Leveraging Diffusion Models for Stylization using Multiple Style Images

Dan Ruta   Abdelaziz Djelouah   Raphael Ortiz   Christopher Schroers
DisneyResearch|Studios
dan.ruta@disney.com   abdelaziz.djelouah@disney.com

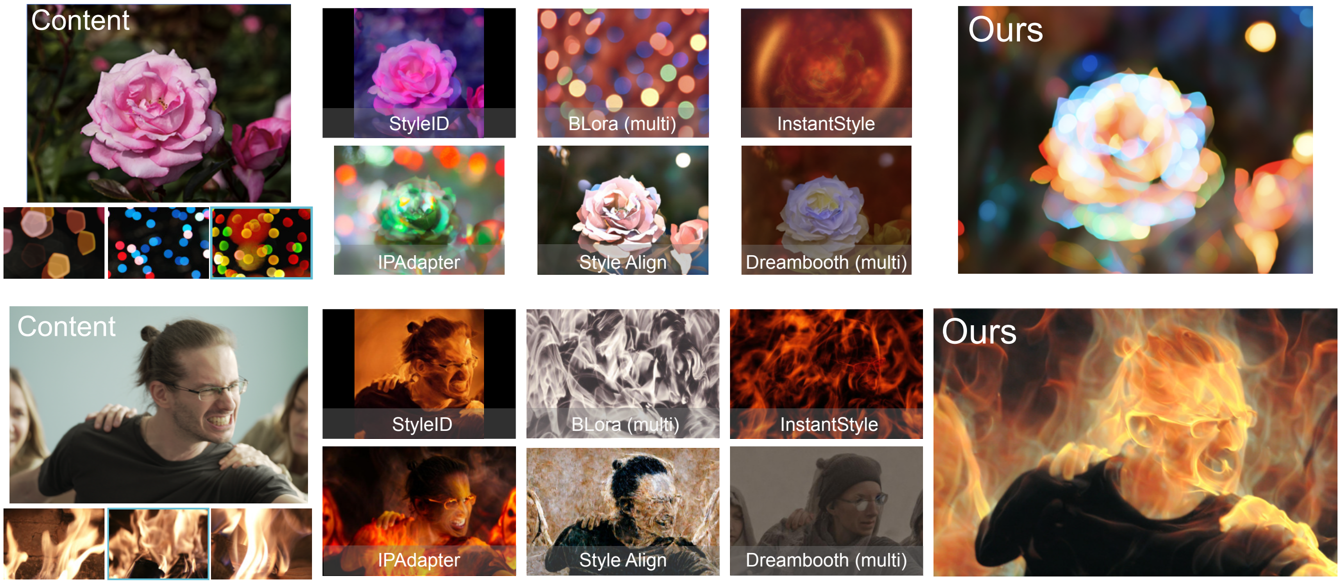

Figure 1. We propose a style transfer method that uses multiple style images and achieves state-of-the-art results. In each case, our result preserves the content while closely matching the style. Existing methods struggle to achieve both content preservation and high quality stylization, even the ones using multiple style images such as Dreambooth [31] and BLora [9]. Besides our result, we indicate with *(multi)* the methods using multiple images. We highlight in cyan the style image provided for single image based stylization methods.

Abstract

Recent advances in latent diffusion models have enabled exciting progress in image style transfer. However, several key issues remain. For example, existing methods still struggle to accurately match styles. They are often limited in the number of style images that can be used. Furthermore, they tend to entangle content and style in undesired ways. To address this, we propose leveraging multiple style images which helps better represent style features and prevent content leaking from the style images. We design a method that leverages both image prompt adapters and statistical alignment of the features during the denoising process. With this, our approach is designed such that it can intervene both at the cross-attention and the self-attention layers of the denoising UNet. For the statistical alignment, we employ clustering to distill a small representative set of attention features from the large number of attention values extracted from the style samples. As demonstrated in our experimental section, the resulting method achieves state-of-the-art results for stylization.

1. Introduction

Artists are in constant exploration to create new artistic renderings, that can offer fresh and different looks. In this context, image style transfer aims at simplifying style exploration with the objective of allowing faster iteration during this artistic research phase.

Recently, diffusion based image stylization methods [6, 11, 43] have shown impressive results. For example, image prompt adaptation methods [43] use style information derived from CLIP-image embeddings. To limit content and style entanglement, it is possible to use statistical alignment [6, 11] of the content and style image during the denoising process. In the context of multiple style images, personalization methods such as Dreambooth [31], Lora [12] or CustomDiffusion [20] can leverage the available style samples for fine-tuning (albeit with different strategies). In all these mentioned works, we observe two issues clearly visible in Figure 1: entanglement of content and style, and lower quality style transfer.

Building on insights from recent works, we design a method that achieves both better content preservation and higher quality stylization, using several style images to address the aforementioned problems. Our approach can be summarized in the following 3 steps: *First*, we fine-tune an image prompt adapter model on the style images. To help disentangle style from content, we compute an average token vector from the style images that will be used as prompt for the diffusion model. *Second*, we distill style features from multiple images through a clustering approach. *Finally*, we adopt a two-stage strategy to achieve high quality results with some control on the structural level at which the stylization happens.

Our contributions are the following:

- A method that combines model adapters and statistics alignment;
- A solution to scale up to multiple style images;
- State-of-the-art stylization results as demonstrated in our thorough evaluation, including a user study.

