Diffusion in Style (Supplementary material)

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/

In this document, we provide supplementary material that additionally supports the claims of the manuscript. The outline of this supplementary document is as follows:

- **Appendix A:** Additional details on the styles
- **Appendix B:** Implementation and evaluation details
- **Appendix C:** Ablation study on the number of target style images
- **Appendix D:** Additional results and discussions
- **Appendix E:** User-study
- **Appendix F:** Flexibility of *Diffusion in Style* and applications
- **Appendix G:** Failure cases

For convenience, we reiterate the different steps of our method, described in Section 3:

(1) In the first step of our method, we obtain the style-specific noise distribution by computing the element-wise mean $\boldsymbol{\mu}_{\text{style}}$ and element-wise variance $\boldsymbol{\sigma}^2_{\text{style}}$ of the VAE encodings of the target style images. These VAE encodings are tensors with $d = 4 \times 64 \times 64$ dimensions, hence we compute d mean and variance values, *i.e.*, $\boldsymbol{\mu}_{\text{style}} \in \mathbb{R}^d$ and $\boldsymbol{\sigma}^2_{\text{style}} \in \mathbb{R}^d$. Different noises $\boldsymbol{\epsilon} \in \mathbb{R}^d$ can be sampled from a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$ with the computed mean and variance, assuming diagonal covariance $\boldsymbol{\Sigma}_{\text{style}} = \text{diag}(\boldsymbol{\sigma}^2_{\text{style}}) \in \mathbb{R}^{d \times d}$.

(2) In a second step, we fine-tune the U-Net following the regular fine-tuning strategy, but we sample noise $\boldsymbol{\epsilon}$ from the style-specific distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$ instead of $\mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d \times d})$.

(3) At inference time, we sample initial latent tensors $\hat{\mathbf{z}}_T$ from the style-adapted distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$ instead of $\mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d \times d})$, and we use the fine-tuned U-Net to progressively denoise it and obtain the generated image $\mathcal{D}(\hat{\mathbf{z}}_0)$

A. Additional details on the presented styles

Original Stable Diffusion: Stable Diffusion v1.5. [9] was trained on LAION-2B-en [11] for 237k iterations, then on LAION-high-res for 194k iterations, then on LAION-improved-aesthetics for 515k steps and finally on LAION-aesthetics-v2-5+ for 595k steps.

Style 1, *anime sketch*: The first style refers to the style of the *anime sketch* dataset[1]. We fine-tune Stable Diffusion on this style with *Diffusion in Style* using the 50 first images from the training set of this dataset.

Style 2, *few-shot Pokemon*: The second style refers to the style of the *few-shot Pokemon* dataset[2]. We fine-tune Stable Diffusion on this style with *Diffusion in Style* using the 50 first images of this dataset.

Style 3, *48 Famous Americans* (1947): The third style refers to the style of the comic panels from the *48 Famous Americans* comic book[3], from the Digital Comic Museum. We extract these panels using annotations from the DCM772 dataset [5], giving us 190 images. We fine-tune Stable Diffusion on this style with *Diffusion in Style* using the 190 images.

Style 4, *Salvador Dalí*: The fourth style refers to the style of the 116 images tagged as Symbolism art by Salvador Dalí in the WikiArt dataset[4]. We fine-tune Stable Diffusion on this style with *Diffusion in Style* using these 116 images.

Style 5, *pictograms*: The fifth style contains 67 pictograms. Pictograms have a white background and large black strokes. We fine-tune Stable Diffusion on this style with *Diffusion in Style* using the 67 images.

Style 6, *Starry Night*: The sixth style refers to the style of three paintings by Vincent Van Gogh, namely *Café Terrace at Night* (1888), *Starry Night Over the Rhône* (1888), and *The Starry Night* (1889), downloaded from Wikimedia[5]. We

[1] https://www.kaggle.com/datasets/ktaebum/anime-sketch-colorization-pair
[2] https://huggingface.co/datasets/huggan/few-shot-pokemon, https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions
[3] https://digitalcomicmuseum.com/index.php?dlid=24742
[4] https://www.wikiart.org/
[5] https://commons.wikimedia.org/

fine-tune Stable Diffusion on this style with *Diffusion in Style* using the 3 images, with the modifications we mention in Section 6.

Style 7, *negated anime sketch*: The target style images of the seventh style were obtained by negating the 50 target style images of the first style *anime sketch*. We used the function `PIL.ImageOps.invert` from the Pillow library. We fine-tune Stable Diffusion on this style with *Diffusion in Style* using the 50 negated images.

Style 8, *Wash Tubbs* (September to December 1944): The eight style refers to the style of the comic panels from the *Wash Tubbs* comic book[6], from the Digital Comic Museum. We use the batch of comics for the dates 1944-09-04 to 1944-12-08 and extract the panels using annotations from the DCM772 dataset [5], giving us 260 images. We fine-tune Stable Diffusion on this style with *Diffusion in Style* using the 260 images.

Style 9, *The Berrys 1* (May 1956): The nineth style refers to the style of the comic panels from *the Berrys 1* comic book[7], from the Digital Comic Museum. We use annotations from the DCM772 dataset [5] to extract the panels, giving us 150 images. We fine-tune Stable Diffusion on this style with *Diffusion in Style* using the 150 images.

B. Implementation and evaluation details

B.1. Training and inference details

We implemented *Diffusion in Style* on top of the Diffusers library [15] with the weights of Stable Diffusion v1.5 from Hugging Face Hub [13]. In the following paragraphs, we provide additional implementation details for each step.

Step 1: Adapting the noise distribution We now provide additional details on the first step of *Diffusion in Style*, explained in Section 3.1.

In Equation 1, to compute the mean $\boldsymbol{\mu}_{\text{style},k}$ and variance $\boldsymbol{\sigma}_{\text{style},k}$ of each element $\mathcal{E}_k(i)$ of the VAE encodings, we use the naive estimators, *i.e.*, empirical mean and biased sample variance:

$$\begin{aligned} \boldsymbol{\mu}_{\text{style},k} &= \frac{\sum_{i \in I_{\text{style}}} \mathcal{E}_k(i)}{|I_{\text{style}}|} \qquad \forall k \in [1 \,..\, d] \\ \boldsymbol{\sigma}^2_{\text{style},k} &= \frac{\sum_{i \in I_{\text{style}}} \left(\mathcal{E}_k(i) - \boldsymbol{\mu}_{\text{style},k}\right)^2}{|I_{\text{style}}|} \qquad \forall k \in [1 \,..\, d] \end{aligned} \tag{3}$$

Here, I_{style} is the set of target style images, $|I_{\text{style}}|$ is the number of target style images, $\mathcal{E}$ is the VAE encoder, and $\mathcal{E}_k(i)$ is the k-th element of the VAE encoding $\mathcal{E}(i)$ of image i. Similarly, we also use the naive estimators of mean and variance for Equation 2. In practice, it would also be possible to use other estimators of the variance. In our preliminary experiments, we found no significant difference using Bessel's correction for the estimation of the variance, *i.e.*, dividing by $(|I_{\text{style}}| - 1)$ instead of $|I_{\text{style}}|$.

