

# StyleBlend: Enhancing Style-Specific Content Creation in Text-to-Image Diffusion Models

Zichong Chen Shijin Wang Yang Zhou<sup>†</sup>

Visual Computing Research Center, CSSE, Shenzhen University



**Figure 1: Style-specific text-to-image generation.** Given a limited set of style images arranged vertically on the left side, our method generates content images that align with the text semantics while also exhibiting satisfactory stylistic effects (left). Additionally, our approach can be applied to the Stable Diffusion series without any degeneracy or incompatibility (right).

## Abstract

Synthesizing visually impressive images that seamlessly align both text prompts and specific artistic styles remains a significant challenge in Text-to-Image (T2I) diffusion models. This paper introduces StyleBlend, a method designed to learn and apply style representations from a limited set of reference images, enabling content synthesis of both text-aligned and stylistically coherent. Our approach uniquely decomposes style into two components, composition and texture, each learned through different strategies. We then leverage two synthesis branches, each focusing on a corresponding style component, to facilitate effective style blending through shared features without affecting content generation. StyleBlend addresses the common issues of text misalignment and weak style representation that previous methods have struggled with. Extensive qualitative and quantitative comparisons demonstrate the superiority of our approach. Project page: <https://zichongc.github.io/StyleBlend>.

## CCS Concepts

- Computing methodologies → Image processing; Image representations;

## 1. Introduction

In recent years, Text-to-Image (T2I) diffusion models have made remarkable progress in generating high-quality images based on

textual prompts. Models like DALL·E [RPG\*21] and Stable Diffusion [EKB\*24; PEL\*24; RBL\*22] have shown immense potential in various fields, including artistic creation. However, one of the remaining challenges is synthesizing images that seamlessly integrate textual prompts with specific artistic styles derived from a set

<sup>†</sup> Corresponding author.

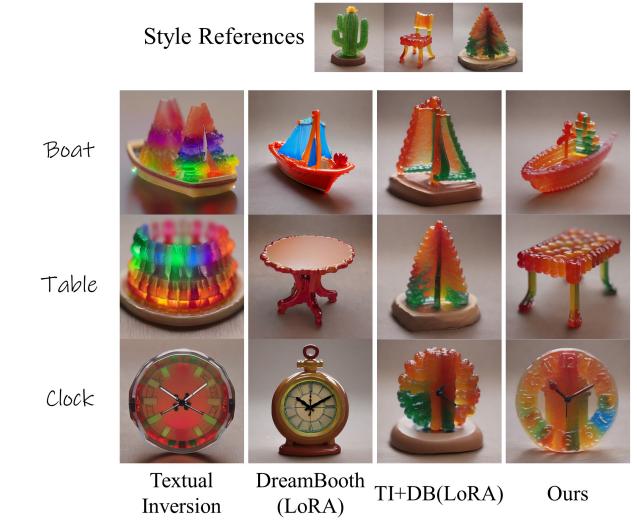
of images. This integration is crucial for applications that demand a high level of style alignment, such as personalized art creation.

Current methods [GAA\*22; KZZ\*23; RLJ\*23; RLJ\*24; ARMC23; ZDT\*23] typically finetune a pretrained T2I model over a set of images that share the same style, but often fall short in two crucial aspects: text alignment and style representation. Most of them struggle to balance style and content in the generated images, especially when there is very limited reference data. As shown in Fig. 2, Textual Inversion (TI) [GAA\*22] fails to reproduce the fine details of the given style and meanwhile suffers from poor semantic content alignment. DreamBooth (DB) retains semantic integrity much better but doesn't capture the style info precisely. Surprisingly, combining TI and DB yields notable stylistic effects in mimicking the references' appearance and texture, but overfitting issues occur; see the 3rd column of Fig. 2 for example. We attribute this to the strong correlation between appearance and semantics, particularly under the challenge of data scarcity (limited data and lack of diversity). Overfitting enhances style representation but meanwhile sacrifices semantic alignment. This gap highlights the need for a method that can robustly blend text-aligned content with stylistic coherence, especially in few-shot cases.

To address these challenges, we propose StyleBlend, a method that decomposes the image style into two components: composition and texture. Composition refers to the semantic structure and layout information of a scene, focusing on the arrangement and relationships between objects and elements without sacrificing semantic integrity. In contrast, texture emphasizes finer details and local appearance. To generate style-specific images, StyleBlend employs a dual-branch framework, where each branch corresponds to one style component. To learn the style components separately, we start from our insights on the two simple yet widely-used techniques [GAA\*22; RLJ\*23]. As discussed above, narrowing the gap between the generated domain and the target domain can significantly enhance the capture of style information. We thus learn the texture style of images using this approach. Then, we introduce a newly designed strategy upon SDEdit [MHS\*22] technique to capture the composition style, which can provide accurate semantic content with correct layout information in inference. Based on the two learned style representations, we blend them through mutual feature injection between the branches during synthesis. Qualitative and quantitative results show that, compared with existing methods, StyleBlend overcomes the common issues of text misalignment and weak style representation, offering a more robust solution for style-specific content generation in T2I diffusion models.

Our contributions are summarized as follows:

- We introduce a new method for decomposing and learning style representations, enabling the synthesis of images that are both text-aligned and stylistically coherent.
- We develop a dual-branch synthesis framework with style features blending, achieving high-quality stylized image generation.
- We demonstrate the superiority of our approach through extensive qualitative and quantitative comparisons with existing state-of-the-art methods.



**Figure 2: Motivation of StyleBlend.** Given the three reference images at the top, we leverage existing methods to learn style representations and generate results conditioned by the class-specific prompts. The widely-used techniques, such as Textual Inversion (TI) [GAA\*22] and DreamBooth (DB) [RLJ\*23], struggle to produce high-quality style-specific images that maintain style coherence and text alignment simultaneously. We also experimented with combining these two approaches. While yielding surprising stylistic effects, it fell into overfitting. Our approach builds upon these observations and achieves much higher quality. Note that we only train additional LoRAs instead of the entire denoising network in DreamBooth for efficiency.

## 2. Related work

### 2.1. T2I diffusion models and personalization

Recent advancements in T2I diffusion models, particularly within the Stable Diffusion series [EKB\*24; PEL\*24; RBL\*22], have greatly improved the quality and diversity of images generated from text prompts. Building on this progress, a new task has emerged, called personalization, which tries to learn and generate new images of a specific subject based on a few reference images while preserving the key characteristics of that subject. Some approaches [ARMC23; GAA\*22; VCCA24] introduce new tokens paired with corresponding textual embeddings to capture the subject. These extended embeddings are optimized by traditional diffusion loss and used to prompt the model in inference to generate desired results. Some other methods [HHZW24; HLD\*23; KZZ\*23; MSM\*23; PYZ\*24; RLJ\*23; RLJ\*24; SRC\*23] focus on refining layer weights or integrating additional features to enhance the model's generative capabilities for personalized characteristics. Recently, inspired by inversion encoders in GAN [BSP\*19; RAP\*21; RMBC22], image adapters have also been introduced for concept learning in T2I personalization [GAA\*23; YZL\*23]. However, the subject or concept of an image is usually more abstract and unspecific than style. As a result, these methods often struggle to maintain consistent text alignment and fully capture fine details of styles, particularly when data is limited.