2. Related work

Image style transfer remains a fundamental challenge in computer vision, aiming to modify the appearance of a content image based on a given reference. Our discussion here mostly focuses on recent works, in particular the ones using latent diffusion models. For a more detailed overview, we refer to the review on style transfer from Jing *et al.* [15].

Optimization based stylization. Early works such as the seminal Gatys style transfer algorithm [10] rely on inference-time optimization to achieve style transfer. As this is generally an impractical time and resource consuming process, the field has focused research efforts on fast zero-shot approaches. Still recent work [19] circled back to an iterative approach and achieved some of the best style transfer results.

Using Multiple Images for style transfer. Recent works have extended NST to incorporate multiple style references, facilitating better style interpolation and mixing. Some approaches [14, 23, 40] focus on learning a robust style representation capable of blending multiple sources seamlessly. Others employ generative adversarial networks (GANs) [17], which, when trained on small datasets, allow for rapid style adaptation through fine-tuning [18, 26]. Despite their success in domain-specific applications such as facial stylization, these methods often struggle with generalization beyond constrained settings. In contrast, diffusion-based models have emerged as a powerful alternative for achieving high-quality, diverse style transfer.

Statistical alignment and moment matching. Style representation can also be captured through statistical properties of images. Early work introduced Adaptive Instance Normalization (AdaIN) [13] and Whitening and Coloring Transform (WCT) [21], which align the mean and variance of content and style features to achieve stylization. More recent techniques extend this paradigm by incorporating higher-order moments (e.g., skewness and kurtosis) to enhance fidelity in style transfer [16, 34]. These methods provide explicit control over style attributes, complementing the implicit representations learned by deep networks.

Diffusion Based. Latent diffusion models (LDMs) [30] have recently revolutionized style transfer, offering three primary strategies: customization, adaptation modules, and feature alignment. Customization techniques, such as DreamBooth [31] and Custom Diffusion [20], fine-tune all model parameters to encode new styles, achieving high-fidelity results at the cost of computational efficiency. Low-Rank Adaptation (LoRA) [12] mitigates some of these inefficiencies by introducing lightweight fine-tuning mechanisms. Alternatively, adaptation modules condition pre-trained diffusion models on external style information. IP-Adapter [43], for instance, transforms CLIP [29] embeddings of style images into inputs compatible with diffusion models, enabling zero-shot style transfer. However, these methods often suffer from content leakage, as they struggle to disentangle content and style effectively.

Feature alignment methods offer a training-free approach to style transfer by manipulating the self-attention layers of diffusion models. DIFF-NST [35] and Style Injection [6] replace attention values from the generated image with those from a style reference, effectively transferring texture and color characteristics. StyleAligned [11] refines this process by incorporating statistical AdaIN operations within attention layers, ensuring robust style adaptation.

Recent advancements aim to improve content-style disentanglement within diffusion models. BLoRA [9] explicitly separates content and style representations using SDXL [28] and LoRA-based fine-tuning. InstantStyle [39] takes a similar approach but introduces feature decoupling and targeted injection into specific layers, reducing content leakage and preserving style fidelity.

3. Multi-Image Style Transfer

Given a set of style images $\mathcal{S} = \{I_1^s, \ldots, I_n^s\}$, our objective is to transform a content image I_c into its stylized version

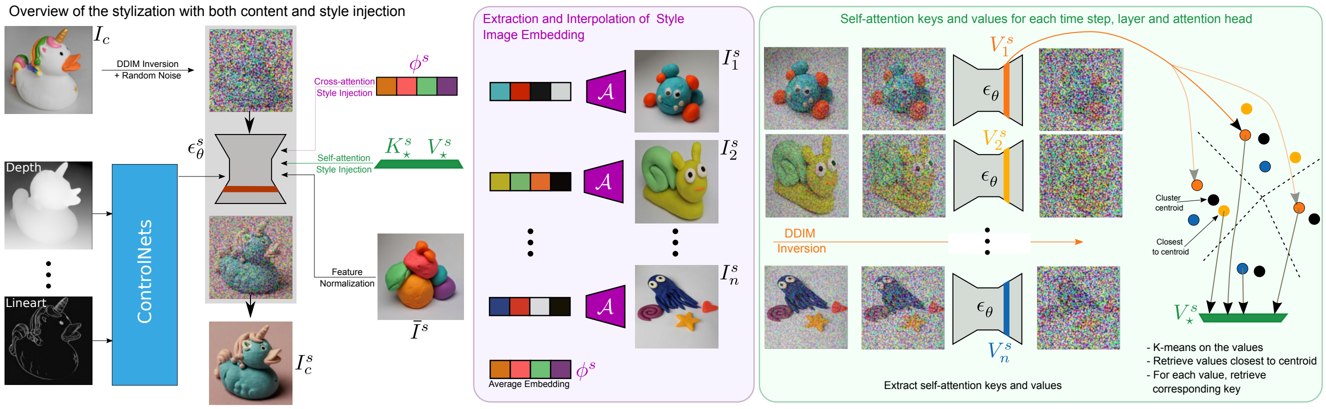

Figure 2. Overview of the diffusion based stylization method. **(Left)** Given a content image I_c we extract line art and depth maps to guide the content during denoising with ControlNets. Style is enforced at the cross and self attention layers. We use the average style embedding ϕ^s, representative attention maps keys and values $(K_\star^s, V_\star^s)$ and an average style image $\bar{I}^s$. **(Middle)** From the set of input style images $\{I_1^s, \ldots, I_n^s\}$, we train the image prompt adapter $\mathcal{A}$. The average embedding ϕ^s is obtained by interpolating the style image embeddings. **(Right)** We use DDIM to invert each style image and extract keys and values for each layer and time step. We use k-means clustering to reduce the keys and values to a manageable size, keeping the ones closest to the cluster centroid. As keys and values are paired, we perform the clustering on the values only, and retrieve their matching keys.