In Figure 10, we visualize 4 random samples from the style-adapted noise distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$ for each style. These visualizations in the image space are obtained using the VAE decoder $\mathcal{D}$. Intuitively, it can be understood that the adapted initial latent distribution $\mathcal{N}(\boldsymbol{\mu}_{\text{style}}, \boldsymbol{\Sigma}_{\text{style}})$ better represents the style, while the style-agnostic distribution $\mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d\times d})$ better represents the original training images of Stable Diffusion.

Step 2: Fine-tuning the U-Net on the style-specific noise distribution We now provide additional details on the second step of *Diffusion in Style*, explained in Section 3.2.

Similar to how Stable Diffusion was trained, we randomly drop the captions 10% of the time while fine-tuning the U-Net, to improve image generation with classifier-free guidance [3], which we explain again in the next paragraph. At the end of the training, we save an exponential moving average of the parameters of the U-Net.

All results presented in the main paper, except for the sixth style, use the following set of hyperparameters, which we found to yield good results: a learning rate of 10^{-5}, a center cropping with a resolution of 512×512, a gradient clipping of 1, 1000 training iterations, a batch size of 4, and a decay factor of 5% for the final exponential moving average. For the sixth style, *Starry Night*, we fine-tune for 250 steps instead of 1000, and we save the exponential moving average of the U-Net with a decay factor of 50%.

Inference: generating images in the desired style We now provide additional details on inference with *Diffusion in Style*, explained in Section 3.3.

As we mentioned in Section 4.2., the guidance weight is particularly useful in the case of *Diffusion in Style*, as it controls how close the generated images are to the target style or to the textual prompt. This guidance weight is used for classifier-free guidance [3].

In practice, the guidance weight $w \geq 1$ is used to combine the noise predictions $\hat{\boldsymbol{\epsilon}}_{\text{uncond}}$ and $\hat{\boldsymbol{\epsilon}}_{\text{prompt}}$ of the U-Net conditioned with and without the textual prompt with the following equation:

$$\begin{aligned} \hat{\boldsymbol{\epsilon}} &= \hat{\boldsymbol{\epsilon}}_{\text{uncond}} + w \cdot \boldsymbol{d}_{\text{prompt}} \\ \text{with } \boldsymbol{d}_{\text{prompt}} &= \hat{\boldsymbol{\epsilon}}_{\text{prompt}} - \hat{\boldsymbol{\epsilon}}_{\text{uncond}} \end{aligned} \tag{4}$$

A guidance weight $w = 1$ corresponds to taking denoising steps without classifier-free guidance, that corresponds to $\hat{\boldsymbol{\epsilon}} = \hat{\boldsymbol{\epsilon}}_{\text{prompt}}$. A guidance weight $w > 1$ moves it in the direction $\boldsymbol{d}_{\text{prompt}}$, aligning the generated images more with the textual prompt.

[6] https://digitalcomicmuseum.com/preview/index.php?did=10788

[7] https://digitalcomicmuseum.com/index.php?dlid=17781

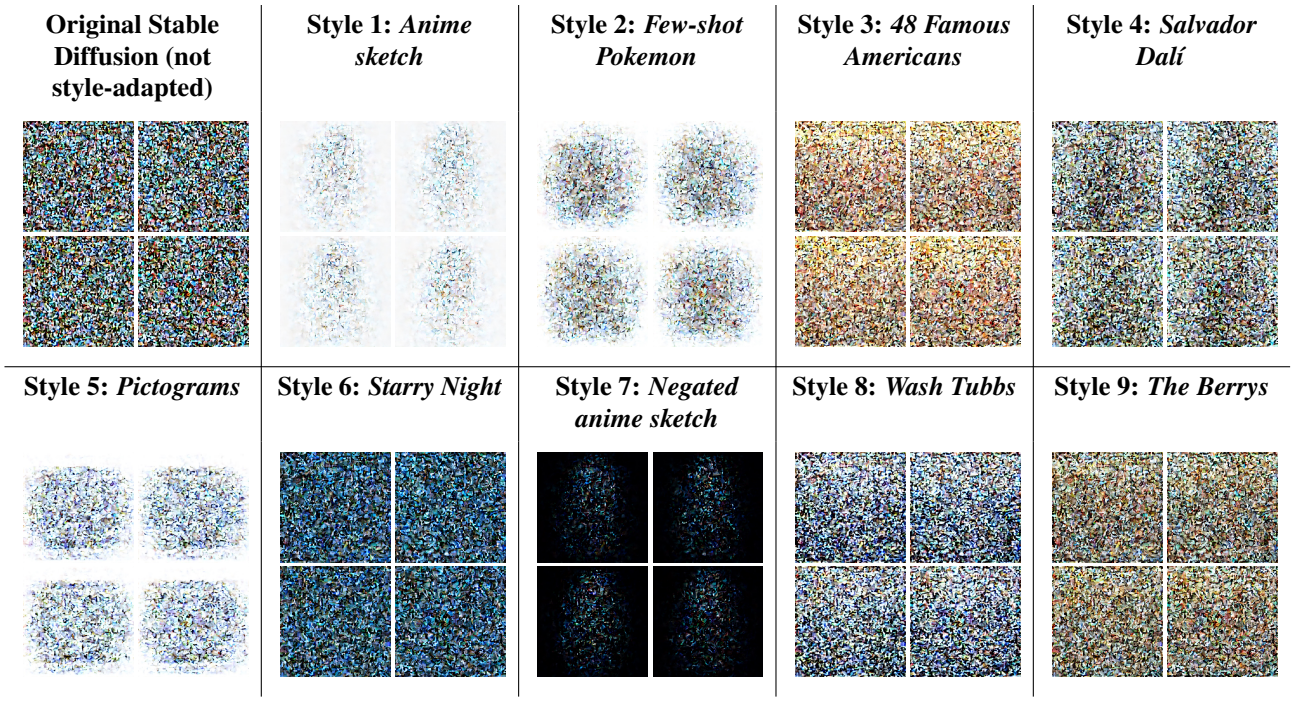


Figure 10. **Visual representation of random samples from the noise distribution, *i.e.*, of possible initial latent tensors.** We visualize $\mathcal{D}(\hat{\mathbf{z}}_T)$ with $\hat{\mathbf{z}}_T \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For the original Stable Diffusion, $\boldsymbol{\mu} = \mathbf{0}_d$ and $\boldsymbol{\Sigma} = \mathbf{I}_{d\times d}$; for the 9 styles, $\boldsymbol{\mu} = \boldsymbol{\mu}_{\text{style}}$ and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\text{style}}$. We show 4 random samples for each style.