**Figure 3: Style representation and learning in StyleBlend.** We decompose the image style into two components: texture style and composition style. Texture style refers to the texture and appearance of images. We first optimize a global embedding with an identifier “ $<T>$ ” to learn the subject of the given images, which can narrow the generated domain, followed by training the LoRA weights for finer-detailed style learning (left). For composition style, we generate a set of training data using SDEdit [MHS\*22] and then train the LoRA weights of the text encoder with an identifier “ $<C>$ ” to capture the semantic structure and layout of images (right).

## 2.2. Style-specific image generation

Starting with the traditional style transfer [GEB16], which applies the artistic style of one image to the content of another, recent works [CHH24; HLZ\*24; WZX23; ZHT\*23; DHTD24] have advanced by incorporating diffusion models. Style-specific text-to-image generation, a notable subfield of personalization, aims to produce images that not only align with the provided text prompts but also embody specific artistic styles. Recent research in this area can be categorized into three main types: fine-tuning, training-free, and pretrained methods. Fine-tuning methods [ALL\*24; EBA\*23; FVSC24; GAA\*22; JWK\*24; KZZ\*23; RLJ\*23; SJB\*23; FVSC24], similar to the personalization approaches discussed above, learn the target style by fine-tuning T2I models on a given set of reference images that share the same style. Training-free methods, such as StyleAligned [HVFC24], RB-Modulation [RCR\*24], Visual Style Prompting [JKC\*24] and InstantStyle [WWB\*24], leverage the prior knowledge learned by pretrained T2I models to integrate styles without the need for additional training. Pretrained methods [CPYM23; JYY\*24; WWX\*24; XWS\*24; YZL\*23] are designed to handle arbitrary styles based on large-scale datasets, offering a versatile solution for style-specific image generation. However, existing fine-tuning methods still encounter the same issues as personalization approaches, while the training-free and pretrained methods struggle with capturing the precise style information defined by a given set of style images. Instead, we developed a new fine-tuning approach that can vividly reproduce the style (both global and local) while mitigating text misalignment issues found in existing personalization methods.

## 3. Preliminaries

### 3.1. Diffusion models and personalization

Diffusion models [SWMG15; HJA20] simulate a diffusion process where images are progressively perturbed with noise and then reconstructed through a learned denoising mechanism. Our approach builds upon the prominent Stable Diffusion models [PEL\*24; RBL\*22], which performs the diffusion process in latent space and

effectively utilize a U-Net architecture comprising multiple convolution and transformer [VSP\*17] blocks as the diffuser.

Specifically, during the training of Stable Diffusion, an image  $x$  is first compressed into latent space using a pretrained encoder  $\mathcal{E}$ . Then, noise is added to latent code  $z$ , and a denoising U-Net  $\epsilon_\theta(\cdot)$  is trained to predict the added noise with the following objective:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{z \sim \mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0, I), t} [\|\epsilon - \epsilon_\theta(z_t, t, c(y))\|_2^2], \quad (1)$$

where  $t$  denotes the timestep, and  $y$  represents the textual condition, which is typically encoded into text embeddings using a CLIP encoder  $c(\cdot)$  [RKH\*21].

There are two widely used techniques to personalize pretrained T2I models based on a set of images that share the same concept, Textual Inversion [GAA\*22] and DreamBooth [RLJ\*23]. Textual Inversion optimizes a new token with a corresponding embedding vector  $e^*$  using the same objective as Eq. 1. DreamBooth modifies the parameters throughout the entire denoising network, also with the same objective. Additionally, we can introduce LoRAs [HWA\*21] into specific layers of pre-trained diffusion networks. A finetuned LoRA works the same as DreamBooth but is more memory efficient. See the first two columns of Fig. 2 for examples of style-specific T2I generation using these two approaches.

### 3.2. Feature injection in T2I diffusion models

The features in the convolutional layers and attention blocks of the denoising U-Net have demonstrated their effectiveness in information encoding [TGBD23; CWQ\*23; HVFC24; AGP\*24]. In this work, we focus on the U-Net self-attention features, which are calculated according to the self-attention mechanism:

$$\text{Self-Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V, \quad (2)$$

where  $d$  is the feature dimension,  $Q$ ,  $K$ , and  $V \in \mathbb{R}^{s \times d}$  represent the Query, Key, and Value, respectively. The Query here has been empirically shown to effectively capture semantic structures, while

the Key and Value components excel at representing texture and appearance. Feature injection typically refers to the copying or concatenation of QKV features across synthesis branches within the same layers.

#### 4. StyleBlend

Given a limited set of images with category-specific text prompts that share the same style, our goal is to synthesize new images using T2I diffusion models that align the image semantics with user-provided text prompts while maintaining style coherence with the given style references. In this section, we introduce the proposed StyleBlend method. We first present the representation and learning of image styles in Sec. 4.1, followed by an illustration of the inference process using our dual-branch synthesis framework with feature blending in Sec. 4.2.

##### 4.1. Style representation and learning

$$\text{Image Style} = \text{Composition Style} + \text{Texture Style}$$

The style of an image encompasses various elements arranged hierarchically. Based on empirical observations, we define the formulation above, decomposing image style into two key components: composition and texture. Composition style captures the semantic structure and layout of an image, while texture style focuses on the finer details of texture and appearance. To learn these two style representations, we employ distinct training strategies tailored to each component, as depicted in Fig. 3.

**Texture style representation.** The key to our task is to learn a robust style representation, where existing methods, as shown in Fig. 2, often fall short. Our key insight is that narrowing the gap between the generated style domain and the target domain at the outset can improve style transfer. For example, we can manually write a detailed style description, such as “gelatin candy toy”, to serve as an identifier for the target style. However, this approach often struggles to precisely match the desired style, and obtaining such a description can be very challenging for users.

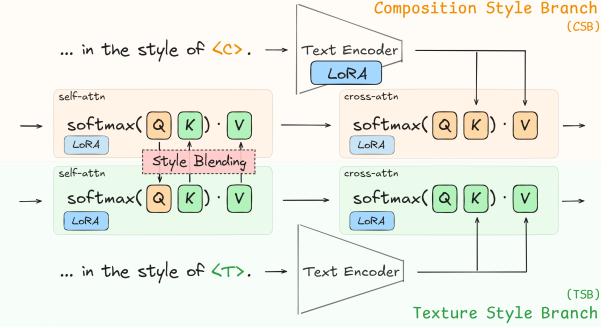
We, therefore, adopt Textual Inversion [GAA\*22] to optimize a global embedding to narrow the domain gap. As illustrated on the left side of Fig. 3, we use a new token “<T>” with a corresponding embeddings vector  $e_T$  as a global style identifier, and optimize it with the following objective:

$$e_T = \arg \min_e \mathbb{E}_{z,y,e,t} [\|\epsilon - \epsilon_\theta(z_t, t, c(y, e))\|_2^2]. \quad (3)$$

Afterward, we introduce LoRAs into the denoising network, which is trained similarly to Dreambooth [RLJ\*23] to capture the finer details of style. We optimize the LoRA weights  $\theta_{LoRA}^T$  using the same reference images with the same objective, defined as:

$$\theta_{LoRA}^T = \arg \min_{\theta_{LoRA}} \mathbb{E}_{z,y,e_T,t} [\|\epsilon - \epsilon_{\theta, \theta_{LoRA}}(z_t, t, c(y, e_T))\|_2^2]. \quad (4)$$

However, directly synthesizing results with this approach is prone to overfitting, as shown in the third column of Fig. 2, yet it produces strong stylistic effects. Therefore, we design to extract the learned texture style here and further develop a separate process to capture the composition style, ensuring the preservation of semantic structure and layout.



