Diffusion in Style

Martin Nicolas Everaert[1] Marco Bocchio[2] Sami Arpa[2] Sabine Süsstrunk[1] Radhakrishna Achanta[1]
[1]School of Computer and Communication Sciences, EPFL, Switzerland [2]Largo.ai, Lausanne, Switzerland
Project page: https://ivrl.github.io/diffusion-in-style/

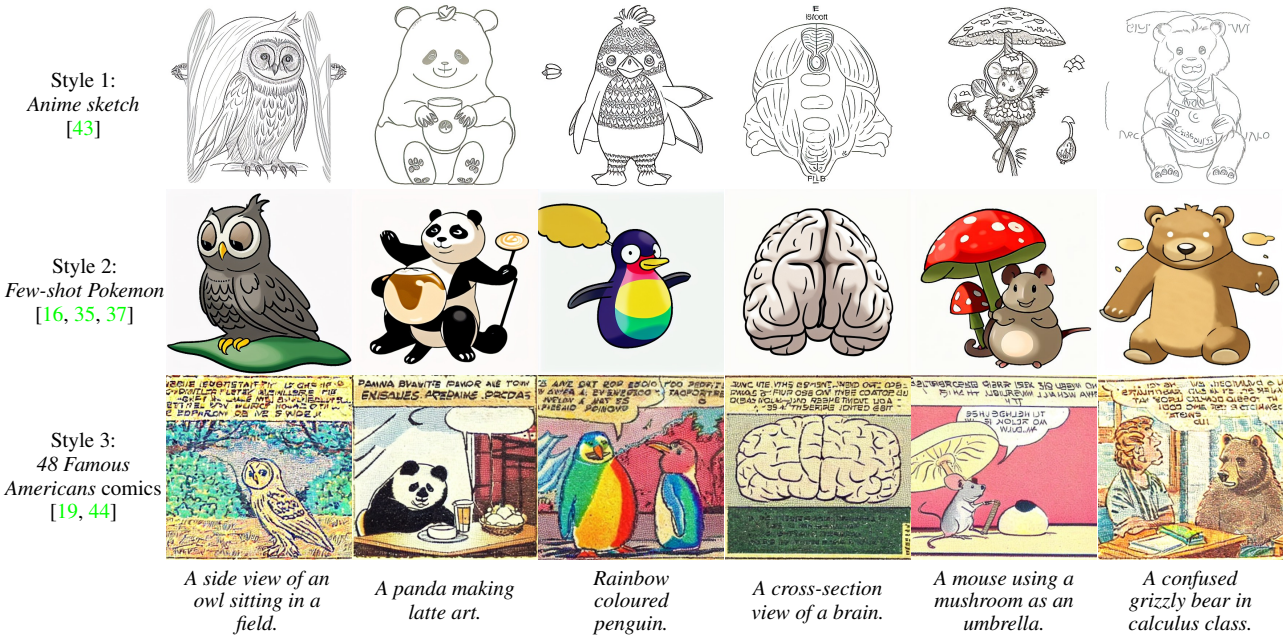

Figure 1. **Examples of style adaptations induced by *Diffusion in Style*.** A small number of target style images are used to efficiently adapt Stable Diffusion to a desired style: *anime sketch* (first row, [43]), *few-shot Pokemon* (middle row, [16, 35, 37]), and the *48 Famous Americans* comics (bottom row, [19, 44]). The adapted model can generate images in the desired style with any textual prompt, without conditioning on a target style image at inference time. Each column is generated from only the textual prompt indicated at the bottom.

Abstract

We present Diffusion in Style*, a simple method to adapt Stable Diffusion to any desired style, using only a small set of target images. It is based on the key observation that the style of the images generated by Stable Diffusion is tied to the initial latent tensor. Not adapting this initial latent tensor to the style makes fine-tuning slow, expensive, and impractical, especially when only a few target style images are available. In contrast, fine-tuning is much easier if this initial latent tensor is also adapted. Our* Diffusion in Style *is orders of magnitude more sample-efficient and faster. It also generates more pleasing images than existing approaches, as shown qualitatively and with quantitative comparisons.*

1. Introduction

Generating images of a specific style using large-scale text-to-image models, such as Stable Diffusion [24], is an attractive idea, owing to the high quality of the output images. However, enforcing a coherent style on the generated images is not straightforward. Describing the style in the input textual prompt is often insufficient to obtain images in the desired style. As a consequence, the model needs to be fine-tuned. Yet, current approaches for fine-tuning Stable Diffusion to a particular style suffer from one or more of the following limitations: results can be far from aesthetically pleasing [40], results may not match the desired style precisely [38, 39], the method may require impractical amounts of data and computational resources [46, 47], or fine-tuned

models may undergo catastrophic forgetting [36].

To generate images, Stable Diffusion uses a U-Net [25] to progressively denoise a tensor in the latent space of a Variational Auto-Encoder (VAE) [13]. This latent tensor is initially sampled from a standard Gaussian distribution. The U-Net is conditioned on a textual prompt, preprocessed by a CLIP text encoder [23], to iteratively denoise the noisy latent tensor. Finally, the denoised latent tensor is passed through the VAE decoder to obtain the generated image.

We observe empirically that the initial latent tensors influence the style and layout of generated images. Images generated with the same initial latent tensor and different textual prompts often lead to images with shared attributes, such as similar colors, brightness, and object positioning. We therefore hypothesize that the standard Gaussian distribution, from which the initial latent tensors are sampled, prevents generating images in a desired style.

We propose *Diffusion in Style*, a new method for adapting Stable Diffusion to a target style. The key idea behind *Diffusion in Style* is to start the denoising process with style-relevant initial latent tensors. We obtain the style-specific distribution of initial latent tensors by simply estimating the element-wise mean and standard deviation of the latent encodings of a small set of target style images. This leaves us, in a second step, with a simple fine-tuning that requires orders of magnitude fewer images and/or training iterations than the previous approaches. *Diffusion in Style* generates visually pleasing results and does not suffer from catastrophic forgetting. The highlights of *Diffusion in Style* are:

(1) To our knowledge, it is the first method that modifies the initial latent distribution for style adaptation.

(2) *Diffusion in Style* requires only a small amount of images from the target style, typically 50 to 200. This opens the door to many practical applications where thousands of images of the desired style might not be available. Through minor modifications presented in Section 6, our method can also work with as few as 3 target style images.

(3) *Diffusion in Style* is computationally efficient. Fine-tuning the U-Net on the style-specific distribution takes less than 20 minutes on a Tesla V100 GPU.