The images presented in Figures 1 and 4 were generated with the following guidance weights: $w = 4$ for the *anime sketch* style (style 1), $w = 5$ for the *Salvador Dalí* style (style 4), $w = 8$ for the *few-shot Pokemon*, *48 Famous Americans*, and *The Berrys* styles (styles 2, 3, and 6), and $w = 15$ for the *pictograms* style (style 5). These guidance weights were chosen through trial-and-error, looking qualitatively at the generated images. While we use the same guidance weight to generate all images of a specific style, results can be further improved in practice by choosing a different guidance weight for each textual prompts.

A variation of classifier-free guidance, where the unconditional prediction $\hat{\epsilon}_{\text{uncond}}$ is replaced by a "negative" prediction $\hat{\epsilon}_{\text{negative}}$, is commonly used and also works with *Diffusion in Style*, as we show in Appendix F.3.

B.2. Evaluation details

Our quantitative evaluation presented in Section 5 relies on the 200 prompts from the DrawBench benchmark [10]. For each model and guidance weight, 800 images were generated, 4 for each prompt. 800 different seeds, to sample the initial latent tensor from the noise distribution, were used to generate the 800 images. To make the comparison fair, the same 4 seeds per prompt were used to generate 4 images per prompt with the different models and guidance weights.

Average CLIP score for image-text alignment The alignment between the input prompt and the generated image is measured by the CLIP score with the ViT-B/32 model of CLIP [8]. One CLIP score is computed for each of the 800 images and the average of the 800 CLIP scores is reported as CLIP score for the model and guidance weight pair. The same procedure is repeated for each model and guidance weight.

Normalized FID score for style-matching As we mention in Section 5, we use FID with an Inception model [12] trained on ArtFID dataset [14] to evaluate *style-matching*. We normalize all FID scores for each guidance weight with the FID scores of the original Stable Diffusion. We show in Figure 11 the quantitative evaluation without this normalization.

From Figure 11, we can notice that the non-normalized FID does not only measure *style-matching*. Indeed, the FID score of the original Stable Diffusion is not constant, while we expect the *style-matching* metric to be constantly *bad* for images generated with Stable Diffusion ignoring the style. We further notice that the FID score tends to decrease when the guidance weight increases, across different models and styles. We then hypothesize that normalizing the FID scores of a model with the FID scores of the original Stable Diffusion cancels out the shared tendency to decrease with the guidance weight, leading to a more appropriate metric

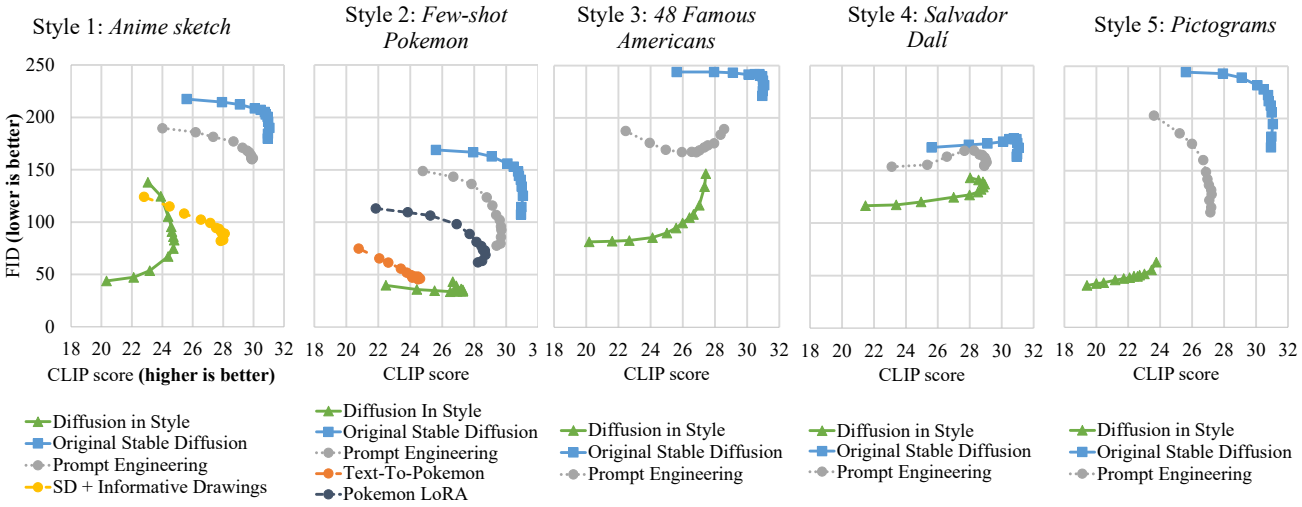


Figure 11. **Quantitative evaluation with the *non-normalized* FID score.** Curves of FID and CLIP scores along a range of guidance weights. Evaluation is performed with a range of guidance weights ($\{1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 10.0, 15.0, 20.0\}$), leading to a curve for each model. 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 [10]. 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.

for style-matching.

Overall, normalizing the FID scores eases understanding and interpretation of the FID score as a potential, yet likely imperfect, *style-matching* score. We thus decided to report the *normalized* FID scores in Figure 8.

C. Ablation study on the number of target style images

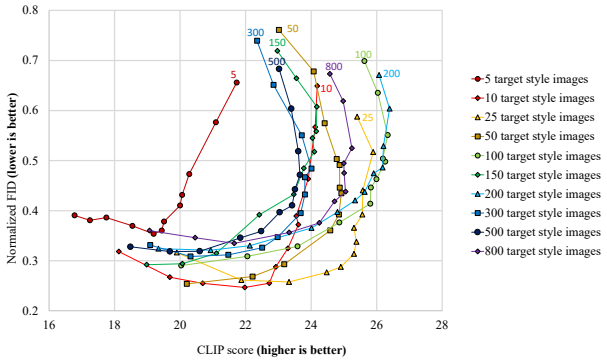


Figure 12. **Quantitative evaluation of *Diffusion in Style* with varying number of target style images**. We follow the same evaluation strategy as we mention in Section 5 and Figure 8, with a range of numbers of target style images, from 5 to 800. For each point of each curve, 200 images were generated, from the 200 prompts from DrawBench [10]. Note again the ⅃-shape of the curves, indicating the trade-off between content and style, which we analyze in Section 4.2.