**Figure 4: Dual-branch synthesis framework for inference.** The framework comprises two branches: a composition style branch (CSB) and a texture style branch (TSB), which produce composition style and texture style, respectively. We employ mutual feature injection to blend these style components. In the injected features, the  $Q$  features from CSB represent the composition style, while the  $KV$  features from TSB describe the texture style.

**Composition style representation.** The composition style emphasizes semantic structure and layout rather than texture. To effectively capture the compositional information, we start by generating a set of images using an image editing tool [MHS\*22], as depicted in the middle of Fig. 3. We corrupt a reference image by adding noise to a certain extent and then denoise the noisy image back to a clear state with simple text prompts from its class name, e.g., “A bicycle.” for an image containing a bicycle. Such generated images share highly similar structures and layouts with reference images while introducing variations in texture style.

We learn the composition style solely from the generated dataset. Unlike the texture style representation, here we add LoRAs to the text encoder, as illustrated on the right side of Fig. 3. Additionally, We introduce a new token “<C>” with the corresponding embedding vector  $e_C$  to identify the composition style and train the LoRA weights  $\theta_{LoRA}^C$  with the following objective:

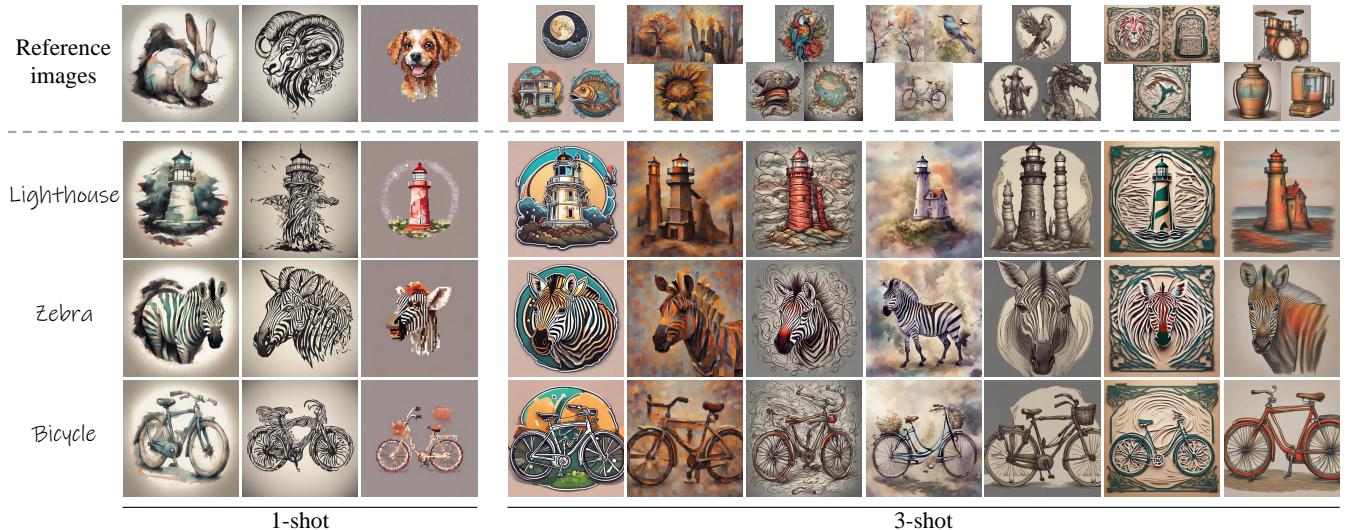
$$\theta_{LoRA}^C = \arg \min_{\theta_{LoRA}} \mathbb{E}_{z,y,e_C,t} [\|\epsilon - \epsilon_\theta(z_t, t, c_{\theta_{LoRA}}(y, e_C))\|_2^2]. \quad (5)$$

Note that we empirically found that directly training the LoRAs without optimizing  $e_C$  can effectively prevent overfitting and, in advance, yield better performance. Therefore, we only optimize the LoRA weights of the text encoder to capture the composition style.

##### 4.2. Features blending across style components

After style learning, we obtain two sets of parameters  $(\theta_{LoRA}^C, e_C)$  and  $(\theta_{LoRA}^T, e_T)$ : one for synthesizing results with composition style and the other for texture style. To generate results that match the desired style effects, we need to effectively “add” these two distinct styles following our formulation outlined above.

As illustrated in Fig. 4, we construct a dual-branch synthesis framework to separately generate composition and texture styles. Inspired by recent works [TGBD23; CWQ\*23; HVFC24] on feature manipulations in T2I models, we fetch the QKV features from



**Figure 5:** Gallery of StyleBlend results for 1-shot (left) and 3-shot (right) cases, where style references are shown at the top.

the self-attention blocks and blend these features across the two synthesis branches for effective style integration. Empirical findings suggest that Q features capture semantic structure and layout while K and V represent texture and appearance. Accordingly, we swap QKV features between the two branches: the Q features from the texture style branch (TSB) are replaced with those from the composition style branch (CSB), while the KV features from CSB are replaced with those from TSB. This ensures proper blending of composition and texture in the final output, which is taken from the texture style branch. We refer to this approach as StyleBlend.

## 5. Experiments

### 5.1. Implementation details

We have implemented our method over the Stable Diffusion series (v1.5, v2.1, SDXL) [RBL\*22; PEL\*24] on a single RTX 4090. The learning processes for composition and texture styles are independent, with no specific priority assigned to their training order. We initialize  $e_T$  with the token embedding of “artistic” and  $e_C$  with a rare word such as the exclamation mark. During training, we apply early stopping after a few hundred steps, which is enough to get satisfactory results. In practice, we typically optimize 500 steps for style embeddings and LoRA weights during texture style learning, while only 300 steps are required to optimize the LoRA weights of the text encoder during composition style learning. We set the learning rate to 0.0001 for optimizing all LoRA weights, using a larger learning rate of 0.01 for the style embeddings. The rank of LoRAs is set to 16 across all our experiments. To generate a plausible dataset for composition style learning, we synthesize 10 images for each reference image. We apply noise perturbation to the reference images at 0.8T for Stable Diffusion v1.5 and v2.1, and 0.9T for SDXL, serving as the initial code for SDEdit sampling. The sampling process involves 20 linear steps.

For the computation time, training on Stable Diffusion 2.1 at a resolution of 768x768 takes approximately 75 seconds for dataset

creation, 40 seconds for composition style learning, and 2 minutes for texture style learning. Inference requires around 4 seconds per image. Unless specified, the results presented are generated using Stable Diffusion v2.1 as the foundation model.