We evaluate *Diffusion in Style* quantitatively and qualitatively, and compare it to existing alternatives: prompt engineering, classical fine-tuning [36], LoRA-based fine-tuning [10, 42], and state-of-the-art image translation [2]. As presented in Figures 6, 7, and 8, *Diffusion in Style* consistently outputs better qualitative results than prior art.

2. Related Work

2.1. Latent space statistics for style representation

The style of a set of images can be assessed using statistics or correlations between features of a neural network, as widely done in neural style transfer works [6, 11, 12, 31]. The style can be thought of as the feature distribution [15], for example, mean and covariance of deep features of an Inception model [30] trained on style classification [32] or mean and variance of features in a VGG model [22, 29].

In our work, we use the element-wise mean and variance of the latent tensors as a prior of the style. In a second step, Stable Diffusion is fine-tuned on target style images, which is much more efficient to do using our style prior.

2.2. Controlling the style of Stable Diffusion

Adapting Stable Diffusion to a particular style is typically done by *prompt engineering*, or by *fine-tuning* the U-Net on a set of target style images.

Prompt engineering. A natural way to influence the style of the generated images is to describe the style in the textual prompt. Modifier words or sentences [21] are typically added to the prompts. Examples of these are names of artists, *e.g.*, "Alphonse Mucha"; art forms, *e.g.*, "#pixelart"; visual art types, *e.g.*, "in the style of a cartoon"; camera parameters, *e.g.*, "Polaroid" or "80mm Sigma f/1.4"; etc. But text alone frequently falls short of accurately describing a desired style. Furthermore, this approach utterly fails in straightforward scenarios such as "on a white background" [34].

Textual Inversion [5] introduces a new way of prompt engineering, by way of learning new "words" from a small set of exemplar images. To generate images of a specific concept or particular style, Textual Inversion optimizes the vocabulary strategy of new tokens in a frozen diffusion model. However, akin to the prompt engineering technique, this approach is limited by text embedding's capacity to capture the style characteristics. Since the U-Net is frozen, the output is also confined to the model's initial output domain [26], and hence, may not accurately match the intended style.

Fine-tuning Stable Diffusion. It is possible to fine-tune the U-Net of Stable Diffusion on a set of images [36], but it takes an extensive amount of computational power or images. To cite a few, Waifu Diffusion v1.4 [46] and OpenJourney v2 [47], required tens of thousands of target images and underwent fine-tuning for up to almost a quarter million iterations. When the disparity between the target style and the natural image domains is too big, classical fine-tuning is also not sufficient. For instance, Text-To-Pokemon [38] was fine-tuned for 15k iterations, with a batch size of 4, on the 833 images [16, 35, 37] that all have a white background. But the images it generates frequently lack white background.

To save computational resources, more parameter-efficient fine-tuning methods, such as LoRA [10], can be used to reduce computational resources. Nonetheless, fine-tuning based on LoRA [42] matches the style less precisely [39] than usual fine-tuning.

With 3 to 5 examplar images, one can fine-tune Stable Diffusion to specific objects or people using a method called DreamBooth [26]. However, using it for style adaptation is

not covered by the original paper [26] and is more challenging because the optimal settings and regularization images for that case remain unclear. While DreamBooth initially requires about 1000 iterations and 3-5 images, some models tuned with DreamBooth on particular styles required thousands of target style images and up to 400k iterations [41].

Gradient guidance. Gradient guidance is a technique to influence the generated images to have desired characteristics. It entails using a frozen auxiliary model, for instance, an image classifier [4] or a CLIP model [20]. The gradient of the score predicted by the auxiliary model is incorporated into the noise predicted by the diffusion model [4] at each denoising step, influencing image generation. Because of forward and backward passes on the auxiliary model, generation is significantly slower compared to without gradient guidance. Pan *et al*. [22] propose a similar technique with a style feature function as the auxiliary model. They are able to generate images in desired styles with the GLIDE diffusion model [20]. However, we were unable to obtain comparable results with this technique on Stable Diffusion. We hypothesize that the initial latent tensors might influence the generated images more strongly with Stable Diffusion.

Towards our approach. As observed empirically [33], images generated with Stable Diffusion from the same initial latent tensor often share attributes, such as similar colors and object positioning. Additionally, Meng *et al*. [18] show it is possible to maintain some attributes of a reference image by denoising a noisy version of the reference image instead of a random initial latent tensor. This requires using the lowest timesteps only. Given all the above observations, we foresee that, in Stable Diffusion, the standard noise distribution, from which the initial latent tensors are sampled, prevents generating images in a desired style, and should be adapted to a style-specific noise distribution.

2.3. Modifying the forward diffusion process

Bansal *et al*. [1] show that diffusion models can be generalized to image deteriorations other than noise in the forward diffusion process. These include, for instance, blurring, masking, or pixelating. Daras *et al*. [3] show that such types of image corruption can result in faster image generation and better image quality than regular image diffusion models with Gaussian noise. These approaches re-train diffusion models to perform other reverse diffusion tasks, *e.g*., deblurring or super-resolution instead of denoising.

Instead of changing the type of image degradation, *e.g*. to pixelation or blur, we propose to change standard noise distribution $\mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d\times d})$ to a style-specific noise distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$. By not changing the type of forward diffusion, but only the location $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ of the noise distribution, we avoid re-training Stable Diffusion from scratch, and fine-tune it to the style-specific distribution for only 1000 iterations and a small set of images.

3. Diffusion in Style

In Stable Diffusion, the forward diffusion, *i.e*., the noising process, degrades the training data using noise sampled from zero-mean identity-covariance multivariate Gaussian distribution, $\mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d\times d})$, in the latent space of a VAE with $d = 4 \times 64 \times 64$ dimensions. As discussed, style adaptation of Stable Diffusion is currently done with the same noise distribution, as visualized in the first row of Figure 2. We present here a simple way to achieve superior results, which is also computationally inexpensive. The key idea of our two-step method is to use a style-adapted noise distribution for noising and for sampling the initial latent tensors. Our method adapts the initial latent distribution to the style. By requiring relatively few target images, typically 50 to 200, individual artistic styles can be leveraged.

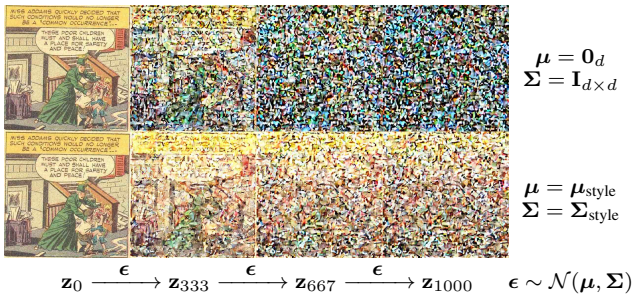

Figure 2. **Conventional versus style-adapted forward diffusion,** visualized in the image space via the VAE decoder $\mathcal{D}$. Conventional forward diffusion process for Stable Diffusion (top row), and our forward diffusion process adapted to the *48 Famous Americans* style (bottom row). $\mathbf{z}_0 \in \mathbb{R}^d$ is the VAE encoding of an original image from the target style. $\mathbf{z}_0$ is degraded with a noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to obtain more and more noisy latent tensors $\mathbf{z}_t, t \in [1 \,..\, T]$. Targeting the *48 Famous Americans* style, we obtain better results when the model is fine-tuned with the diffusion process of the bottom row.