For the *anime sketch* style, we evaluate our method *Diffusion in Style*, training it with a varying number of target style images $|I_{\text{style}}| \in \{5, 10, 25, 50, 100, 150, 200, 300, 500, 800\}$.

We follow the same evaluation strategy as before, CLIP versus normalized FID for a range of guidance weights, except that we generate only one image per prompt from DrawBench [10], instead of 4, to speed up evaluation. The results are provided in Figure 12.

Except for the training with 5 target style images, which performs poorly both in terms of FID scores and CLIP scores, the overall influence of the number of target style images is not easy to analyze. All other models appear to perform relatively similarly. The poor performance of the model trained with only 5 images emphasizes the need for the modifications we present in Section 6.

Figure 13. **Lowest FID score achieved**, as a function of the number of target style images, in our ablation study. For each number of target style images, we look at the minimum value of the normalized FID score in Figure 12.

Figure 14. **Highest CLIP score achieved**, as a function of the number of target style images, in our ablation study. For each number of target style images, we look at the maximum value of the CLIP score in Figure 12.

As we show in Figures 13 and 14, the three models that obtain the lowest normalized FID scores are trained with 10, 25, and 50 images; and the two models obtaining the highest CLIP scores are trained with 100 and 200 images, while, surprisingly, the one trained with 150 target style images only reaches lower maximum CLIP score. It then seems reasonable to recommend using 50 to 200 target style images, as a low number of target style images may give unreliable estimates of the style, and a high number of target style images might be inconvenient.

We also want to point out that the optimal number of target style images might also depend on the target style, although that is something we have not studied.

D. Additional experiments and discussions

D.1. Prompt engineering variations

We provide more variations of prompt engineering to support the claim that the desired styles cannot be obtained by way of prompt engineering.

We compare visually, in Figures 24, 25, and 26, images generated with the original Stable Diffusion, with five different prompt engineering templates, and with *Diffusion in Style*. None of the three styles can be precisely obtained with prompt engineering, while it is the case for *Diffusion in Style*.

In each of the Figures 24, 25, and 26, the first prompt engineering template, that is the second row of each picture, is the one used for the quantitative evaluation in Section 5 and Figure 8. Especially, for our quantitative evaluation, we use the following prompt engineering template: "`[prompt]` *In the style of an anime drawing.*" (style 1), "`[prompt]` *In the style of Pokemon, white background*" (style 2), "`[prompt]` *In the style of comics from 1940s.*" (style 3), "`[prompt]` *In the painting style of Salvador Dalí.*" (style 4), and "*A minimalist pictogram of* "`[prompt]`"" (style 5).

D.2. Style gradient guidance

As mentioned in Section 2.2, 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 [2] or a CLIP model [6]. The gradient of the score predicted by the auxiliary model is incorporated into the noise predicted by the diffusion model [2] at each denoising step.

Pan *et al*. [7] 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 [6].

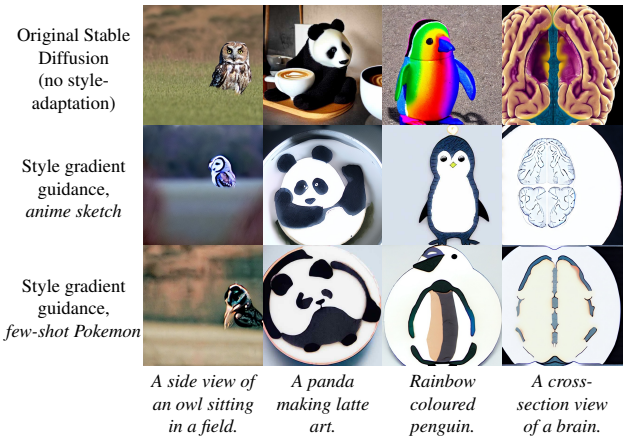


Figure 15. **Qualitative results for style gradient guidance**, on the *anime sketch* style (second row) and the *few-shot Pokemon* style, compared with the original Stable Diffusion model (first row). In each column, we generate images from the textual prompt indicated at the bottom. Results are obtained with a classifier-free guidance weight of 8.0, an overall style-gradient guidance scale of 200, and a layer weight of 1.0 for the 4 VGG layers.

We reimplemented this idea on the Stable Diffusion model. Our implementation is based on CLIP-guided Stable Diffusion[8], and we replaced the CLIP-score function (in `cond_fn`) by the style loss from AdaIN[9]. Through trial and error, we tried various overall style guidance scales, with and without increasing the scale at lower noise level (guide 2 in [7]), and various weights for each of the 4 considered VGG layers. However, we were unable to obtain results on Stable Diffusion comparable to the ones reported from GLIDE. We hypothesize that the initial latent tensors may have a stronger influence on the style of the generated images with Stable Diffusion than with GLIDE.

We show in Figure 15 some images we were able to obtain

[8] https://github.com/huggingface/diffusers/blob/main/examples/community/clip_guided_stable_diffusion.py

[9] https://github.com/naoto0804/pytorch-AdaIN/blob/master/net.py#L130C9-L130C24

with this strategy. As can be seen in this figure, while style gradient guidance indeed influences the generated images toward the desired style, it is not sufficient to match the style carefully with Stable Diffusion.

D.3. Image-to-image

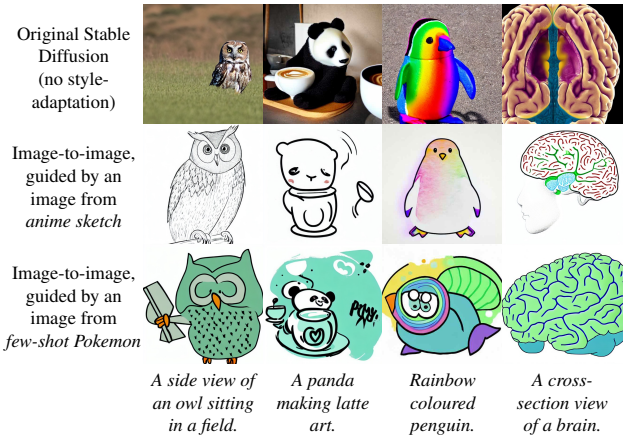


Figure 16. **Qualitative results for image-to-image**, on the *anime sketch* style (second row) and the *few-shot Pokemon* style, compared with the original Stable Diffusion model (first row). In each column, we generate images from the textual prompt indicated at the bottom. Results are obtained with a classifier-free guidance weight of 8.0, and a denoising strength of 0.8.