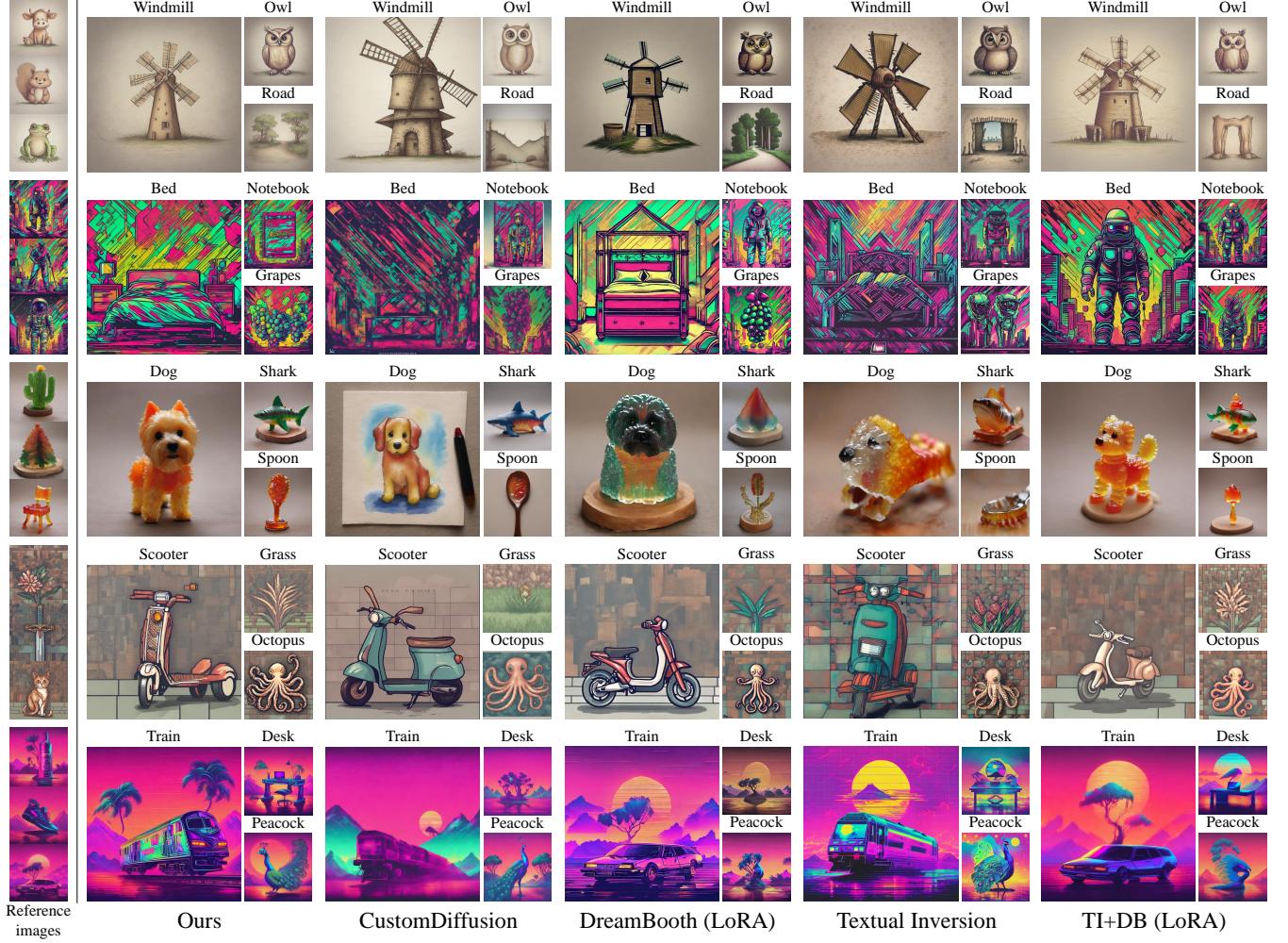
**Dataset.** To test and evaluate our approach, we collected several style images from the Internet, and synthesized additional sets of images that share the same style using StyleAligned [HVFC24] and VSP [JKC\*24] over SDXL [PEL\*24]. We focus on 3-shot and 1-shot cases for fair comparison. We format the text prompts with the same templates: “A ... in the style of <C>” for composition style and “A ... in the style of <T>” for texture style, where the ellipses are filled with class-specific prompts, *i.e.*, the class names.

**Metrics.** We utilize the Contrastive Style Descriptor (CSD) [SGG\*24] to access style similarity to the target style, and the CLIP-Score [RKH\*21] to evaluate text alignment with the input text prompts. Specifically, we measure the style similarity by computing the cosine similarity of style embeddings extracted by CSD between the generated images and the target style image. For the style set with multiple reference images, we first calculate the score between the generated image and each reference image and then take the maximum one as the final score for each generated image. The CLIP-Score is computed by measuring the cosine similarity between the CLIP embeddings of generated images and the corresponding input text prompts.

### 5.2. Results and comparisons

**Style-specific T2I synthesis.** Fig. 1 and Fig. 5 showcase the results of StyleBlend applied to various style examples. We focus on few-shot cases involving three reference images (3-shot) or a single image (1-shot). The results demonstrate that our method effectively manages both style coherence and text alignment simultaneously.

**Comparison.** To validate the effectiveness of our approach, we conducted qualitative and quantitative comparisons to several base-



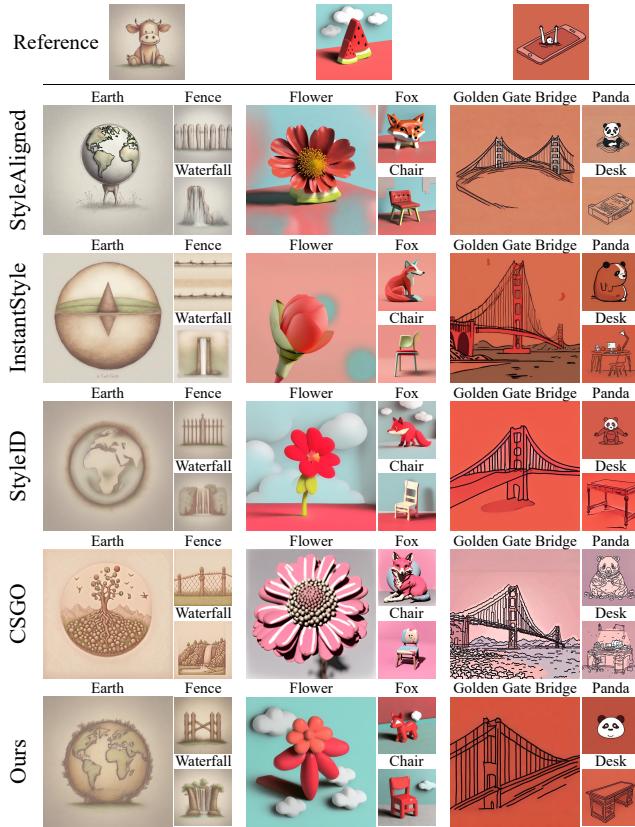
**Figure 6:** Qualitative comparison to finetuning-based methods for style-specific T2I generation in 3-shot cases.

lines. These include finetuning-based methods such as DreamBooth [RLJ\*23], Textual Inversion [GAA\*22], and CustomDiffusion [KZZ\*23], as well as a pretrained model CSGO [XWS\*24] and three training-free methods: StyleAligned [HVFC24], InstantStyle [WWB\*24], and StyleID [CHH24]. For the baselines, we trained DreamBooth and Textual Inversion for 1000 steps using a learning rate of 0.0005 and 0.05, respectively. For other methods, we adhered to the configurations released with the open-source repositories. To run StyleAligned, we use GPT to generate a detailed style description for the DDIM inversion of the reference image and style generation. Furthermore, since StyleID requires a content image as input, we first generate content images using our CSB branch to provide a well-structured layout, and then apply StyleID to transfer the reference style to the generated images. We also compared our approach with a variant that combines Textual Inversion and DreamBooth. Note that we optimize a set of LoRAs for DreamBooth instead of the entire denoising network.