3.1. Step 1: Adapting the noise distribution

In the first step, we obtain a noise distribution that is better suited for the target style. Our style-adapted noise distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$ has a location $\boldsymbol{\mu}_{\text{style}} \in \mathbb{R}^d$ and a diagonal covariance matrix $\boldsymbol{\Sigma}_{\text{style}} = diag(\boldsymbol{\sigma}^2_{\text{style}}) \in \mathbb{R}^{d\times d}$ with diagonal $\boldsymbol{\sigma}^2_{\text{style}} \in \mathbb{R}^d$. In other words, each element $\boldsymbol{\epsilon}_k$ of the noise $\boldsymbol{\epsilon} \in \mathbb{R}^d$ is sampled from $\mathcal{N}(\boldsymbol{\mu}_{\text{style},k}, \boldsymbol{\sigma}^2_{\text{style},k})$, independently of other elements. Note that, in the original Stable Diffusion, the location $\boldsymbol{\mu} \in \mathbb{R}^d$ equals to $\mathbf{0}_d$ and the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ equals to $\mathbf{I}_{d\times d}$. We compute values for the style-adapted location $\boldsymbol{\mu}_{\text{style}}$ and the diagonal $\boldsymbol{\sigma}^2_{\text{style}}$ of the covariance matrix from a set of target style images I_{style}. To this end, we encode the images $i \in I_{\text{style}}$ of the target style with the VAE encoder $\mathcal{E}$, getting the latent tensors $\mathcal{E}(i) \in \mathbb{R}^d$. We then estimate the mean and variance of each element of those latent tensors, to obtain the new

location $\boldsymbol{\mu}_{\text{style}}$ and covariance matrix diagonal $\boldsymbol{\sigma}^2_{\text{style}}$:

$$\begin{aligned} \forall k \in [1\,..\,d], \boldsymbol{\mu}_{\text{style},k} &= \text{Mean}_{i \in I_{\text{style}}} \mathcal{E}_k(i) \\ \forall k \in [1\,..\,d], \boldsymbol{\sigma}_{\text{style},k} &= \text{Std}_{i \in I_{\text{style}}} \mathcal{E}_k(i) \end{aligned} \tag{1}$$

As simple as it is, sampling the initial latent tensors $\hat{\mathbf{z}}_T$ from the noise distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$ helps style-adapting Stable Diffusion very efficiently. As we illustrate in Figures 2 and 3, it can be understood intuitively that our adapted noise distribution better represents the target style, while the original noise distribution $\mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d\times d})$ better represents the entire set of original training images. Thus, it makes sense to sample the initial latent tensor from the noise distribution adapted to the style rather than from the style-agnostic one.

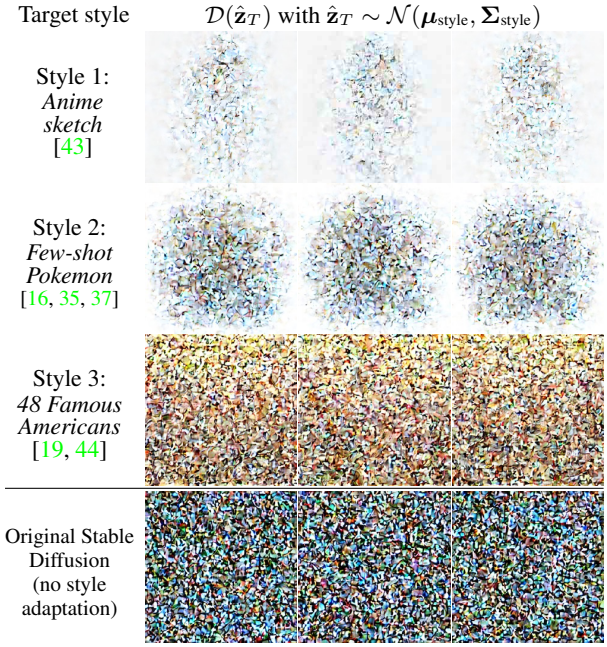

Figure 3. **Samples from the style-specific noise distributions**, visualized in the image space via the VAE decoder $\mathcal{D}$. For each style, we show three initial latent tensors $\hat{\mathbf{z}}_T$ randomly sampled from the adapted noise distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$. Additional styles and visualizations of $\boldsymbol{\mu}_{\text{style}}$ are also presented in Figure 9.

3.2. Step 2: Fine-tuning the U-Net

In the second step, we fine-tune the U-Net on the target style images, using the adapted noise distribution for the forward diffusion process, as illustrated in the second row of Figure 2. We follow regular training strategy, except we sample noise from the adapted noise distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$ instead of $\mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d\times d})$. This makes the fine-tuning require orders of magnitude fewer target style images and iterations.

To fine-tune the U-Net, we need image captions. In our experiments, we obtain the captions with BLIP [14, 37]. The fine-tuning strategy is then similar to the training of Stable Diffusion. More precisely, at each iteration, the VAE encodings of the images of the batch are computed using the VAE encoder $\mathbf{z}_0 = \mathcal{E}(\boldsymbol{i})$. Noisy latent tensors $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$ are generated with a random time-step t and a random noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$. The function $\bar{\alpha}_t$ is defined by the noise schedule of the diffusion model. The U-Net is given a noisy latent tensor $\mathbf{z}_t$, the random time-step t, and the image caption. It outputs a predicted noise $\hat{\boldsymbol{\epsilon}}$. An MSE loss between the predicted noise $\hat{\boldsymbol{\epsilon}}$ and the true noise $\boldsymbol{\epsilon}$ is used to optimize the parameters of the U-Net.

3.3. Inference

To generate an image with *Diffusion in Style*, we sample, as explained before, the initial latent tensor $\hat{\mathbf{z}}_T$ from the adapted noise distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$. The fine-tuned U-Net then progressively denoise this latent tensor. Among a few other parameters, one can select the textual prompt and the guidance weight used to generate the image.