I_c^s, that matches as well as possible the style characteristics while maintaining the original content. Our stylization method leverages pre-trained diffusion models. Specifically we use the latent diffusion model proposed by Rombach *et al.* [30], where an image I is encoded using a variational auto-encoder into its latent representation x. Diffusion and denoising are done on this latent representation, where a denoising model is trained to estimate the noise on x_t at each given step t, with τ_θ transforming the text prompt y and θ denoting the parameters of the model.

$$\epsilon_t = \epsilon_\theta(x_t, \tau_\theta(y), t) \tag{1}$$

We express our stylization problem in the same framework, by making the adjustments needed to the denoising model, adding both content/style images: I_c, and $\mathcal{S}$

$$\epsilon_t = \epsilon_\theta^s(x_t, \tau_\theta(y), t, I_c, \mathcal{S}). \tag{2}$$

We illustrate the adjustments we make to the denoising model ϵ_θ^s on the left side of Fig. 2. We use ControlNets to provide a content related signal to the denoiser using depth and linart features extracted from I_c. Regarding style, we provide guidance at 2 different levels: at the *cross-attention* level, with an average style prompt embedding ϕ^S; at the *self-attention* level with style representative keys $K_s^\star$ and values $V_s^\star$ used in the self-attention mechanism, that we normalize using the average style image $\bar{I}^s$. In the next subsections we present these different aspects of the method in more detail, with corresponding visualizations in Fig 2.

3.1. Prompt Adaptation with Multiple Style Images

With multiple style images, it becomes possible to mitigate the issues observed with image adapters. We propose 2 adjustments: First, the fine-tuning of the image prompt adapter model on the style images. Second, the interpolation of the style image embeddings.

Fine-tuning the Projection Network. The image adapter model [43] takes as input an image, extracts the corresponding CLIP-embedding and then trains a projection network $\mathcal{A}$ to learn the mapping into a sequence of 4 tokens, with dimensions matching the one for text. In the fine-tuning stage we train the model on the style images only, for roughly 100 steps, minimizing the following loss

$$\mathcal{L}_{\mathcal{A}} = \mathop{\mathbb{E}}_{I_i^s \in \mathcal{S}} ||\epsilon - \epsilon_\theta^A(x_t, \tau_\theta(y), t, x_i^s)||^2 \tag{3}$$

with ϵ_θ^A indicating the denoiser consisting of the original model ϵ_θ and the image projection modules. The model is trained to reconstruct the style images $(I_i^s \in \mathcal{S})$, updating only the projection module parameters.

Interpolation of the Style Image Embeddings. Fine-tuning helps better capture the style features, but doesn't address the issue of style and content entanglement. We observe that the different style images share the same style while the content varies. Hence, by averaging the corresponding embeddings we keep the shared property (i.e. the style) while the differences are toned down (i.e. the content). During the stylization of any content image I_c, we will use the average features sequence ϕ^s:

$$\phi^s = \sum_{I_i^s \in \mathcal{S}} \frac{1}{n} \mathcal{A}(I_i^s) \tag{4}$$

The middle part of Figure 2 illustrates this process.

3.2. Feature Alignment with Multiple Images

Although we are able to address content entanglement thanks to our proposed changes, this is not sufficient to achieve high quality stylization as the image prompting doesn't capture all the aspects of the style. As a result, we also consider the statistics of the deep features. Contrary to existing works [6, 11] we use multiple images which requires solving a few technical challenges.

Single image style injection and normalization. Let's start with a single style image I_s^i, similarly to existing alignment methods [6, 11]. Style injection is achieved by first using DDIM inversion to obtain the denoising features for a fixed number of steps T, then injecting the features at each time step t during image stylization.

In the image synthesis pipeline illustrated in Figure 2, the injection of the style features is done in the self-attention layers during the denoising process. More specifically, at every self-attention layer and every time step t, we can modify the attention as follows:

$$\text{Attention}(\hat{Q}_c^t, [\hat{K}_c^t \quad K_s^{i,t}], [V_c \quad V_s^{i,t}]). \tag{5}$$

The keys $K_s^{i,t}$ and values $V_s^{i,t}$ are obtained during the inversion of the style image I_s^i. Whereas content queries Q_c^t and keys K_c^t, are obtained in the current denoising step t, then normalized using the adaptive normalization [13]

$$\hat{Q}_c^t = \text{AdaIN}(Q_c^t, Q_s^{i,t}) \text{ and } \hat{K}_c^t = \text{AdaIN}(K_c^t, K_s^{i,t}). \tag{6}$$

For simplicity, we drop the time step t in the following equations from our notation as the same operations are applied independently of t.