Meng *et al*. [4] propose a technique called SDEdit, implemented in the Diffusers Python library as an "Image-to-image" pipeline. Instead of starting from a random initial latent tensor and denoising through all $T \rightarrow 0$ timesteps of the diffusion model, image-to-image starts denoising from a noisy version of an input guide image, and only uses the lowest timesteps $t_0 \rightarrow 0$ for denoising. The ratio t_0/T is also known as "denoising strength".

We show in Figure 16 some images we were able to obtain with this strategy. Interestingly, it appears that the generated images are able to match the low-frequency characteristics of the input guide image, such as white background or green object, even when using very high denoising strength. Image-to-image reaches reasonable results by starting denoising from a noisy version of the input guide image instead of random noise. This observation seems to validate that the (style-agnostic) noise distribution used in Stable Diffusion to sample the initial latent is not adapted for generating stylized images.

However, the results from Figure 16 also show that details (high-frequency characteristics) of the generated images do not match the desired style perfectly. This observation highlights the need for fine-tuning the U-Net. For each style, we then fine-tuned the model on the target style images (with style-agnostic noise, as in the original training of Stable Diffusion), and were able to obtain good results by combining the fine-tuned model with image-to-image. We show such results in Figure 17.

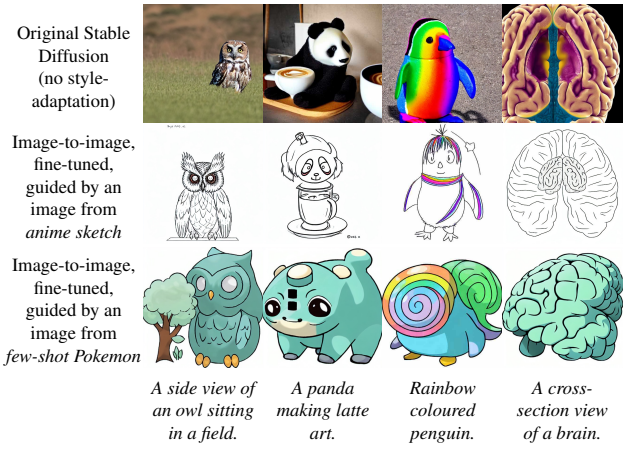


Figure 17. **Qualitative results for image-to-image after fine-tuning**, on the *anime sketch* style (second row) and the *few-shot Pokemon* style, compared with the original Stable Diffusion model (first row). In each column, we generate images from the textual prompt indicated at the bottom. Results are obtained with a classifier-free guidance weight of 8.0, and a denoising strength of 0.8.

The obtained results match the desired style correctly. Especially, it appears that the low-frequency characteristics of the style are obtained from the input guide image and its high-frequency characteristics are obtained through the fine-tuned model.

These good preliminary results highlight the need to start denoising from a style-specific latent tensor: here, a noisy version of a target style image; in *Diffusion in Style* with an initial latent tensor sampled from a style-specific noise distribution. It also highlights the need for fine-tuning to learn the high-frequency characteristics of the target style images.

Note that combining fine-tuning and image-to-image as above has some drawbacks. First, it is necessary to share one or several target style images, to give as input to the image-to-image pipeline. This may be undesirable. Secondly, the low-frequency characteristics of the generated images, for instance the main colors, appear copied from those of the input guide image. Especially, if the same input guide image is reused, as in Figures 16 and 17, generated images may always have the same color. This is the case here: notice the light turquoise color (from the input guide image) on the generated images for the *few-shot Pokemon* style in Figures 16 and 17. Furthermore, in the event of outliers in the set target style images, the low-frequency characteristics of the generated images might not match the desired style.

Our proposed *Diffusion in Style* overcomes these draw-

Approach/Model	Style	Content	Style	Content
Diffusion in Style (**ours**)	73±4%	58±4%	73±4%	38±4%
Prompt engineering	13±3%	67±4%	20±3%	53±4%
Stable Diffusion + Informative Drawings [1]	-	-	80±4%	48±4%
Text-to-Pokemon [16]	68±4%	30±3%	-	-
Pokemon LoRA [17]	39±4%	60±4%	-	-
Image-to-image [4] (Figure 16)	83±3%	39±4%	68±4%	56±4%
Fine-tuning + Image-to-image [4] (Figure 17)	52±4%	44±4%	35±4%	56±4%
Style gradient guidance (Figure 15)	22±3%	52±4%	24±4%	49±4%
	Few-shot Pokemon style		*Anime sketch* style	

Table 1. **User-study results**. Percentages indicate how likely an image generated by the method is preferred over an image generated (with the same prompt) by any of the other methods, selected randomly. Users were asked to "select which of the two generated images better reproduces the reference style." (resp. "[...] has a content that is best described by the reference text."). We obtained a total of 1662 valid image pair comparisons, from 43 users (excluding rejected) on Amazon Mechanical Turk. Note that the dashes in the column indicate that the models are not applicable to that style.

backs by computing a style-specific distribution from the set of target style images. Instead of having to start denoising from a noisy version of an existing image, we simply start denoising from a sample of this computed style-specific distribution.

By not using the image-to-image pipeline, but the original diffusion pipeline (except for the sampling of the initial latent tensors), our *Diffusion in Style* models stay as flexible as the original Stable Diffusion. For instance, it is possible to use image-to-image on top of a *Diffusion in Style* model to perform in-style image variation or in-style local image editing, as we show in Appendix F.

E. User study

In Figures 8, 11, and 12, we presented CLIP/FID scores as it is customary for related works [9, 10]. In Table 1, we show our user-study results, including image-to-image and style gradient guidance described in Appendix D. The results of the user study seem to agree with the reported CLIP/FID scores. We provide details of the user study below.

Considered methods and images We performed four questionnaires for the user study, corresponding to the four columns in Table 1. Eight different methods or models, corresponding to the eight rows, were compared in total, 6 for the *anime sketch* style and 7 for the *few-shot Pokemon* style. The users were not aware of the methods, and were only told these were "generated images".

To simplify the comprehension, we focused on 5 textual prompts whose content is easy to understand, namely "*A side view of an owl sitting on a field*", "*A panda making latte art*", "*Rainbow coloured penguin*", "*A cross-section view of a brain*", and "*A confused grizzly bear in calculus class*". Generated images were not curated.