Figs. 6 and 7 present the qualitative comparisons. Specifically, we compare our approach to finetuning-based methods in the 3-

shot case in Fig. 6. We can see that recent advances suffer from weak style representation: the generated styles are not coherent with the target styles. Some of these methods even exhibit text misalignment, as seen in the 2nd and 5th rows, where the image content does not correspond accurately to the given text prompts. In contrast, our method effectively addresses these shortcomings, surpassing the baselines both in style and text alignment, leading to high-quality results. In Fig. 7, we compare our method with the pretrained model and training-free methods in the 1-shot case, given that these methods accept only a single style image as input. Although the baselines perform well in terms of text alignment, they also struggle with style representation, much like the finetuning-based methods. Moreover, we found StyleAligned produced some destroyed structures, which we infer was caused by content leakage. For StyleID, the generated content images greatly affect the final results, resulting in less stylization than ours.

Fig. 8 further presents quantitative comparisons on the metrics mentioned above, i.e., the CSD-based style similarity and the CLIP-Score for text alignment. All the scores are computed across



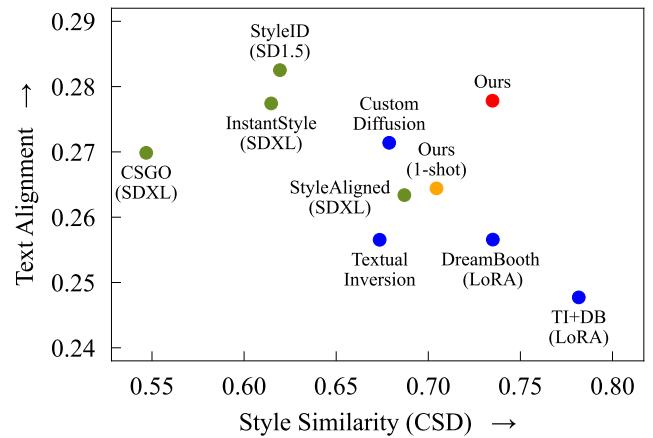
**Figure 7:** Qualitative comparison to existing 1-shot methods in style-specific T2I generation, where CSGO [XWS\*24] is a model pretrained on large style dataset, and the other three (StyleAligned [HVFC24], InstantStyle [WWB\*24], and StyleID [CHH24]) are training-free methods.

the five styles presented in Fig. 6. From the scores, our method achieves higher values for both style coherence and semantics alignment in the 3-shot case. Although our 1-shot results exhibit a slight weakness in text alignment due to the extremely limited data, they still produce satisfactory style effects compared to many other methods. In contrast, most baselines struggle to balance style coherence and text alignment.

### 5.3. Ablation study

We report the ablation results of StyleBlend on two aspects: style representation and feature injection.

**Style representation.** As shown in Fig. 9, we separately synthesize results using only the composition style branch (CSB) and the texture style branch (TSB) (the first two rows). The results demonstrate that the composition style effectively captures the semantic structure and layout of images without compromising semantic integrity, while the texture style excels in representing texture and appearance but loses nearly all semantic information. Quantitative ablation results in Fig. 11 clearly illustrate the distinct performances of the composition and texture styles. Given that the TSB results



**Figure 8:** Quantitative evaluation. The finetuning-based methods trained with three reference images (3-shot cases) are marked as blue scatter points, the pretrained and training-free approaches that infer from a single image (1-shot cases) are shown in green. Our results for the 3-shot and 1-shot cases are indicated by red and orange scatter points, respectively.



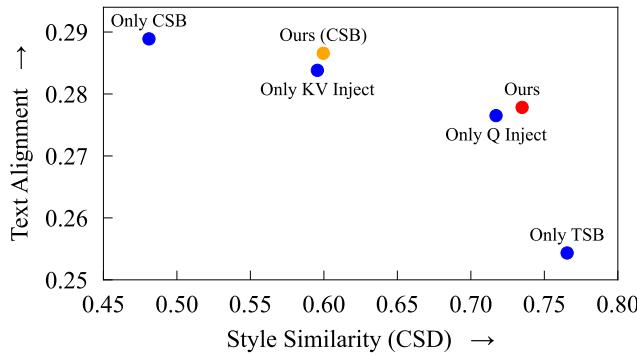
**Figure 9:** Ablation of the style representations.

significantly surpass CSB in style coherence, we choose the outputs from TSB as our final results. We also present the results of StyleBlend without learning the composition style from SDEdit data to highlight the importance of semantic structure and layout in synthesis (the 3rd row). When the learned composition style is removed—by excluding the learned LoRAs from the text encoder—the generated layout becomes disorganized, which further emphasizes the importance of composition style in capturing structure and layout information and preserving semantic coherence.

**Feature injection.** In the dual-branch synthesis framework, we extract the Q and KV features from the self-attention layers in the denoising U-Net to represent the composition and texture style features, respectively. Figs. 10 and 11 show the qualitative and quantitative ablation study, respectively, on feature injection. When only KV features from the texture style branch (TSB) are injected into



**Figure 10:** Ablation of the feature injection.

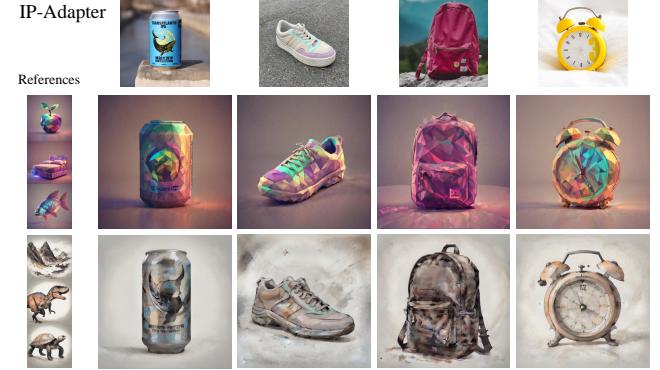
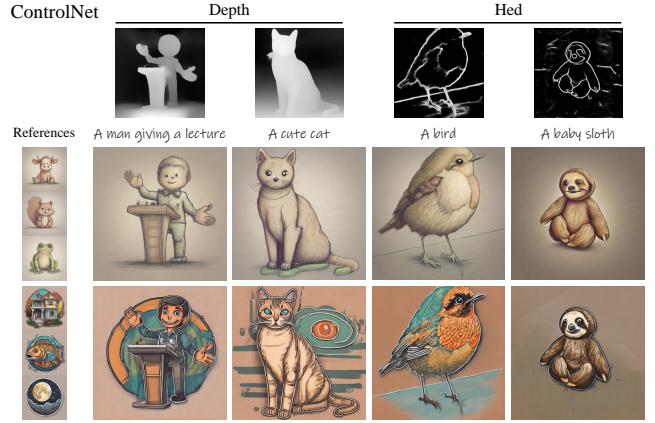


**Figure 11:** Quantitative results of ablation study. Note Ours (CSB) refers to the results produced by the composition style branch (CSB) of our full model, which is significantly inferior in style to the TSB results (Ours in the figure). We thus choose the outputs of TSB branch as our final results.

the composition style branch (CSB), the style of the results is apparently getting weak. On the other hand, injecting only Q features from the composition branch into the texture branch introduces some artifacts in the details, which indicates that running the two branches separately might cause incompatibility between the features of the two branches. In contrast, our full method, which leverages both branches and uses mutual feature injection, produces higher-quality results from the texture style branch.