Multi-image style injection. A naive extension to multiple images would be to adjust the attention layer as:

$$\text{Attention}(\hat{Q}_c, [\hat{K}_c \quad K_s^1 \ldots K_s^n], [V_c \quad V_s^1 \ldots V_s^n]). \tag{7}$$

This is however unfeasible, as the attention values extracted from the diffusion generation process for a single image over 50 time steps adds up to almost 7GB of data, without even considering the increase in computations.

Given all attention values extracted from all style images, our solution is to rely on clustering to pick the most unique vectors, for some lower number of total vectors. We use KMeans clustering [24] to cluster values $\{V_s^1, \ldots, V_s^n\}$. After clustering, we sample the value vector v closest to each centroid, and select the matching key k. The result is a new set of keys $K_s^\star$ and values $V_s^\star$ which represent the style better than a single image while being as compact

$$\text{Attention}(\hat{Q}_c, [\hat{K}_c \quad K_s^\star], [V_c \quad V_s^\star]). \tag{8}$$

In practice, we aim to have a number of clusters that matches the target number of vectors found in a single image (the typical count varies across the UNet), to compress only the most important and unique style concepts from across all style images. This is not a hard limit, and future experiments scaling the number of clusters can be explored. We perform GPU-accelerated clustering of attention values from each UNet layer, each timestep, and attention head, separately. This separation enables strong parallelization.

Normalization through an average style image. When we scale up to multiple images, the normalization process needs to account for the attention values from all the style images. The distribution of attention values from multiple images can be multimodal, and computing a mean across these different groups of features results in a value which falls outside the distribution of any individual image, and produces failed or sub-optimal results.

An effective solution to resolve this issue is to instead use an *average* style image generated with our multi-image prompt adapter. Providing only the average style feature embedding ϕ^s, and no other guidance for the generation process, produces an *average style image* $\bar{I}^s$ with random content. The content depicted in $\bar{I}^s$ is unimportant; however the attention values do fall within the same single distribution, while covering a wider range of the style (albeit less than in the clustered values). We can extract the statistics from these attention values, for use in the normalization and alignment.

$$\hat{Q}_c = \text{AdaIN}(Q_c, \bar{Q}^s) \quad \text{and} \quad \hat{K}_c = \text{AdaIN}(K_c, \bar{K}^s) \tag{9}$$

where $\bar{Q}^s$ and $\bar{K}^s$ the queries and keys obtained when generating the average style image $\bar{I}^s$, respectively.

In addition to the first two moments, higher orders such as skewness and kurtosis can be used. Finally, we also align the statistics of the latents at each timestep with the statistics of the latents from $\bar{I}^s$. This improves style and color quality.

3.3. High Resolution and Texture Quality

In addition to the core aspects of the method, there are a few key points to take into consideration to achieve best quality results on high resolution output images.

High Resolution Output. We use Stable Diffusion as our core model. It can be used to generate almost arbitrary resolutions and aspect ratios. However, scaling the images to higher resolutions, such as 1024px and larger, can lead to tiling artifacts (in part due to the fact that it is trained at a resolution of 512px). To avoid this issue, we can rely on the timestep-based content disentanglement described by Wu *et a.* [42], and focus only the first few timesteps on generating the image structure. We can use a lower resolution for this,

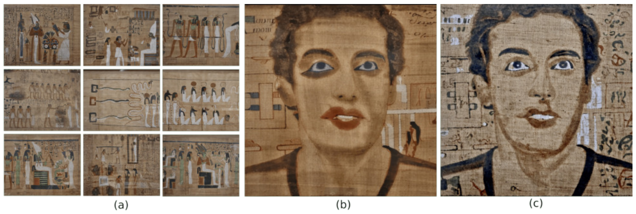

Figure 3. Visualization of the effect of our two-stage approach. (a) Starting from the set of style images, (b) the first stage (in low resolution) produces an initial stylized output however texture details are missing. (c) The second stage helps to add much more detail from the style images.

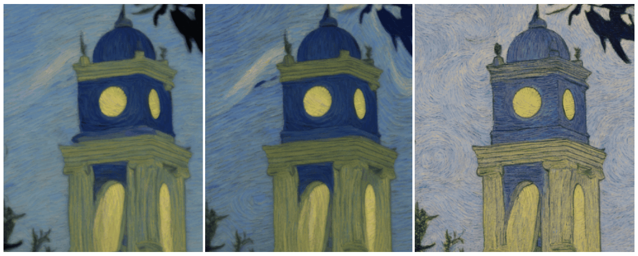

Figure 4. It is possible to control the scale of the textures in the syle transfer output, through scaling the style images used in the cross-attention. In this example, *Starry Night* painting is used as style source. Changing the scale of the style image crops (from left to right), has a clear effect on the brush strokes texture.

targeting 512px on the shortest side. We can then either spatially resize the latents and resume the rest of the timesteps, or first generate the image in lower resolution then scale up and add details through an image-to-image operation. As illustrated in Figure 3, this process avoids tiling artifacts, while allowing the later timesteps to add the lower level stylistic and textural details of a higher resolution image. We use ControlNet depth and LineArt inputs computed separately for the lower and higher resolution content images, during this process, to accurately condition the structure for each appropriate resolution.

Better Texture Quality. The CLIP embeddings require the input image to be downsampled, but stylistic details are lost in such a rescaling operation. To remedy this issue, we instead opt to compute our final embedding from a number of patches extracted from the high resolution images. Thus treating each as a separate style image, in our multi-image pipeline. In addition to this, the selection of image crops offers a control over the scale of the style information. Smaller crops can guide the stylization to focus more on the low level textural details of a style such as brush strokes and lines, whereas larger crops can place more importance on the structural components of items depicted in the style image. Fig. 4 visualises this effect, where varying the sizes of the crops can account for such artistic intent.