The guidance weight is for classifier-free guidance [9], which most large-scale text-to-image models rely on. Especially, at each inference step, two noise predictions $\hat{\boldsymbol{\epsilon}}_{\text{prompt}}$ and $\hat{\boldsymbol{\epsilon}}_{\text{uncond}}$ are made by conditioning the U-Net with and without the textual prompt. The two predictions are combined into one that is more strongly aligned with the textual prompt. More precisely, the guidance weight $w > 1$ amplifies the direction $\boldsymbol{d}_{\text{prompt}} = \hat{\boldsymbol{\epsilon}}_{\text{prompt}} - \hat{\boldsymbol{\epsilon}}_{\text{uncond}}$, leading to $\hat{\boldsymbol{\epsilon}} = \hat{\boldsymbol{\epsilon}}_{\text{uncond}} + w \cdot \boldsymbol{d}_{\text{prompt}}$. This is obtained at the expense of style matching, as we show in Section 4.2.

4. Experiments and results

4.1. Images generated with *Diffusion in Style*

We show images generated with *Diffusion in Style* in Figures 1 and 4 for six different styles. Note that the objects in the generated images do not need to be present in the target style images.

The first style consists of *anime sketches* [43]. The second style consists of images from the *few-shot Pokemon* dataset [16, 35, 37]. The third style consists of 190 comic panels from *48 Famous Americans* (1947) [44], extracted with annotations from the DCM772 dataset [19]. The fourth style consists of 116 paintings tagged as symbolism art from *Salvador Dalí* in the WikiArt dataset [28, 45]. The fifth style consists of 67 *pictograms*. Finally, the sixth style is composed of 3 paintings by Vincent van Gogh, namely *Café Terrace at Night* (1888), *Starry Night Over the Rhône* (1888), and *The Starry Night* (1889). To obtain the six presented *Diffusion in Style* models, we use the number of target style images as follows: 50 images from the training set for the *anime sketch* style, 50 images for the *few-shot Pokemon* style, all 190 images for the *48 Famous Americans* style, all 116 images for the style of *Dalí*, all 67 images for the *pictograms* style, and all 3 images for *Starry Night* style. For the *Starry Night* style, we perform some modifications to our method,

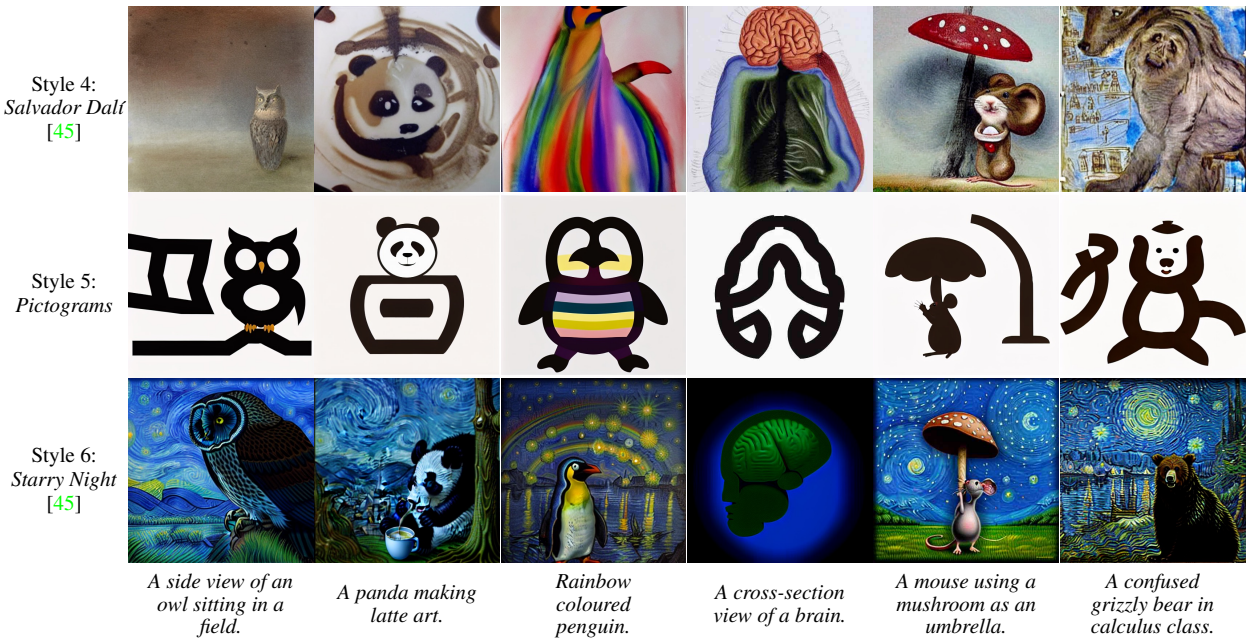

Figure 4. **Style adaptations induced by *Diffusion in Style* for styles 4 to 6:** *Salvador Dalí* (first row), *pictograms* (middle row), and *Starry Night* (bottom row). Each column is generated from the textual prompt indicated at the bottom. Note that the adaptation of Stable Diffusion to the sixth style, *Starry Night*, is performed with *only 3 images* from the target style. Styles 1 to 3 are presented in Figure 1.

as explained in Section 6. For the *anime sketch* style results in Figure 1, we increased the exposure of the generated images as post-processing for better visual presentation.

Figure 9 further show uncurated results, including 3 additional styles. Additional applications, such as in-style image editing, are also presented in Appendix F.

4.2. Trade-off between *style* and *content*

At inference time when generating images with Stable Diffusion, recall that one can choose a guidance weight for classifier-free guidance [9]. This guidance weight is particularly useful with *Diffusion in Style*: it controls how close the generated images are to the target style or to the textual prompt. For low guidance weights, the generated images are closer to the target style but do not match the textual prompt well. For high guidance weights, the images resemble the textual prompt more but the style might suffer. We illustrate this in Figure 5. Similar to the original Stable Diffusion, we also observe that the image quality is degraded for very large guidance weights, as shown in the right-most columns of Figure 5. Note that, the optimal guidance weight may differ depending on the style. In the results we present in this paper, we manually select one guidance weight for each of the six styles. In practice, it is worthwhile to generate each image with different guidance weights, and visually choose the best one.

4.3. Visual comparison with prior art

For the *anime sketch* and *few-shot Pokemon* styles, we qualitatively compare images generated by *Diffusion in Style* with alternative approaches. For the remaining styles, no methods are suitable for comparison to our knowledge. We thus do not discuss them within our qualitative comparisons, but hope the reader still notices the high quality of the style adaptations in the Figures 1, 4, and 9.

Style 1: *Anime sketch*. In Figure 6, we visually compare our *Diffusion in Style* model with the original Stable Diffusion model, with prompt engineering, and with a SoTA *anime-sketches* image translation model [2].