#### 5.4. Additional results and applications

**StyleBlend with ControlNet and IP-Adapter.** Our method builds on pretrained SD models without modifying the original architecture or parameters. As a result, it can be easily integrated with many other pretrained plug-in methods, leveraging the extensive capabilities of the Stable Diffusion community. Fig. 12 presents the results of StyleBlend combined with ControlNet [ZRA23] and IP-Adapter [YZL\*23]. When used along with ControlNet, we employ depth maps and HED [XT15] images as the conditions, synthesizing visual contents prompted by the texts shown beneath the conditional images. As we can see, this combination produces quite



**Figure 12:** Applying StyleBlend with ControlNet [ZRA23] and IP-Adapter [YZL\*23].

acceptable outputs. When combined with IP-Adapter, we input a set of images featuring different objects. The results show that this combination not only generates the target IPs but also effectively characterizes the key components of the specified style.

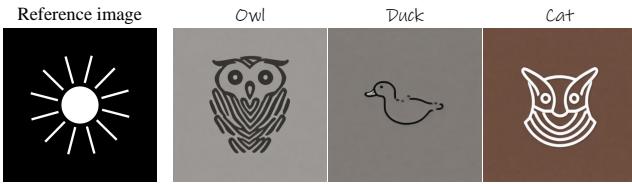
**StyleBlend on more T2I diffusion models.** Since the Stable Diffusion series [RBL\*22; PEL\*24] share a similar architecture, we can easily apply StyleBlend across different base models. For SDXL, which includes two text encoders, we train separate LoRAs on each to learn the composition style. Given SDXL’s larger number of parameters, we increase the optimization steps to 400 for composition style learning and 600 for texture style learning. The configurations for SD v1.5 and v2.1 follow the same setup as described in Sec. 5.1. Fig. 13 shows the creative results of StyleBlend on three Stable Diffusion models. By providing identical prompts and using the pretrained weights of StyleBlend for a specific style, our method can generate creative visual content that is both correct in semantics and consistent and coherent in style.

#### 5.5. Limitations and discussions

While we have demonstrated the effectiveness of StyleBlend across various style examples, several limitations exist. First, while our method performs well in 3-shot cases, it occasionally struggles with text-semantics alignment when using only one reference image, as



**Figure 13:** Creative StyleBlend results obtained with various Stable Diffusion base models.



**Figure 14:** Limitation of StyleBlend.

reflected by the text alignment score in Fig. 8. Besides, for extreme cases shown in Fig. 14, although our method captures the line style from the “sun” icon, it fails to replicate the black-and-white color accurately. Moreover, our current style-blending approach, which utilizes two branches for synthesis, is not the most efficient, as it doubles the inference cost. Future work may focus on developing a more efficient way to blend the two learned style components.

## 6. Conclusions

We have presented StyleBlend, an approach designed to enhance style-specific text-to-image generation based on the Stable Diffusion series. By employing two distinct training strategies to capture both the composition and texture styles of images using a limited set of reference images, StyleBlend effectively prepares for the blending of these styles in the synthesis process. The dual-branch synthesis framework integrates composition and texture styles to generate final outputs, achieving style coherence and text alignment in the visual contents. Last but not least, StyleBlend is efficient in terms of training time, the number of reference images required, and the number of learnable parameters.

## Acknowledgments

This work was supported in parts by NSFC (U21B2023), Guangdong Basic and Applied Basic Research Foundation (2023B1515120026), DEGP Innovation Team (2022KCXTD025), and Scientific Development Funds from Shenzhen University.

## References

- [AGP\*24] ALALUF, YUVAL, GARIBI, DANIEL, PATASHNIK, OR, et al. “Cross-image attention for zero-shot appearance transfer”. *Proc. of SIGGRAPH*. 2024, 1–12 3.
- [ALL\*24] AHN, NAMHYUK, LEE, JUNSOO, LEE, CHUNGGI, et al. “DreamStyler: Paint by Style Inversion with Text-to-Image Diffusion Models”. *Proc. AAAI Conf. on Artificial Intelligence*. 2024, 674–681 3.
- [ARMC23] ALALUF, YUVAL, RICHARDSON, ELAD, METZER, GAL, and COHEN-OR, DANIEL. “A Neural Space-Time Representation for Text-to-Image Personalization”. *ACM Trans. on Graphics* 42.6 (2023) 2.
- [BSP\*19] BAU, DAVID, STROBELT, HENDRIK, PEEBLES, WILLIAM, et al. “Semantic photo manipulation with a generative image prior”. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 38.4 (2019) 2.
- [CHH24] CHUNG, JIWOO, HYUN, SANGEEK, and HEO, JAE-PIL. “Style Injection in Diffusion: A Training-free Approach for Adapting Large-scale Diffusion Models for Style Transfer”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2024, 8795–8805 3, 6, 7.
- [CPYM23] CHEN, JINGWEN, PAN, YINGWEI, YAO, TING, and MEI, TAO. “ControlStyle: Text-Driven Stylized Image Generation Using Diffusion Priors”. *ACM International Conference on Multimedia*. Ottawa ON, Canada, 2023, 7540–7548 3.
- [CWQ\*23] CAO, MINGDENG, WANG, XINTAO, QI, ZHONGANG, et al. “MasaCtrl: Tuning-Free Mutual Self-Attention Control for Consistent Image Synthesis and Editing”. *Proc. of Int. Conf. on Computer Vision*. 2023, 22560–22570 3, 4.
- [DHTD24] DENG, YINGYING, HE, XIANGYU, TANG, FAN, and DONG, WEIMING. “Z\*: Zero-shot Style Transfer via Attention Reweighting”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2024, 6934–6944 3.
- [EBA\*23] EVERAERT, MARTIN NICOLAS, BOCCHIO, MARCO, ARPA, SAMI, et al. “Diffusion in Style”. *Proc. of Int. Conf. on Computer Vision*. 2023, 2251–2261 3.
- [EKB\*24] ESSER, PATRICK, KULAL, SUMITH, BLATTMANN, ANDREAS, et al. “Scaling Rectified Flow Transformers for High-Resolution Image Synthesis”. *arXiv preprint arxiv:2403.03206* (2024) 1, 2.
- [FVSC24] FRENKEL, YARDEN, VINKER, YAEL, SHAMIR, ARIEL, and COHEN-OR, DANIEL. “Implicit Style-Content Separation using B-LORA”. *Proc. of Euro. Conf. on Computer Vision*. 2024 3.
- [GAA\*22] GAL, RINON, ALALUF, YUVAL, ATZMON, YUVAL, et al. “An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion”. *Proc. of Int. Conf. on Learning Representations*. 2022 2–4, 6.
- [GAA\*23] GAL, RINON, ARAR, MOAB, ATZMON, YUVAL, et al. “Encoder-based Domain Tuning for Fast Personalization of Text-to-Image Models”. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 42.4 (2023) 2.
- [GEB16] GATYS, LEON A., ECKER, ALEXANDER S., and BETHGE, MATTHIAS. “Image Style Transfer Using Convolutional Neural Networks”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2016, 2414–2423 3.
- [HHZW24] HAO, SHAOZHE, HAN, KAI, ZHAO, SHIAO, and WONG, KWAN-YEE K. “ViCo: Plug-and-play Visual Condition for Personalized Text-to-image Generation”. *arXiv preprint arxiv:2306.00971* (2024) 2.
- [HJA20] HO, JONATHAN, JAIN, AJAY, and ABBEEL, PIETER. “Denoising Diffusion Probabilistic Models”. *Advances in Neural Information Processing Systems*. 2020 3.
- [HLD\*23] HUA, MIAO, LIU, JIAWEI, DING, FEI, et al. “DreamTuner: Single Image is Enough for Subject-Driven Generation”. *arXiv preprint arxiv:2312.13691* (2023) 2.
- [HLZ\*24] HE, FEIHONG, LI, GANG, ZHANG, MENGYUAN, et al. “Freestyle: Free lunch for text-guided style transfer using diffusion models”. *arXiv preprint arxiv:2401.15636* (2024) 3.