4. Experiments

4.1. Implementation Details.

We use Stable Diffusion v1.5 for all our experiments, but we show in supplementary material that our method generalizes to other backbones. We use the standard ControlNet depth and LineArt. Our fine-tuning of the IPAdapter [43] takes about 3 to 5 minutes, depending on the number of style images (can be any number). We use the GPU-accelerated Faiss library for clustering, and we implement heavy parallelization, allowing us to run several concurrent instances on the same RTX 4090. This takes under half an hour for a dozen style images and varies depending on their aspect ratios. Once models and data are loaded, stylization on the same machine elapses roughly 16 seconds, also depending on the aspect ratio.

4.2. Data and Metrics.

We construct a mini dataset for use in testing neural style transfer quality. We form this test set from 50 content images and 200 style images across 15 style groups. Previous style datasets such as BAM [41], BBST-4M [33], and WikiArt [36], big or small, are either not licensed openly, not available, or contain style images not grouped into style-consistent groupings. We compose this dataset from images that are completely public domain, or from personal images for both content and style sets.

We use 5 automated metrics for quantitative experiments, computed between each stylized image compared to each of the style images in their respective styles, and averaged. We use SIFID [37] to measure patch-based style similarity, computed as an FID score between only a pair of images. We use Chamfer distance to measure colour similarity, normalized by the number of pixels to avoid varying image resolution skewing the results. We also use two style embeddings, CSD [38], and ALADIN [32], for measuring the similarity of style in a model's embedding space. In all cases a lower value is preferable. Finally, we use similarity in DINOv2 [1] space - this time a higher value is preferable.

4.3. Comparisons

We compare against a large range of stylization techniques, some based on diffusion while others are not, and some using a single style image, while others can use many. Among the techniques not using diffusion we can mention NNST [19], AdaAttn [22] or SANet [27]. Among the diffusion methods we compare with IPAdapter [43], StyleAligned [11] and StyleInject [6], and also with other recent works such as InST [46] and StyleID [5]. Some methods are able to leverage multiple style images, like Dreambooth [31] and BLoRA [9].

For methods using multiple style images (including ours), the entire set is used. For methods using a single style