For prompt engineering, we append "*In the style of an anime drawing.*" after each prompt. We provide more variations of prompt engineering in Appendix D.1. For image translation, we apply the Informative Drawings model [2] on top of images generated by the original Stable Diffusion. Note that Informative Drawings was trained for 30 epochs on the full training set of the Anime Sketch dataset [43], that is 14k images, while *Diffusion in Style* is only fine-tuned for 1000 steps on 50 of those 14k images.

It can be seen that only *Diffusion in Style* and Informative Drawings [2] consistently generate images in the desired style. Compared to *Diffusion in Style*, image translation has the drawback that generated images appear post-processed and not directly generated in the desired style.

Style 2: *Few-shot Pokemon*. In Figure 7, we visually compare our *Diffusion in Style* model with the original Stable

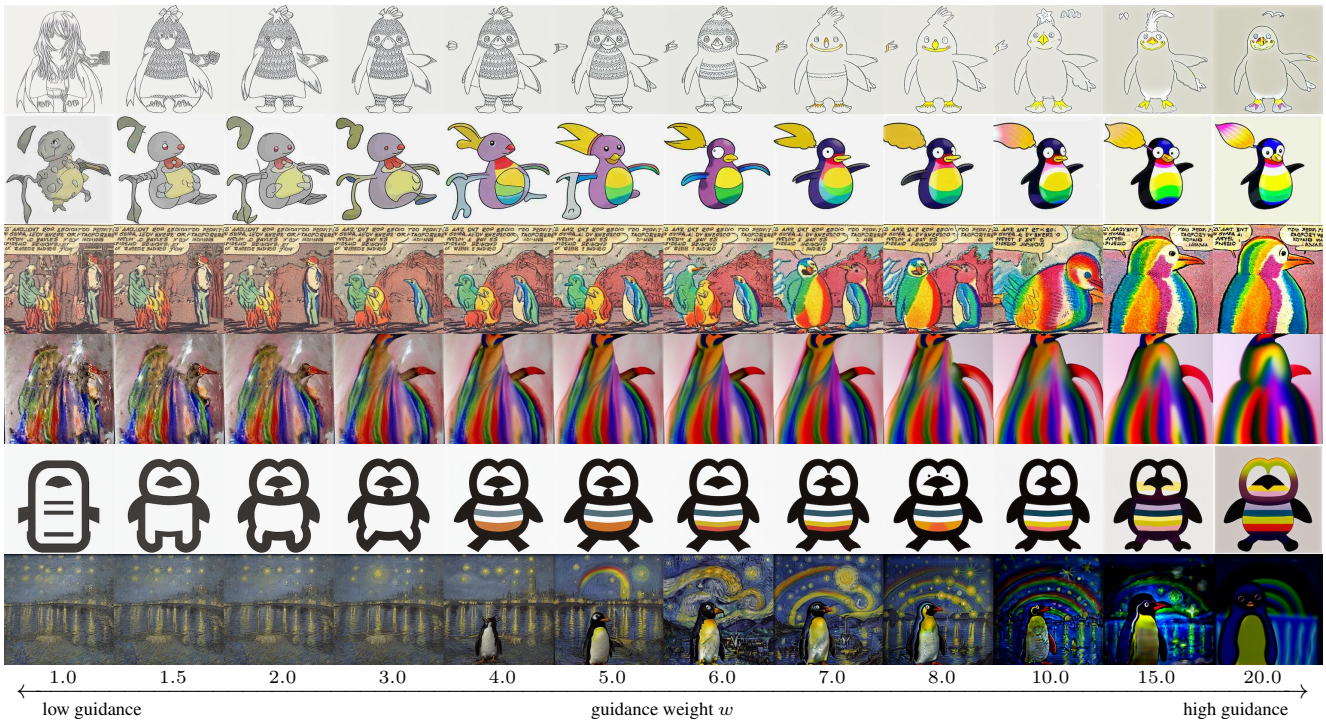

Figure 5. **Trade-off between *style* and *content*.** With the six models (one per row), we generate images from the prompt "*Rainbow coloured penguin.*" with different guidance weights indicated at the bottom of each column. Low guidance weights lead to images that are closer to the target styles, but do not match the prompt well, while high guidance weights lead to images that are closer to the prompt, but do not match the target styles as precisely. A similar observation can be made with any other prompt.

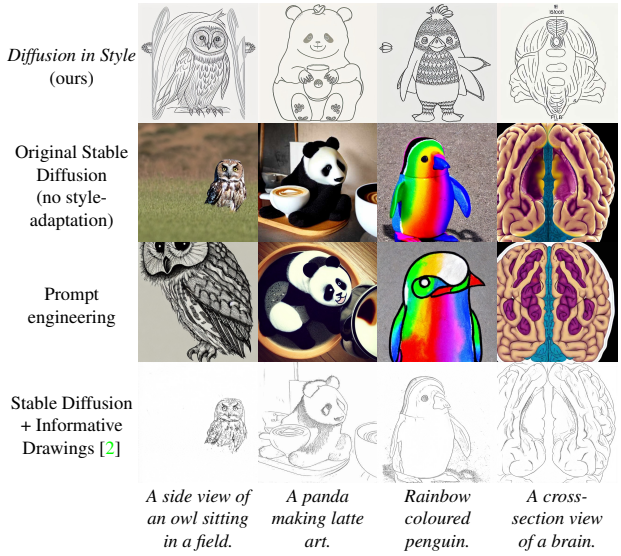

Figure 6. **Qualitative comparison on the *anime sketch* style.** We visually compare our *Diffusion in Style* model (first row), the original Stable Diffusion model (second row), prompt engineering (third row), and Informative Drawings [2] with Stable Diffusion (fourth row). In each column, we generate images from the textual prompt indicated at the bottom. For prompt engineering, we add "*In the style of an anime drawing.*" after each prompt.

Diffusion model, with prompt engineering, with classical fine-tuning [38], and with LoRA-based fine-tuning [39].

For prompt engineering, we append "*In the style of Pokemon, white background.*" after each prompt. For classical fine-tuning, we compare with the Text-To-Pokemon [38] model, which was fine-tuned for 15k steps on all 833 images of the *few-shot Pokemon* dataset [16, 35, 37], while *Diffusion in Style* is only fine-tuned for 1000 steps on 50 of those 833 images. For LoRA-based fine-tuning [10, 42], we compare with Pokemon LoRA [39], which was fine-tuned for 100 epochs on all 833 images.

It can be seen that only *Diffusion in Style* consistently replicates the style of the target style images. Images in the *few-shot Pokemon* dataset all have white backgrounds. This is not always the case with the outputs from prompt engineering, Text-To-Pokemon [38], and Pokemon LoRA [39]. However, it is the case with *Diffusion in Style*.