- [HVFC24] HERTZ, AMIR, VOYNOV, ANDREY, FRUCHTER, SHLOMI, and COHEN-OR, DANIEL. “Style Aligned Image Generation via Shared Attention”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2024, 4775–4785 3–7.
- [HWA\*21] HU, EDWARD J., WALLIS, PHILLIP, ALLEN-ZHU, ZEYUAN, et al. “LoRA: Low-Rank Adaptation of Large Language Models”. *Proc. of Int. Conf. on Learning Representations*. 2021 3.
- [JKC\*24] JEONG, JAESOK, KIM, JUNHO, CHOI, YUNJEY, et al. “Visual Style Prompting with Swapping Self-Attention”. *arXiv preprint arXiv:2402.12974* (2024) 3, 5.
- [JWK\*24] JONES, MAXWELL, WANG, SHENG-YU, KUMARI, NUPUR, et al. “Customizing Text-to-Image Models with a Single Image Pair”. *arXiv preprint arXiv:2405.01536* (2024) 3.
- [JYY\*24] JUNYAO, GAO, YANCHEN, LIU, YANAN, SUN, et al. “StyleShot: A Snapshot on Any Style”. *arXiv preprint arXiv:2407.01414* (2024) 3.
- [KZZ\*23] KUMARI, NUPUR, ZHANG, BINGLIANG, ZHANG, RICHARD, et al. “Multi-Concept Customization of Text-to-Image Diffusion”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2023 2, 3, 6.
- [MHS\*22] MENG, CHENLIN, HE, YUTONG, SONG, YANG, et al. “SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations”. *Proc. of Int. Conf. on Learning Representations*. 2022 2–4.
- [MSM\*23] MARJIT, SHYAM, SINGH, HARSHIT, MATHUR, NITYANAND, et al. “DiffuseKronA: A Parameter Efficient Fine-tuning Method for Personalized Diffusion Model”. *arXiv preprint arXiv:2402.17412* (2023) 2.
- [PEL\*24] PODELL, DUSTIN, ENGLISH, ZION, LACEY, KYLE, et al. “SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis”. *Proc. of Int. Conf. on Learning Representations*. 2024 1–3, 5, 8.
- [PYZ\*24] PANG, LIANYU, YIN, JIAN, ZHAO, BAOQUAN, et al. “AttnDreamBooth: Towards Text-Aligned Personalized Text-to-Image Generation”. *arXiv preprint arXiv:2406.05000* (2024) 2.
- [RAP\*21] RICHARDSON, ELAD, ALALUF, YUVAL, PATASHNIK, OR, et al. “Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. June 2021 2.
- [RBL\*22] ROMBACH, ROBIN, BLATTMANN, ANDREAS, LORENZ, DOMINIK, et al. “High-Resolution Image Synthesis With Latent Diffusion Models”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2022, 10684–10695 1–3, 5, 8.
- [RCR\*24] ROUT, L, CHEN, Y, RUIZ, N, et al. “RB-Modulation: Training-Free Personalization of Diffusion Models using Stochastic Optimal Control”. *arXiv preprint arXiv:2405.17401* (2024) 3.
- [RKH\*21] RADFORD, ALEC, KIM, JONG WOOK, HALLACY, CHRIS, et al. “Learning Transferable Visual Models From Natural Language Supervision”. *Proc. of IEEE Int. Conf. on Machine Learning*. Ed. by MEILA, MARINA and ZHANG, TONG. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, 8748–8763 3, 5.
- [RLJ\*23] RUIZ, NATANIEL, LI, YUANZHEN, JAMPANI, VARUN, et al. “DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2023, 22500–22510 2–4, 6.
- [RLJ\*24] RUIZ, NATANIEL, LI, YUANZHEN, JAMPANI, VARUN, et al. “HyperDreamBooth: HyperNetworks for Fast Personalization of Text-to-Image Models”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2024, 6527–6536 2.
- [RMBC22] ROICH, DANIEL, MOKADY, RON, BERMANO, AMIT H., and COHEN-OR, DANIEL. “Pivotal Tuning for Latent-based Editing of Real Images”. *ACM Trans. on Graphics* 42.1 (Aug. 2022) 2.
- [RPG\*21] RAMESH, ADITYA, PAVLOV, MIKHAIL, GOH, GABRIEL, et al. “Zero-Shot Text-to-Image Generation”. *Proc. of IEEE Int. Conf. on Machine Learning*. 2021, 8821–8831 1.
- [SGG\*24] SOMEPALI, GOWTHAMI, GUPTA, ANUBHAV, GUPTA, KAMAL, et al. “Measuring Style Similarity in Diffusion Models”. *arXiv preprint arXiv:2404.01292* (2024) 5.
- [SJB\*23] SOHN, KIHUK, JIANG, LU, BARBER, JARRED, et al. “StyleDrop: Text-to-Image Synthesis of Any Style”. *Advances in Neural Information Processing Systems*. Ed. by OH, A., NAUMANN, T., GLOBERSON, A., et al. Vol. 36. Curran Associates, Inc., 2023, 66860–66889 3.
- [SRC\*23] SHAH, VIRAJ, RUIZ, NATANIEL, COLE, FORRESTER, et al. “ZipLoRA: Any Subject in Any Style by Effectively Merging LoRAs”. *arXiv preprint arXiv:2311.13600* (2023) 2.
- [SMWG15] SOHL-DICKSTEIN, JASCHA, WEISS, ERIC, MAHESWARANATHAN, NIRU, and GANGULI, SURYA. “Deep unsupervised learning using nonequilibrium thermodynamics”. *Proc. of IEEE Int. Conf. on Machine Learning*. PMLR. 2015, 2256–2265 3.
- [TGBD23] TUMANIAN, NAREK, GEYER, MICHAL, BAGON, SHAI, and DEKEL, TALI. “Plug-and-play diffusion features for text-driven image-to-image translation”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2023, 1921–1930 3, 4.
- [VCCA24] VOYNOV, ANDREY, CHU, QINGHAO, COHEN-OR, DANIEL, and ABERMAN, KFIR. “P+: Extended Textual Conditioning in Text-to-Image Generation”. *arXiv preprint arXiv:2303.09522* (2024) 2.
- [VSP\*17] VASWANI, ASHISH, SHAZER, NOAM, PARMAR, NIKI, et al. “Attention is All you Need”. *Advances in Neural Information Processing Systems*. Ed. by GUYON, I., LUXBURG, U. VON, BENGIO, S., et al. Vol. 30. Curran Associates, Inc., 2017 3.
- [WWB\*24] WANG, HAOFAN, WANG, QIXUN, BAI, XU, et al. “InstantStyle: Free Lunch towards Style-Preserving in Text-to-Image Generation”. *arXiv preprint arXiv:2404.02733* (2024) 3, 6, 7.
- [WWX\*24] WANG, ZHOUXIA, WANG, XINTAO, XIE, LIANGBIN, et al. “StyleAdapter: A Unified Stylized Image Generation Model without Test-Time Fine-Tuning”. *arXiv preprint arXiv:2309.01770* (2024) 3.
- [WZX23] WANG, ZHIZHONG, ZHAO, LEI, and XING, WEI. “StyleDiffusion: Controllable Disentangled Style Transfer via Diffusion Models”. *Proc. of Int. Conf. on Computer Vision*. 2023, 7677–7689 3.
- [XT15] XIE, SAINING and TU, ZHUOWEN. “Holistically-Nested Edge Detection”. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015 8.
- [XWS\*24] XING, PENG, WANG, HAOFAN, SUN, YANPENG, et al. “CSGO: Content-Style Composition in Text-to-Image Generation”. *arXiv preprint arXiv:2408.16766* (2024) 3, 6, 7.
- [YZL\*23] YE, HU, ZHANG, JUN, LIU, SIBO, et al. “IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models”. *arXiv preprint arXiv:2308.06721* (2023) 2, 3, 8.
- [ZDT\*23] ZHANG, YUXIN, DONG, WEIMING, TANG, FAN, et al. “ProSpect: Prompt Spectrum for Attribute-Aware Personalization of Diffusion Models”. *ACM Trans. on Graphics* 42.6 (2023), 244:1–244:14 2.
- [ZHT\*23] ZHANG, YUXIN, HUANG, NISHA, TANG, FAN, et al. “Inversion-Based Style Transfer With Diffusion Models”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2023, 10146–10156 3.
- [ZRA23] ZHANG, LVMIN, RAO, ANYI, and AGRAWALA, MANEESH. “Adding Conditional Control to Text-to-Image Diffusion Models”. *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*. 2023, 3836–3847 8.