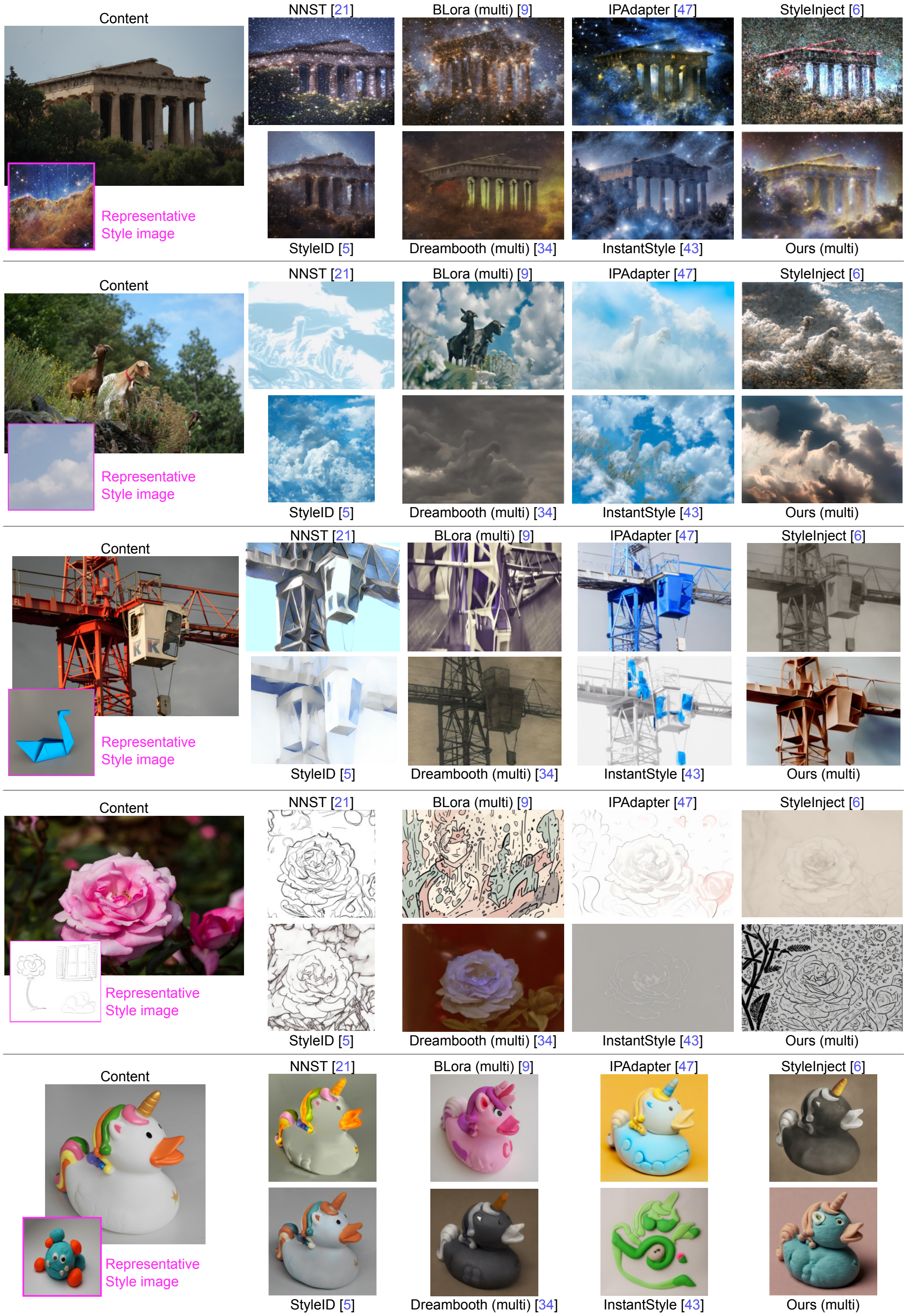

Figure 5. Qualitative results comparison of our method, compared to some of the baselines. Besides ours, we indicate with *(multi)* the methods using multiple images. For the methods using a single style image, we highlight with a colored rectangle the style image that was provided.

image, there is the question of which style to select for each content. To better represent the performance of these single image style transfer methods, we consider all possible combination of style and content images. However, since

Method	SIFID ↓	Chamfer ↓	CSD ↓	ALADIN ↓	DINO ↑
IPAdapter [43]	3.427	83.632	1.091	1.109	0.654
StyleAligned [11]	3.988	8.716	1.326	1.362	0.597
NNST [19]	4.152	136.210	1.165	1.228	0.622
DIFF-NST [35]	6.315	130.956	1.254	1.291	0.600
CAST [45]	2.535	10.172	1.273	1.275	0.558
NeAT [33]	2.808	8.282	1.239	1.278	0.567
StyleID [5]	2.798	139.146	1.222	1.224	0.599
MCCNet [7]	2.592	6.830	1.264	1.274	0.570
StyTr2 [8]	3.136	137.742	1.202	1.213	0.616
PAMA [25]	2.759	8.400	1.238	1.283	0.603
SANet [27]	3.541	**6.289**	1.227	1.252	0.601
AdaAttn [22]	3.724	10.747	1.277	1.278	0.613
ContraAST [2]	2.836	6.699	1.226	1.233	0.592
AdaIN [13]	2.677	6.450	1.265	1.326	0.576
InST [46]	7.032	97.497	1.179	1.227	0.614
S2Wat [44]	3.104	133.548	1.223	1.243	0.595
Dreambooth [31]	5.303	306.707	1.391	1.376	0.571
BLoRA [9]	2.870	40.799	1.120	1.118	0.658
BLoRA (multi) [9]	2.512	30.257	1.119	1.097	0.666
InstantStyle [39]	4.479	64.331	1.150	1.107	0.668
Ours	**2.040**	17.042	**1.088**	**1.054**	**0.680**

Table 1. Quantitative metrics comparing our method against baselines. Chamfer values are scaled $\times 10^{-3}$ for clarity.

this would be too computationally involved, we randomly subsample 10% of these combinations. The objective is to average out the effect of the particular style image selected, and better represent the performance of each method.

Quantitative Evaluation. We present the results of the quantitative evaluation in Table 1. We use several metrics (Sec. 4.2) to evaluate the performance of the different methods. From the evaluation it is clear that ours performs best.

It is also interesting to point to the case of BLoRA [9], which can use a single or multiple style images. Here the usage of multiple style images improves the performance which reinforces our message that having access to multiple style images helps to better capture the style. In the case of Dreambooth [31], the training process itself is less stable and no single training setting (learning rate, number of steps, etc.) is able to achieve good results on all the styles. For this evaluation we have selected a setting that performs well on a few style groups and used it for all the rest. The Chamfer metric measures color matching and we argue it is less important for the style transfer task. This is confirmed next in the qualitative results and the user study.

Qualitative Results. In Figure 5 we show a variety of content images stylized according to different sets of style images. Our method is the only approach that performs well across this wide variety of styles and content. Using multiple style images helps capture the style, but this is not sufficient as it can be observed from the results of BLoRA (multi) [9] or Dreambooth [31]. Of note is the performance of NNST [19] on styles that are mostly operating on low-level features, with degradation on examples that need a better semantic understanding of the content and style.

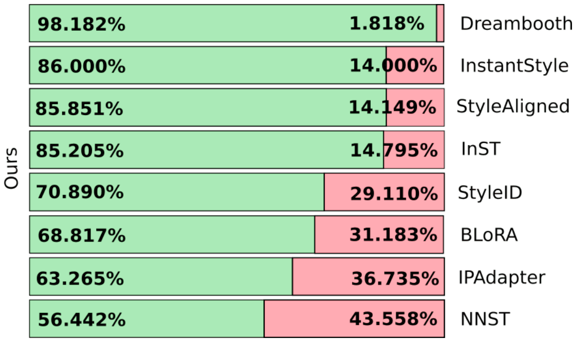

Figure 6. Preference scores of our method, compared to each baseline. Besides Ours, Dreambooth and BLoRA use multiple style images.