5. Quantitative evaluation

Following the usual evaluation strategy of text-to-image models [27], we evaluate *Diffusion in Style* models using Pareto curves of CLIP and FID scores along a range of guidance weights. The results are given in Figure 8.

CLIP score. The CLIP score is measured by the ViT-B/32 model of CLIP [23] and represents the alignment be-

Figure 7. **Qualitative comparison on the *few-shot Pokemon* style.** We visually compare our *Diffusion in Style* model (first row), the original Stable Diffusion model (second row), prompt engineering (third row), the Text-To-Pokemon model [38] (fourth row), and the Pokemon LoRA model [39] (fifth row). In each column, we generate images from the textual prompt indicated at the bottom. For prompt engineering, we add "*In the style of Pokemon, white background.*" after each prompt.

tween the textual prompts and the generated images.

Normalized FID score. As done by Wright *et al*. [32] to evaluate style transfer models, we use images from the target style instead of using a set of realistic images to compute the FID score [8] over the features of an Inception model [30] trained on ArtFID dataset [32]. Additionally, to improve interpretability, we normalize the FID score for each guidance weight with the FID scores of the original Stable Diffusion. Please see Appendix B.2 for more insight.

Style 1: *Anime sketch*. For the *anime sketch* style, the normalized FID is computed with the 3546 images from the validation set of the *anime sketch* dataset. In Figure 8, *Diffusion in Style* is compared to the original Stable Diffusion, prompt engineering, and the image translation method, as explained in Section 4.3. *Diffusion in Style* dominates other methods in terms of style-matching. Only for high guidance weights, the image translation method, Informative Drawings [2], surpasses *Diffusion in Style* quantitatively in both prompt-alignment and style-matching.

Style 2: *Few-shot Pokemon*. For the *few-shot Pokemon* style, the normalized FID is computed with the 833 images from the *few-shot Pokemon* dataset. In Figure 8, *Diffusion in Style* is compared to the original Stable Diffusion, prompt engineering, classical fine-tuning, and LoRA-based fine-tuning, as explained in Section 4.3. *Diffusion in Style* dominates all alternatives in terms of style-matching, it also consistently surpasses Text-To-Pokemon [38] in both style-matching and prompt-alignment. A user-study for styles 1 and 2 is presented in the supplementary material, Appendix E.

Styles 3 to 5: *48 Famous Americans*, *Salvador Dalí*, and *pictograms*. For the 3 other styles used in Figure 8, the normalized FID is computed with the set of target style images. We did not find existing fine-tuned models for these styles, hence we only compare *Diffusion in Style* to the original Stable Diffusion, and to prompt engineering. Details and more variations of prompt engineering are given in Appendix D.1. For those 3 styles, *Diffusion in Style* consistently better matches the desired style, as shown by a lower normalized FID.

6. *Diffusion in Style* with very few images

Diffusion in Style works well with 50 to 200 target style images, which is the case for the first five styles presented in this paper, as detailed in Section 4.1. We refer to Appendix C for an ablation study on the required number of target style images.

To make *Diffusion in Style* work with even fewer images, *e.g.*, $n = 3$, we apply the following modifications to our method. We further impose the location $\boldsymbol{\mu}_{\text{style}} \in \mathbb{R}^{4\times64\times64}$ and the covariance diagonal $\boldsymbol{\sigma}^2_{\text{style}} \in \mathbb{R}^{4\times64\times64}$ to be constant across the spatial dimensions of the latent space. We therefore compute their values, not only from n samples, which may provide an unreliable estimate of the style prior, but from $64 \times 64 \times n$ samples, as follows:

$$\begin{aligned} &\forall k_1 \in [1\,..\,4], \forall k_2, k_3 \in [1\,..\,64] \times [1\,..\,64], \\ &\boldsymbol{\mu}_{\text{style},(k_1,k_2,k_3)} = \underset{\substack{i\in I_{\text{style}} \\ j_2\in[1..64] \\ j_3\in[1..64]}}{\text{Mean}} \mathcal{E}_{(k_1,j_2,j_3)}(i) \\ &\boldsymbol{\sigma}_{\text{style},(k_1,k_2,k_3)} = \underset{\substack{i\in I_{\text{style}} \\ j_2\in[1..64] \\ j_3\in[1..64]}}{\text{Std}} \mathcal{E}_{(k_1,j_2,j_3)}(i) \end{aligned} \tag{2}$$

In other words, we enforce the noise distribution to be spatially constant. Furthermore, to avoid fine-tuning the U-Net with only n captions, which could lead to catastrophic forgetting, we generate with BLIP [14], not only 1 but 30 possible captions per image. At fine-tuning time, the 30 captions are used alternatively as conditioning for the U-Net. Finally, instead of fine-tuning for 1000 iterations, we stop at 250 iterations, which amounts to less than 5 minutes of fine-tuning. The *Diffusion in Style* model presented for the sixth style, *Starry Night*, was obtained with these modifications.

7. Ethical considerations and limitations

Diffusion in Style inherits the limitations and biases of Stable Diffusion. Image generation is slower when compared

