and creative professionals to efficiently and accurately generate stylized images, significantly enhancing productivity. Whether for beginners or experienced professionals, **Co-Painter** allows users to overcome the limitations of traditional creation methods, sparking greater creative expression. Additionally, **Co-Painter**'s seamless integration with other controllable image generation methods fosters cross-platform collaboration, driving the development of more sophisticated and versatile tools and pushing the deeper integration of art and technology.

Potential Negative Social Impacts. While **Co-Painter** has the potential to advance artistic creation, it may also introduce some negative consequences. First, with the widespread adoption of stylized image generation

Method	IQ↑	SS↑	TA↑	CONS↑	BSTS↑	COLS↑	Overall↑
ControlNet[10]	73.4	66.3	32.6	29.8	82.1	55.6	61.2
T2I-Adapter[3]	77.2	62.9	80.5	83.1	58.3	60.1	67.3
IP-Adapter[9]	83.6	81.0	53.9	52.6	81.3	62.5	70.1
DEADiff[4]	82.6	80.3	85.2	91.7	75.3	72.8	84.0
T2I-Adapter(stacked)[3]	80.5	86.7	83.4	90.5	79.6	85.2	83.6
IP-Adapter(stacked)[9]	82.9	88.9	60.3	55.2	88.1	87.7	72.6
CO-PAINTER(Ours)	**89.4**	**90.5**	**87.6**	**93.5**	**88.4**	**89.1**	**89.2**

Table 9. User study results of fine-grained image stylization.

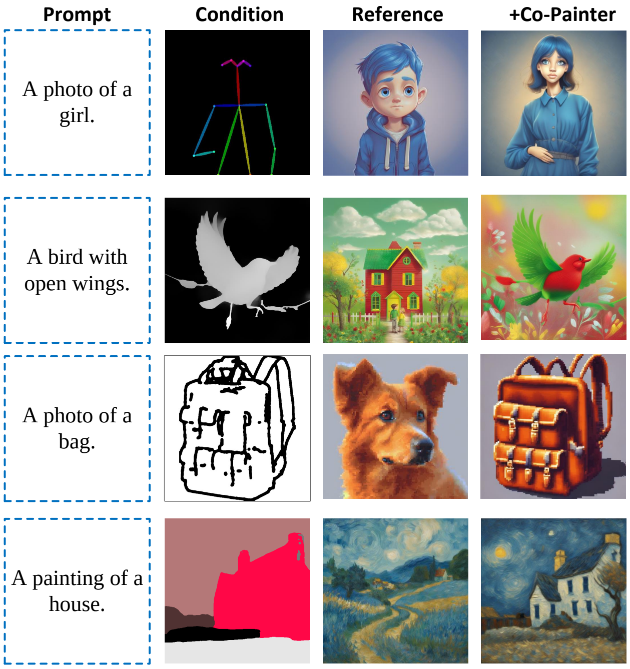


Figure 10. Visualization results of the combination with ControlNet under the standard image stylization setting. CO-PAINTER

technologies, some artists and designers may come to rely on such tools, potentially diminishing their creative abilities and the uniqueness of their artistic expression. Second, **CO-PAINTER** could be misused to generate content that infringes on copyrights or parodies others' works, leading to copyright disputes or misleading audiences. Lastly, the trend toward depersonalization and automation in artistic creation may devalue traditional art forms and handcrafted works, thereby altering the art market and cultural landscape.

Mitigation of Negative Impacts (Security Statement). To minimize potential negative impacts, we will strictly adhere to ethical and legal standards, ensuring that users employ **CO-PAINTER** solely for lawful and legitimate creative activities. Unless authorized, **CO-PAINTER** will only be permitted for use in research domains. Furthermore, access to the proposed dataset will be strictly limited to qualified institutions and organizations, which must provide a clear purpose for its use. We explicitly prohibit the use of the dataset in situations that could lead to potential risks or gen-