Each questionnaire was composed of the instructions, followed by 3 examples, and 35 or 47 questions. Each question, including the 3 examples, consisted of a pair of images (left and right). The user was asked to select the best-matching image, either in terms of style or text alignment, as explained below. The user could answer "Left", "Right", or "Cannot Determine / Both Equally".

The two questionnaires (content and style) for the *anime sketch* style had 35 pairs of images to compare, including 30 effective comparisons, and 5 comparisons used to assess automatically the quality of the answers and reject random answers. The 30 comparisons correspond to the 6 methods, each compared with the 5 other methods twice. The two questionnaires (content and style) for the *few-shot Pokemon* style had 47 pairs of images to compare, including 42 effective comparisons, and 5 comparisons used to assess automatically the quality of the answers and reject random answers. The 42 comparisons correspond to the 7 methods, each compared with the 6 other methods twice.

Style questionnaires On two of these four questionnaires, the user was asked to compare pairs of images, and say for each pair which of the two images better matches the desired style. The five same reference style images were repeated above each question. The instructions were as follows: "Read the task and examples carefully, inspect the reference style images and then inspect the generated images. There are 35 questions in this HIT. For each row, select which of the two generated images (Left or Right) better reproduces the reference style. If you're not sure, select "Cannot Determine / Both Equally". The content/objects of generated images might be different that the content/objects of the reference style images. These should not affect your answer. Do not take the content into account and focus on the style of the image only." Below each pair of images, the user was asked "Which of these two generated images (Left or Right) better matches the reference style above?". Three screenshots from the user-study website are shown in Figure 18.

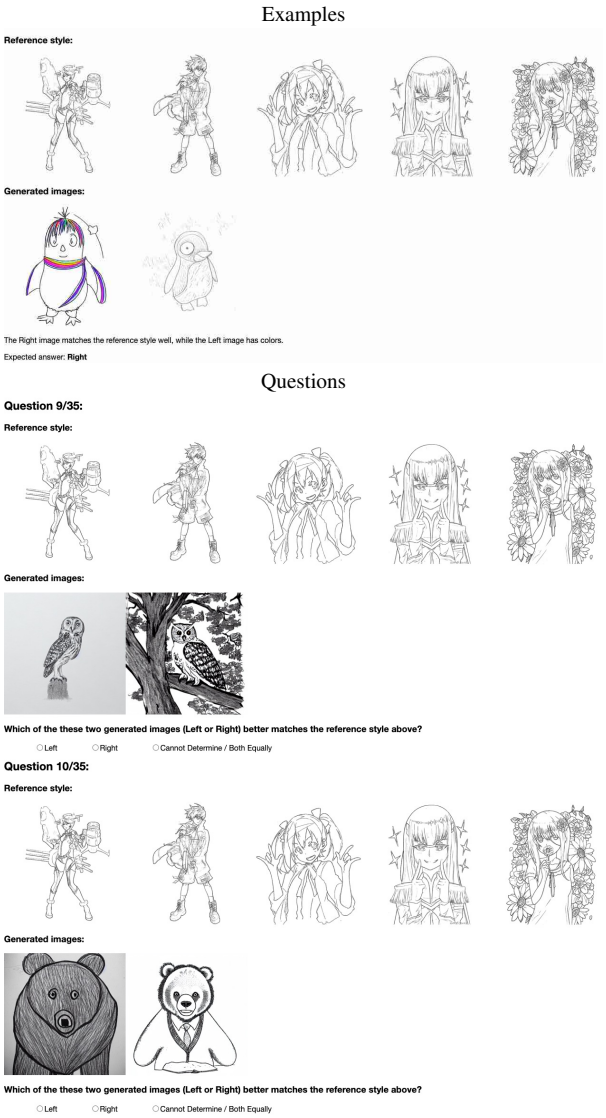


Figure 18. **A style questionnaire for the *anime sketch* style.** A screenshot of one of the 3 examples (top), and screenshots of 2 of the 35 questions.

Content questionnaires On two of these four questionnaires, the user was asked to compare pairs of images, and say for each pair which of the two images better matches the input textual prompt. The textual prompt used to generate the images was indicated above each question. The instructions were as follows: "Read the task and examples carefully, inspect the reference text and then inspect the generated images. There are 35 questions in this HIT. For each row, select which of the two generated images (Left or Right) has a content that is best described by the reference text. If you're not sure, select "Cannot Determine / Both Equally". The style of the image should not affect your answer. Focus on the content of the image only." Below each pair of images, the user was asked "Which of these two generated images

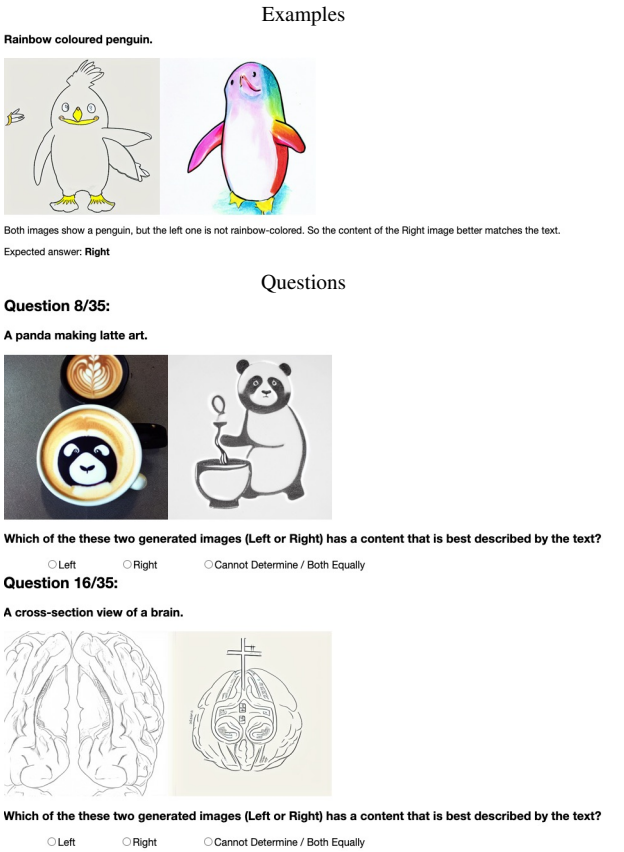


Figure 19. **A content questionnaire for the *anime sketch* style.** A screenshot of one of the 3 examples (top), and screenshots of 2 of the 35 questions.

(Left or Right) has a content that is best described by the text?". Three screenshots from the user-study website are shown in Figure 19.

F. Flexibility of *Diffusion in Style*

Because we only change the distribution of initial latent tensors and fine-tune the U-Net, *Diffusion in Style* is as flexible as Stable Diffusion. In particular, *Diffusion in Style* models can also be used for in-style image editing, in-style local image editing (also known as legacy inpainting), negative prompting, *etc*.