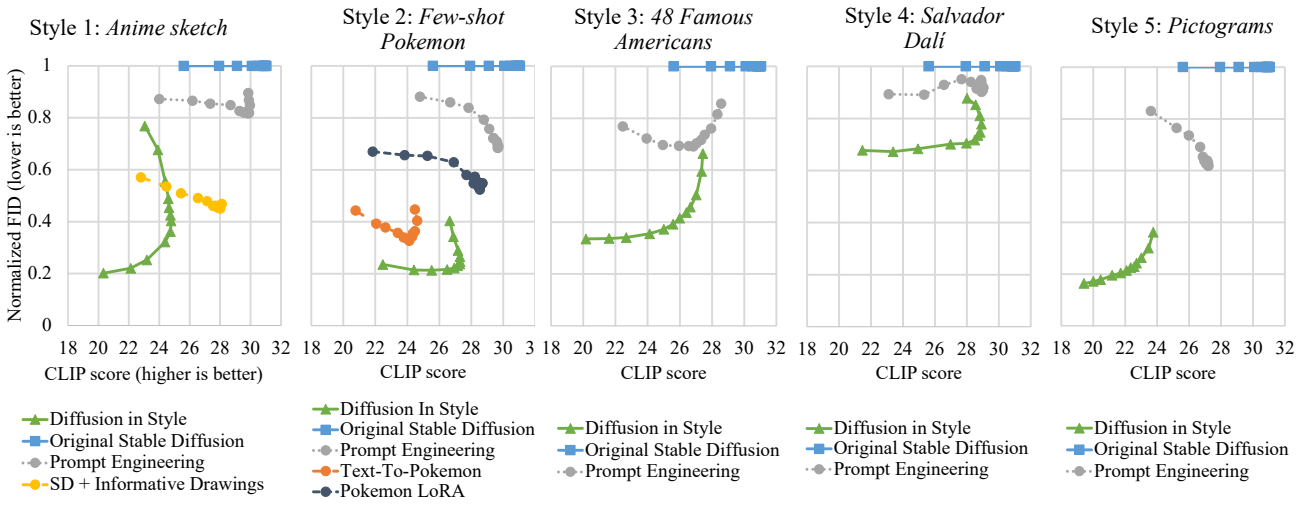

Figure 8. **Curves of FID and CLIP scores along a range of guidance weights**. Evaluation is performed with a range of guidance weights ($\{1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 10.0, 15.0, 20.0\}$), leading to a curve for each model. For each point in the figure, 800 images have been generated with the corresponding model and guidance weight. These 800 images correspond to 4 images for each of the 200 prompts from DrawBench [27]. All 800 images are generated with different initial latent tensors, but the same initial latent tensors are used across the different evaluation points. The left-most point of each curve always corresponds to a guidance weight of 1.0. The overall trend for *Diffusion in Style*, a ⊔-shape, confirms the trade-off between *style* and *content* observed in Section 4.2. Increasing the guidance weight typically improves the prompt-alignment, as measured by the CLIP score, at the cost of degrading the style-matching, as measured by the normalized FID score. A user study, presented in Appendix E, corroborates with the CLIP/FID scores presented here.

to GANs [7]. In our experiments, each image was generated with 50 PLMS sampling steps [17] on a V100 GPU, which amounts to around 4 seconds per image. The models reflect language and social biases that were present in their training data. Works on text-to-image models might lead to potential misuse. In particular, our work should not be used to adapt a model to generate misinformation or any prohibited content.

We acknowledge that it is sometimes difficult to find the exact prompt, the guidance weight, and other parameters, to reach the expected result. A current limitation of our work is the manual selection of the guidance weight. The optimal guidance weight does not only depend on the style but also on the textual prompt. For instance, to generate images in the *pictograms* style in Figure 4, we selected a guidance weight of 15.0, which gives good results for most prompts, but adds colors to the *Rainbow coloured penguin*. This does not match the desired style precisely. As shown in Figure 5, a lower guidance weight, 3.0, would be more adapted for this exact prompt if we want to generate colorless pictograms.

Our *Diffusion in Style* models also inherit from Stable Diffusion another limitation: generating images containing text. In particular, we notice, in Figures 1 and 9, that the text generated by *Diffusion in Style* in speech bubbles for the comics styles is illegible.

Visual examples of such failure cases and limitations are included in Appendix G.

8. Future work

In future work, we aim to understand more theoretically why the original initial latent distribution $\mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d\times d})$ prevents good style adaptation results. Our current method requires fine-tuning the U-Net to adapt it to the new distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$. We would like to find initial latent tensors that are both style-specific and close to the original distribution, which may allow us to skip this fine-tuning step.

9. Conclusion

In this paper, we present *Diffusion in Style*, a new method for style adaptation of Stable Diffusion. We find style-specific initial latent distributions by computing the element-wise mean and variance of the VAE encodings of a small set of target style images. Stable Diffusion is then easily fine-tuned to work with these new style-specific initial latent tensors, producing images in the desired style. Our method, *Diffusion in Style*, is able to generate diverse objects, even if these objects were not present in the target style images, and generate superior results than prior art, as we demonstrate qualitatively and quantitatively. *Diffusion in Style* is a fast and data-efficient method for style adaptation of Stable Diffusion, which opens the door to many practical applications.

Acknowledgements: This work was supported by Innosuisse grant 48552.1 IP-ICT. The authors thank Athanasios Fitsios and the members of the Image and Visual Representation Lab (IVRL) for their advice.