## Additional Results

In the below figures, we provide a selection of generated results from StyleBlend over 14 style examples, demonstrating its ability to integrate and preserve diverse artistic styles across various prompts.



**Figure 15:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 16:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 17:** Generated results of StyleBlend using the style references displayed on the left.



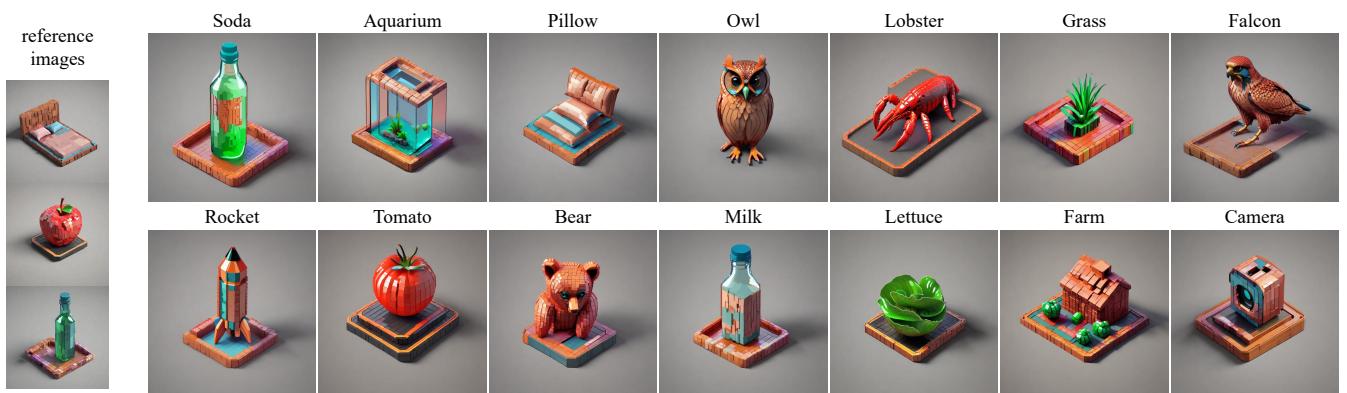
**Figure 18:** Generated results of StyleBlend using the style references displayed on the left.



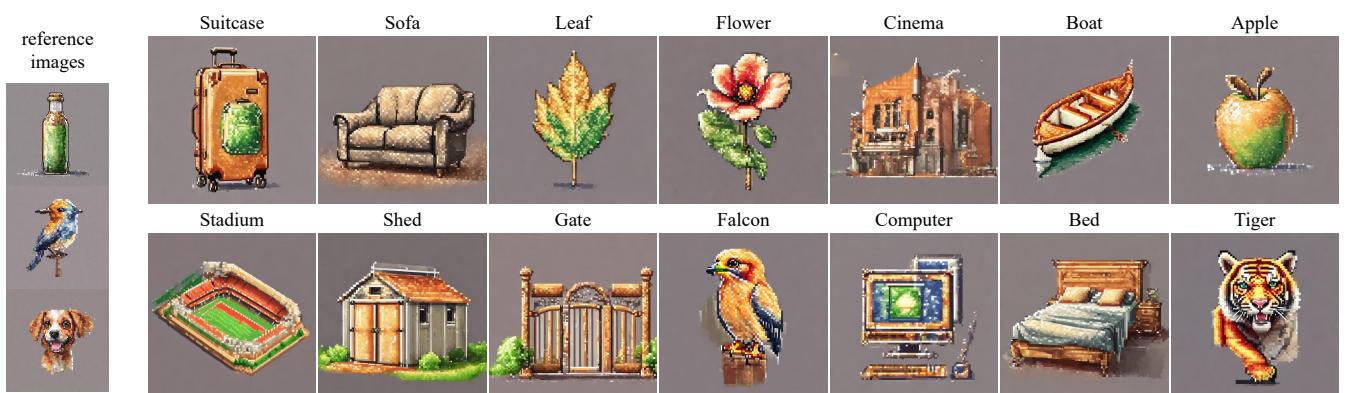
**Figure 19:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 20:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 21:** Generated results of StyleBlend using the style references displayed on the left.



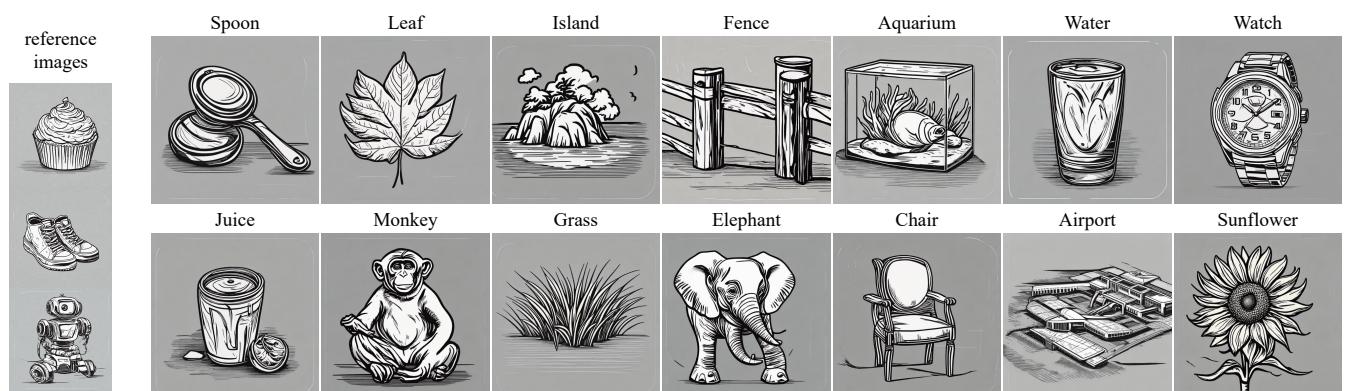
**Figure 22:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 23:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 24:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 25:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 26:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 27:** Generated results of StyleBlend using the style references displayed on the left.



**Figure 28:** Generated results of StyleBlend using the style references displayed on the left.