F.1. Image editing *in Style*

Following the work of Meng *et al*. [4], Stable Diffusion can be used to edit images with a textual prompt without any additional training.

In short, instead of progressively denoising an initial latent tensor sampled from a Gaussian distribution, we denoise a noisy version of an image given as input. This image modification process is implemented for Stable Diffusion as an "Image-to-Image pipeline" in the Diffusers library [15]. A similar process can be achieved with *Diffusion in Style*,

using noise sampled from the style-adapted distribution and denoising with the fine-tuned U-Net.

We show such examples in Figure 27, where we compare them with the original Image-to-Image pipeline. *Diffusion in Style* allows performing image editing while staying inside the style, which is not the case with the original Stable Diffusion.

F.2. Local image editing *in Style*

Diffusion models can be used for local image editing, without any additional training. This pipeline is implemented as a "legacy inpaint image pipeline" in the Diffusers library [15].

To perform local image editing with *Diffusion in Style*, we again sample noise from the style-specific distribution instead of the style-agnostic one, and we use the fine-tuned U-Net to perform denoising steps.

In Figure 28, we compare local image editing obtained with and without *Diffusion in Style*. Given an image from the target style, it is more convenient to use *Diffusion in Style* to perform local image editing. Not only the local image editing is successful more often, but local image editing is also done *in Style*, while the original "legacy inpainting" is not able to stay faithful to the expected style.

F.3. Negative prompting *in Style*

Recall the equation for classifier-free guidance [3]:

$$\hat{\epsilon} = \hat{\epsilon}_{\text{uncond}} + w\,(\hat{\epsilon}_{\text{prompt}} - \hat{\epsilon}_{\text{uncond}}) \tag{5}$$

In this equation, the predicted noise $\hat{\epsilon}_{\text{uncond}}$ when the U-Net is not conditioned on the textual prompt can be replaced by $\hat{\epsilon}_{\text{negative}}$, that is the predicted noise when conditioning the U-Net on a so-called negative prompt. With negative textual prompting, the classifier-free guidance becomes:

$$\hat{\epsilon} = \hat{\epsilon}_{\text{negative}} + w\,(\hat{\epsilon}_{\text{prompt}} - \hat{\epsilon}_{\text{negative}}) \tag{6}$$

Note that the two equations are equal when classifier-free guidance is not used, i.e., when the guidance weight $w = 1$. For $w > 1$, the direction $(\hat{\epsilon}_{\text{prompt}} - \hat{\epsilon}_{\text{negative}})$ is amplified, meaning the generated image should be more aligned with the textual prompt and less aligned with the negative textual prompt.

As expected, *Diffusion in Style* is also compatible with this negative prompting technique, as we show in Figure 29.

G. Failure cases

Generated images may not match the style correctly for high guidance weights In some cases, *Diffusion in Style* models favor content alignment over style by generating images that do not match the style exactly, as we show in Figure 20. This is generally a sign that the chosen guidance weight for this image is too high, and a better image might be generated with a lower classifier-free guidance weight.

For instance, in Figure 4, the image for "*Rainbow coloured penguin.*" in Pictogram style has colors, which does not match the style of pictogram images. In Figure 4, the image for "*A cross section view of a brain.*" also misses the typical characteristics of the *Starry Night* style, *i.e.* sky and stars.

A panda making latte art. *Rainbow coloured penguin.*

Figure 20. ***Diffusion in Style* with too high guidance weight**. As explain in Section 4.2, too high guidance weight leads to poor style matching. Increasing guidance weight typically generates images that better match the textual prompt, with a trade-off on the style. It can also be seen from Figure 8 that increasing the guidance weight too high can worsen both text-alignment and style-matching (styles 1, 2, and 4). In this figure, we generate images from the textual prompt indicated at the bottom, with a classifier-free guidance weight of 15.0, which appears to be too high.

Generated images may not match the textual prompt correctly, especially for low guidance weights In some cases, *Diffusion in Style* models generate images that match the style carefully, but do not show the desired content described in the textual prompt. This is generally a sign that the chosen guidance weight for this image is too low, as in Figure 21. In other cases, it can also indicate that the prompt is too complicated for the model.

A side view of an owl sitting in a field. *A panda making latte art.* *Rainbow coloured penguin.* *A cross-section view of a brain.*

Figure 21. ***Diffusion in Style* with too low guidance weight**. As explained in Section 4.2, too low guidance weight leads to poor content matching, meaning the generated images do not match the textual prompt precisely. Lowering guidance weight typically generates images that better match the desired style, with a trade-off on the content. In this figure, we generate images from the textual prompt indicated at the bottom, with a classifier-free guidance weight of 1.0, *i.e.* without classifier-free guidance.

In some cases, some details of the prompt are missing, *e.g.* in Figure 4, the owl does not appear side-viewed in the

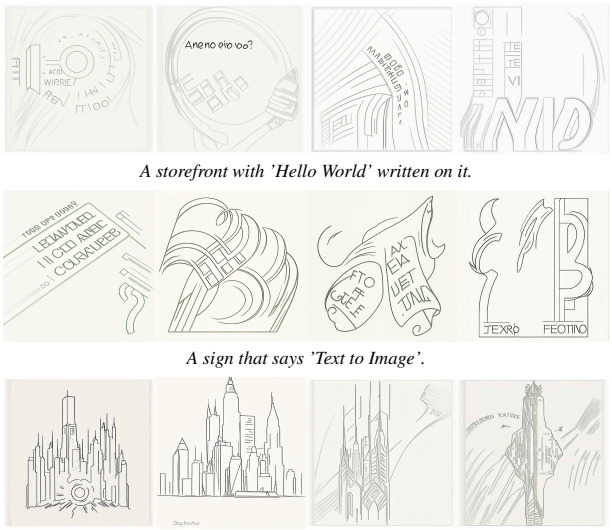
pictogram style. In some cases, two elements (*e.g. rainbow* and *penguin*) are represented separately instead of together. For instance, the image generated for "*Rainbow coloured penguin*" in the *Starry Night* style in Figure 4 shows a penguin in front of a rainbow, instead of a rainbow-coloured penguin. Conversely, two distinct elements (*e.g. panda* and *latte art*) can be merged into a single one. For instance, the image generated for "*Panda making latte art*" in the style of *Dalí* in Figure 4 represents a panda as a latter art, instead of a panda making latte art.