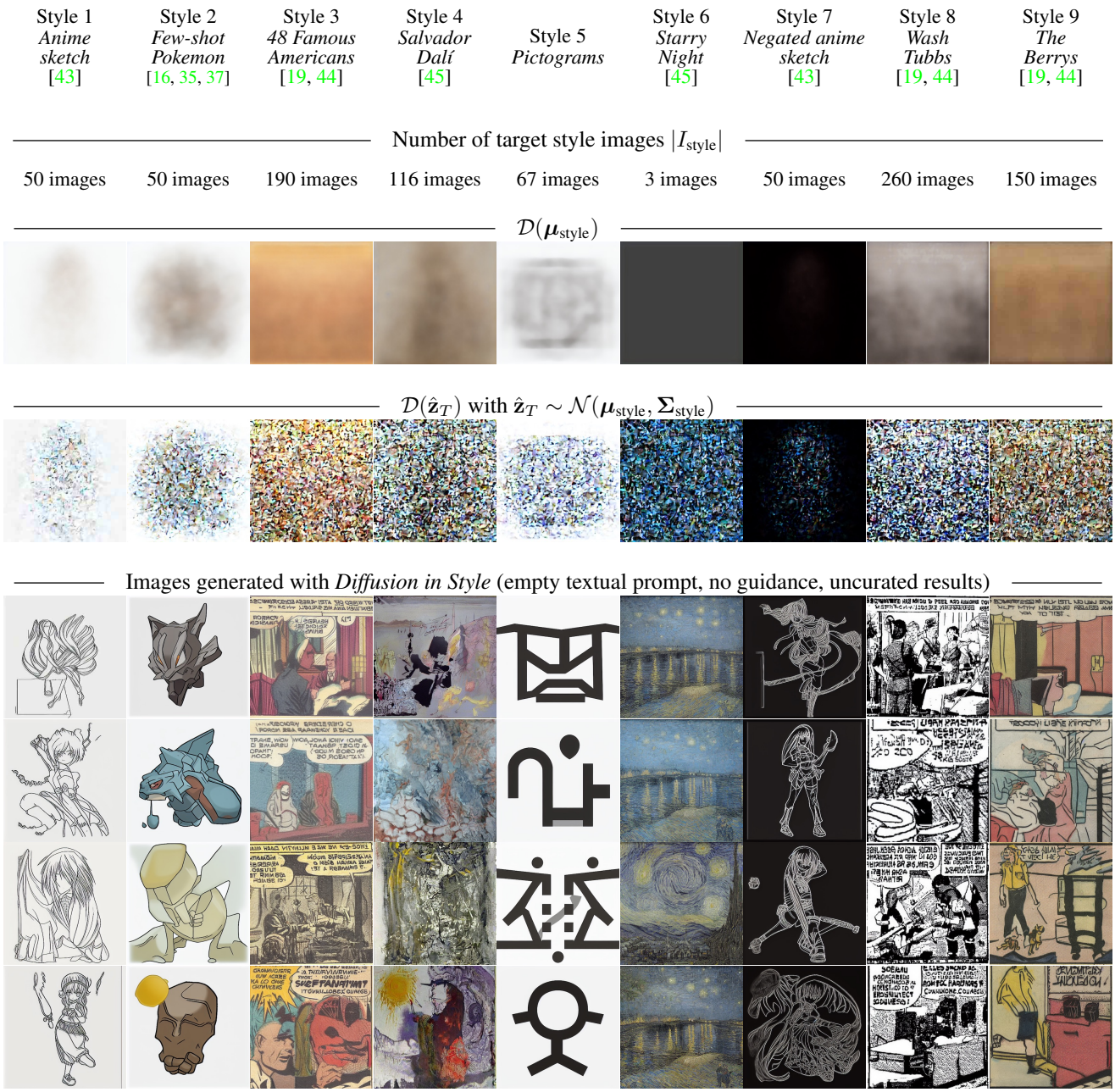

Figure 9. **Number of target style images, locations and samples of the style-adapted initial latent distributions, and uncurated images generated by our proposed *Diffusion in Style*.** For each of the nine different styles, one per column, we show, in the first row the number of target style images. The location $\boldsymbol{\mu}_{\text{style}}$ of the style-specific initial latent distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$ computed in the first stage of our method, is shown in the second row. The third row depicts a random sample $\hat{\mathbf{z}}_T$ from this style-specific initial latent distribution. The remaining rows display uncurated images generated by *Diffusion in Style* from an empty textual prompt.
Diffusion in Style produces images that faithfully match the desired style. By adapting the initial latent distribution to the target style, we ease the process of fine-tuning to the specific style. Generated images exhibit the specific characteristics and aesthetics of the style provided by the user. By not conditioning or guiding the image generation at inference time, we maintain the flexibility and speed of Stable Diffusion. Our approach is a practical and reliable solution for adapting Stable Diffusion to different styles.

References

[1] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise. *arXiv preprint arXiv:2208.09392*, 2022. 3

[2] Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7915–7925, 2022. 2, 5, 6, 7

[3] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G Dimakis, and Peyman Milanfar. Soft Diffusion: Score Matching for General Corruptions. *Transactions on Machine Learning Research*, 2023. 3

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3

[5] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[6] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style. *arXiv preprint arXiv:1508.06576*, 2015. 2

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *Communications of the ACM*, 63(11):139–144, 2014. 8

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[9] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 4, 5

[10] 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. In *International Conference on Learning Representations*, 2021. 2, 6

[11] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 2

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 2

[13] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[14] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 4, 7

[15] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying Neural Style Transfer. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2230–2236, 2017. 2

[16] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards Faster and Stabilized GAN Training for High-fidelity Few-shot Image Synthesis. In *International Conference on Learning Representations*, 2021. 1, 2, 4, 6, 9

[17] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2021. 8

[18] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021. 3

[19] Nhu-Van Nguyen, Christophe Rigaud, and Jean-Christophe Burie. Digital Comics Image Indexing Based on Deep Learning. *Journal of Imaging*, 4(7):89, 2018. 1, 4, 9

[20] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022. 3

[21] Jonas Oppenlaender. Prompt Engineering for Text-Based Generative Art. *arXiv preprint arXiv:2204.13988*, 2022. 2

[22] Zhihong Pan, Xin Zhou, and Hao Tian. Arbitrary style guidance for enhanced diffusion-based text-to-image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4461–4471, 2023. 2, 3

[23] 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. 2, 6

[24] 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

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2

[26] 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. 2, 3

[27] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 6, 8

[28] Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *International Journal for Digital Art History*, 2016. 4

[29] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015. 2

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2, 7

[31] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 1349–1357, 2016. 2

[32] Matthias Wright and Björn Ommer. ArtFID: Quantitative Evaluation of Neural Style Transfer. In *Pattern Recognition: 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September 27–30, 2022, Proceedings*, pages 560–576. Springer, 2022. 2, 7

[33] Pedro Cuenca. Stable Diffusion with Repeatable Seeds — Notebook. https://github.com/pcuenca/diffusers-examples/blob/main/notebooks/stable-diffusion-seeds.ipynb, 2022. 3

[34] Nicholas Guttenberg. Diffusion with offset noise – Cross Labs blog. https://www.crosslabs.org/blog/diffusion-with-offset-noise, 2023. 2

[35] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Few-shot-Pokemon — Hugging Face dataset. https://huggingface.co/datasets/huggan/few-shot-pokemon, 2022. 1, 2, 4, 6, 9

[36] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Stable Diffusion text-to-image fine-tuning — Hugging Face Diffusers documentation. https://huggingface.co/docs/diffusers/v0.13.0/en/training/text2image, 2023. 2

[37] Justin Pinkney. Pokemon BLIP captions — Hugging Face dataset. https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions/, 2022. 1, 2, 4, 6, 9

[38] Justin Pinkney. Text-To-Pokemon — Replicate repository. https://replicate.com/lambdal/text-to-pokemon, 2022. 1, 2, 6, 7

[39] Paul Sayak. Pokemon LoRA — Hugging Face repository. https://huggingface.co/sayakpaul/sd-model-finetuned-lora-t4, 2023. 1, 2, 6, 7

[40] Paula Sayak and Park Chansung. Fine-tuning stable diffusion — Keras documentation. https://keras.io/examples/generative/finetune_stable_diffusion/, 2022. 1

[41] Yehia Serag. Anime-Pencil-Diffusion — Hugging Face repository. https://huggingface.co/yehiaserag/anime-pencil-diffusion, 2022. 3

[42] Ryu Simo. Low-rank Adaptation for Fast Text-to-Image Diffusion Fine-tuning – Github repository, Dec 2022. https://github.com/cloneofsimo/lora. 2, 6

[43] Kim Taebum. Anime Sketch Colorization Pair — Kaggle dataset, 2018. https://www.kaggle.com/datasets/ktaebum/anime-sketch-colorization-pair. 1, 4, 5, 9

[44] Digital Comic Museum. Digital Comic Museum, 2010. https://digitalcomicmuseum.com/. 1, 4, 9

[45] WikiArt. WikiArt, 2010. https://www.wikiart.org/. 4, 5, 9

[46] Cjwbw (username). Waifu Diffusion — Replicate repository. https://replicate.com/cjwbw/waifu-diffusion, 2022. Accessed on September 2022. 1, 2

[47] PromptHero. Openjourney v2 — Hugging Face repository. https://huggingface.co/prompthero/openjourney-v2, 2022. Accessed on February 2023. 1, 2