User Study. For the user study, we select a subset of the most relevant techniques to present them for comparison. We select primarily diffusion based techniques, but also NNST [19], a strong traditional technique. We present a private team of 23 diverse workers with a labeling task, where a selection of real style images is shown, alongside a shuffled pair of stylization results. One from our own method, and one from a baseline method. We additionally show the real content image being stylized, for context. We ask users to examine the style images, both stylized images, and to select the stylized image which best matches the style for all the real style images shown. In supplementary material, we show a screenshot of an example task shown to a worker. Figure 6 displays the user preference of our method, compared to baselines. Our work outperforms all other methods. It is interesting to note the good performance of NNST [19] in this user study. This illustrates that the metrics (Table 1) do not cover all aspects of the problem. Users tend to favor NNST [19] when the low level features of the style are well preserved, which works well with many of the styles present in this user study.

4.4. Ablation Study

Clustering. As mentioned previously, the clustering step is necessary as the statistical alignment process does not scale well, and is limited to around 3 images on a single GPU with 24GB of VRAM. Still, we would like to evaluate any difference or loss in quality due to using the selected values from clustering instead of using all the available attention data. To make this comparison possible, we apply the stylization using dynamic loading of attention values from disk for use in the concatenation step of $\mathcal{K}$ and $\mathcal{V}$ self-attention values. Of course, this step introduces an extremely unpractical amount of disk reading overhead. For a small set of 9 style images and using an RTX 4090 GPU, this dynamic process needs around 9 minutes and 40 seconds on average for stylizing one image. When using clustered values takes around 14 seconds. Figure 7 shows that the clustering strategy has little effect on the style transfer

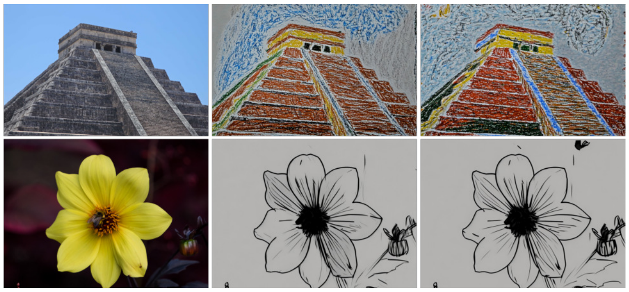

Figure 7. The left-most column shows the content. While being significantly faster (14s vs 10min), the style transfer using attention clustering (middle) has very negligible impact compared to dynamic loading (right-most).

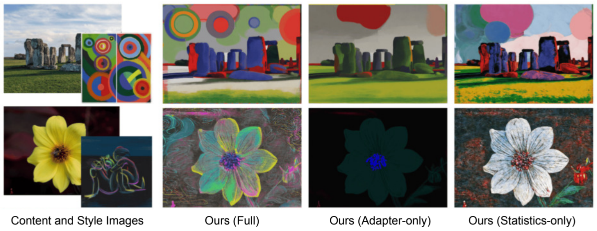

Figure 8. Ablation of cross-attention (adapter-based) and self-attention (statistics-based) components

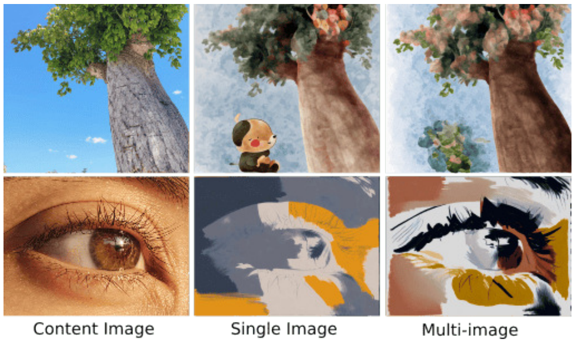

Figure 9. Comparison of stylization using either a single source style image, or several. In the single image case, content entanglement can emerge through the erroneous insertion of content from the style images into the results (here, an animal). On the bottom row, we see a far narrower range of colors and brush stroke compared to the range contained in the style group as a whole.

results, while being much faster at inference time.

Self-attention vs Cross-attention. We describe our method as having two main components. First, cross-attention components, using an image adapter module to inject style features into the diffusion model. Second, a self-attention component using statistics alignment and concatenation on the self-attention values. Both components play a valuable role, as we visualize in Fig 8. The statistics component more heavily affects the global appearance of the image, not as heavily dependent on the content, whereas the adapter component more heavily modifies the depicted content. Together, the global style of the image is more thoroughly stylized, while also modifying the content.

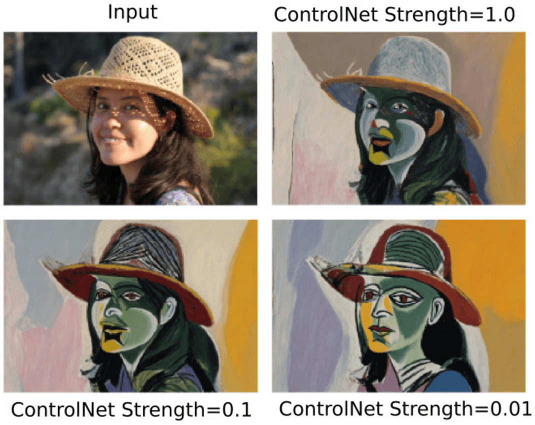

Figure 10. The weight of the LineArt ControlNet can be used to control deformation, when used together with depth ControlNet. A lower strength leads to higher deformation, which may be desirable for certain styles.

Benefits of Using Multiple Style images. Our strategy of image adaptation and feature alignment is applicable with single image stylization. However using multiple style images, is always beneficial (Fig. 9). We avoid content entanglement in the results and achieve higher stylization quality.