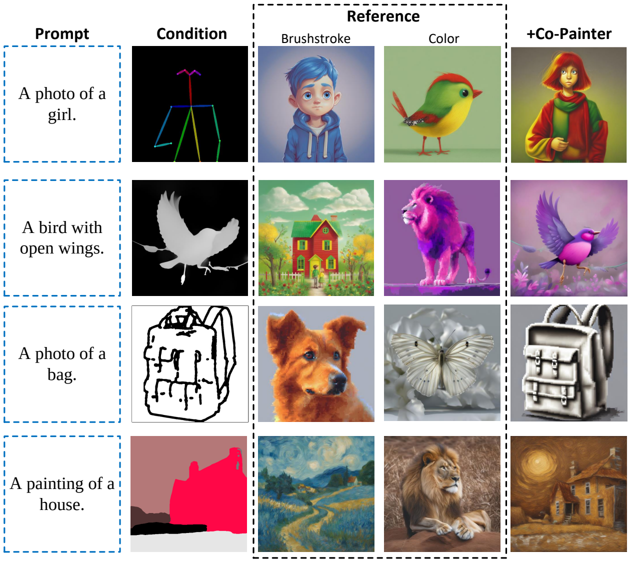


Figure 11. Visualization results of the combination with ControlNet under the fine-grained image stylization setting.

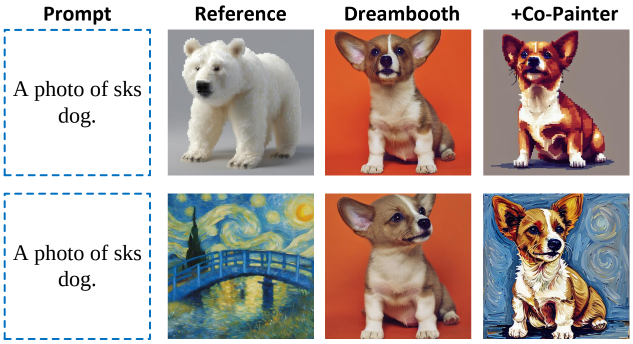


Figure 12. Visualization results of the combination with Dreambooth.

erate significant societal consequences.

17. Limitations and Future Work

17.1. Limitations

This paper provides a detailed analysis of fine-grained controllable image stylization tasks and proposes a superior model, **Co-Painter**. However, the model does not decouple more style attributes from the reference image, such as lighting, and texture. Exploring the effective transfer of a broader range of style attributes is a crucial aspect of fine-grained image stylization tasks. Additionally, as the number of control conditions increases, the model may encounter challenges in handling the more complex problem of multi-style attribute transfer. Ultimately, although our dataset includes a variety of brushstrokes and colors, it is still challenging for the model to fully adapt to the countless styles present in real-world scenarios. This represents a drawback of our study. To address this limitation, we propose the creation of a larger and more diverse fine-grained style dataset.

17.2. Future Work

Considering these limitations, future work could focus on the following studies: First, we can further optimize the model to achieve decoupling and transfer of a broader range of style attributes, while also enhancing its generalization capability and robustness across diverse application scenarios. Second, exploring the integration of multi-modal learning methods could allow for a closer alignment between text prompt and image style attributes, enabling more flexible and precise style control. Finally, constructing a larger and more diverse fine-grained style dataset would enhance the generalization ability of image stylization models and enable zero-shot transfer of fine-grained style attributes.

With these improvements, the model is expected to demonstrate greater adaptability in complex scenarios and meet a wider range of application needs.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 4, 7

[2] 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. 2, 3, 11

[3] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4296–4304, 2024. 2, 3, 11, 13

[4] Tianhao Qi, Shancheng Fang, Yanze Wu, Hongtao Xie, Jiawei Liu, Lang Chen, Qian He, and Yongdong Zhang. Deadiff: An efficient stylization diffusion model with disentangled representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8693–8702, 2024. 2, 3, 5, 11, 13

[5] 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. 1, 3, 4, 7

[6] 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. 1, 2

[7] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 9, 10

[8] Alex Wang. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018. 2

[9] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 1, 2, 3, 11, 13

[10] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2, 3, 4, 7, 9, 10, 13

[11] Yuxin Zhang, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, Tong-Yee Lee, and Changsheng Xu. Domain enhanced arbitrary image style transfer via contrastive learning. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–8, 2022. 2, 3, 11

[12] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023. 2, 3, 11