ControlNet Strength for deformation. ControlNet affects the visual style elements in a stylized image. The importance weight of this auxiliary conditioning can expose artistic controls over the stylization process to artists. For example, as shown in Fig 10, a lower strength LineArt ControlNet can result in favorable output when used together with a ”deformed” style such as Picasso’s cubism.

5. Discussion

We propose a model-agnostic diffusion-based style transfer technique that leverages multiple source style images. We avoid entanglement issues and encode more style variance from a wider range of style examples. We show in the quantitative evaluation metrics and user studies that our method is state-of-the-art. We already show initial results with the larger SDXL [28] model. Others, such Pixart-α(-σ) [3, 4], could be tested.

A key identified limitation of both our technique and the other methods in literature is the lack of control over more specific or specialized aspects of the stylization, such as line work. This is an interesting and important future direction of research.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 5

[2] Haibo Chen, Lei Zhao, Zhizhong Wang, Zhang Hui Ming, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Artistic style transfer with internal-external learning and contrastive learning. In *Advances in Neural Information Processing Systems*, 2021. 7

[3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. 8

[4] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024. 8

[5] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. *arXiv preprint arXiv:2312.09008*, 2023. 5, 7

[6] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8795–8805, 2024. 1, 2, 4, 5

[7] Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. Arbitrary video style transfer via multi-channel correlation. *CoRR*, abs/2009.08003, 2020. 7

[8] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr^2: Image style transfer with transformers, 2022. 7

[9] Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. Implicit style-content separation using b-lora, 2024. 1, 2, 5, 7

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[11] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention, 2024. 1, 2, 4, 5, 7

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[13] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. ICCV*, 2017. 2, 4, 7

[14] Zixuan Huang, Jinghuai Zhang, and Jing Liao. Style mixer: Semantic-aware multi-style transfer network. In *Computer Graphics Forum*, pages 469–480. Wiley Online Library, 2019. 2

[15] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019. 2

[16] N. Kalischek, J. D. Wegner, and K. Schindler. In the light of feature distributions: moment matching for neural style transfer. *CoRR*, abs/2103.07208, 2021. 2

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[18] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020. 2

[19] Nicholas Kolkin, Michal Kucera, Sylvain Paris, Daniel Sykora, Eli Shechtman, and Greg Shakhnarovich. Neural neighbor style transfer, 2022. 2, 5, 7

[20] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion, 2022. 2

[21] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *CoRR*, abs/1705.08086, 2017. 2

[22] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. *CoRR*, abs/2108.03647, 2021. 5, 7

[23] Zhi-Song Liu, Vicky Kalogeiton, and Marie-Paule Cani. Multiple style transfer via variational autoencoder. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2413–2417. IEEE, 2021. 2

[24] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 4

[25] Xuan Luo, Zhen Han, Lingkang Yang, and Lingling Zhang. Consistent style transfer. *CoRR*, abs/2201.02233, 2022. 7

[26] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10743–10752, 2021. 2

[27] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. *CoRR*, abs/1812.02342, 2018. 5, 7

[28] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 2, 8

[29] A. Radford, J. Wook Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 2, 3

[31] 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, 2023. 1, 2, 5, 7

[32] Dan Ruta, Saeid Motiian, Baldo Faieta, Zhe Lin, Hailin Jin, Alex Filipkowski, Andrew Gilbert, and John Collomosse. Aladin: All layer adaptive instance normalization for fine-grained style similarity, 2021. 5

[33] Dan Ruta, Andrew Gilbert, John Collomosse, Eli Shechtman, and Nicholas Kolkin. Neat: Neural artistic tracing for beautiful style transfer, 2023. 5, 7

[34] Dan Ruta, Gemma Canet Tarres, Alexander Black, Andrew Gilbert, and John Collomosse. Aladin-nst: Self-supervised disentangled representation learning of artistic style through neural style transfer, 2023. 2

[35] Dan Ruta, Gemma Canet Tarrés, Andrew Gilbert, Eli Shechtman, Nicholas Kolkin, and John Collomosse. Diff-nst: Diffusion interleaving for deformable neural style transfer, 2023. 2, 7

[36] Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature, 2015. 5

[37] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. *CoRR*, abs/1905.01164, 2019. 5

[38] Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Measuring style similarity in diffusion models, 2024. 5

[39] Haofan Wang, Matteo Spinelli, Qixun Wang, Xu Bai, Zekui Qin, and Anthony Chen. Instantstyle: Free lunch towards style-preserving in text-to-image generation, 2024. 2, 7

[40] Quan Wang, Sheng Li, Zichi Wang, Xinpeng Zhang, and Guorui Feng. Multi-source style transfer via style disentanglement network. *IEEE Transactions on Multimedia*, 26: 1373–1383, 2023. 2

[41] Michael J. Wilber, Chen Fang, Hailin Jin, Aaron Hertzmann, John Collomosse, and Serge Belongie. Bam! the behance artistic media dataset for recognition beyond photography. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 5

[42] Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in text-to-image diffusion models, 2022. 4

[43] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023. 1, 2, 3, 5, 7

[44] Chiyu Zhang, Jun Yang, Lei Wang, and Zaiyan Dai. S2wat: Image style transfer via hierarchical vision transformer using strips window attention, 2022. 7

[45] 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. 7

[46] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models, 2023. 5, 7