Insufficient number of target images As shown in the ablation study (Appendix C), a too-small set of target style images, for instance only 5 target style images, can lead to a model generating images with poor quality. We illustrate such an example in Figure 22. We propose modifications to our approach in Section 6, to obtain good results when the number of target images is that small.

A side view of an owl sitting in a field. | *A panda making latte art.* | *Rainbow coloured penguin.* | *A cross-section view of a brain.*

Figure 22. ***Diffusion in Style* with insufficient number of images**, without the modifications proposed in Section 6. We train a *Diffusion in Style* model on the *anime sketch* style using only 5 target images. As explain in Appendix C, this leads to poor performances. In this figure, we generate images from the textual prompt indicated at the bottom, with a classifier-free guidance weight of 8.0.

Illegible text When the generated images contains text, the generated text is often illegible. 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. In Figure 23, we further show examples of this limitation.



A storefront with 'Hello World' written on it.

A sign that says 'Text to Image'.

New York Skyline with 'Deep Learning' written with fireworks on the sky.

Figure 23. **Trying to generate images containing text with *Diffusion in Style***. Using the *Diffusion in Style* model for *anime sketches* to generate images containing text often does not work. Note that the same limitation applies to the original Stable Diffusion. In this figure, we used prompts from DrawBench [10] that involve text. The prompts are indicated below the generated images.

References

[1] 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. 7

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 5

[3] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2, 9

[4] 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. 6, 7, 8

[5] 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, 2

[6] 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. 5

[7] 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. 5

[8] 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. 3

[9] 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, 7

[10] 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. 3, 4, 7, 10

[11] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 1

[12] 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. 3

[13] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. 2

[14] 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. 3

[15] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models – GitHub repository. https://github.com/huggingface/diffusers, 2022. 2, 8, 9, 15, 16

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

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

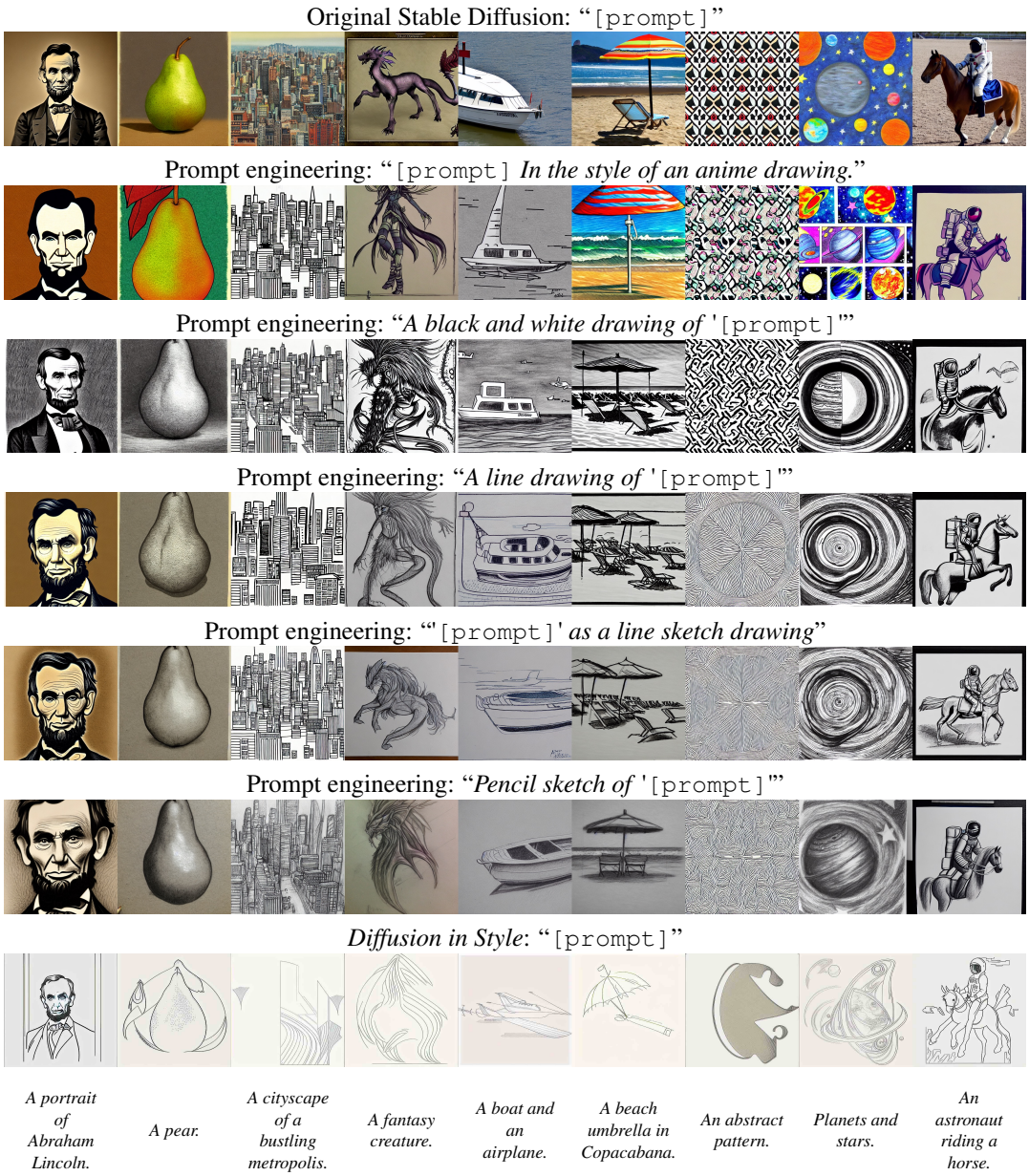


Figure 24. **Visual comparison of images generated with the original Stable Diffusion, with prompt engineering and with *Diffusion in Style*, for the *anime sketch* style.** The top row contains images generated with the original Stable Diffusion, the five middle rows contain images generated with the five indicated prompt engineering templates, and the bottom row contains images generated with *Diffusion in Style*. Each column is generated from the textual prompt indicated at the bottom. For quantitative evaluation of prompt engineering, Section 5 and Figure 8, we use the first template: "[prompt] *In the style of an anime drawing.*".

Figure 25. **Visual comparison of images generated with the original Stable Diffusion, with prompt engineering and with *Diffusion in Style*, for the *few-shot Pokemon* style.** The top row contains images generated with the original Stable Diffusion, the five middle rows contain images generated with the five indicated prompt engineering templates, and the bottom row contains images generated with *Diffusion in Style*. Each column is generated from the textual prompt indicated at the bottom. For quantitative evaluation of prompt engineering, Section 5 and Figure 8, we use the first template: "[prompt] *In the style of Pokemon, white background.*".

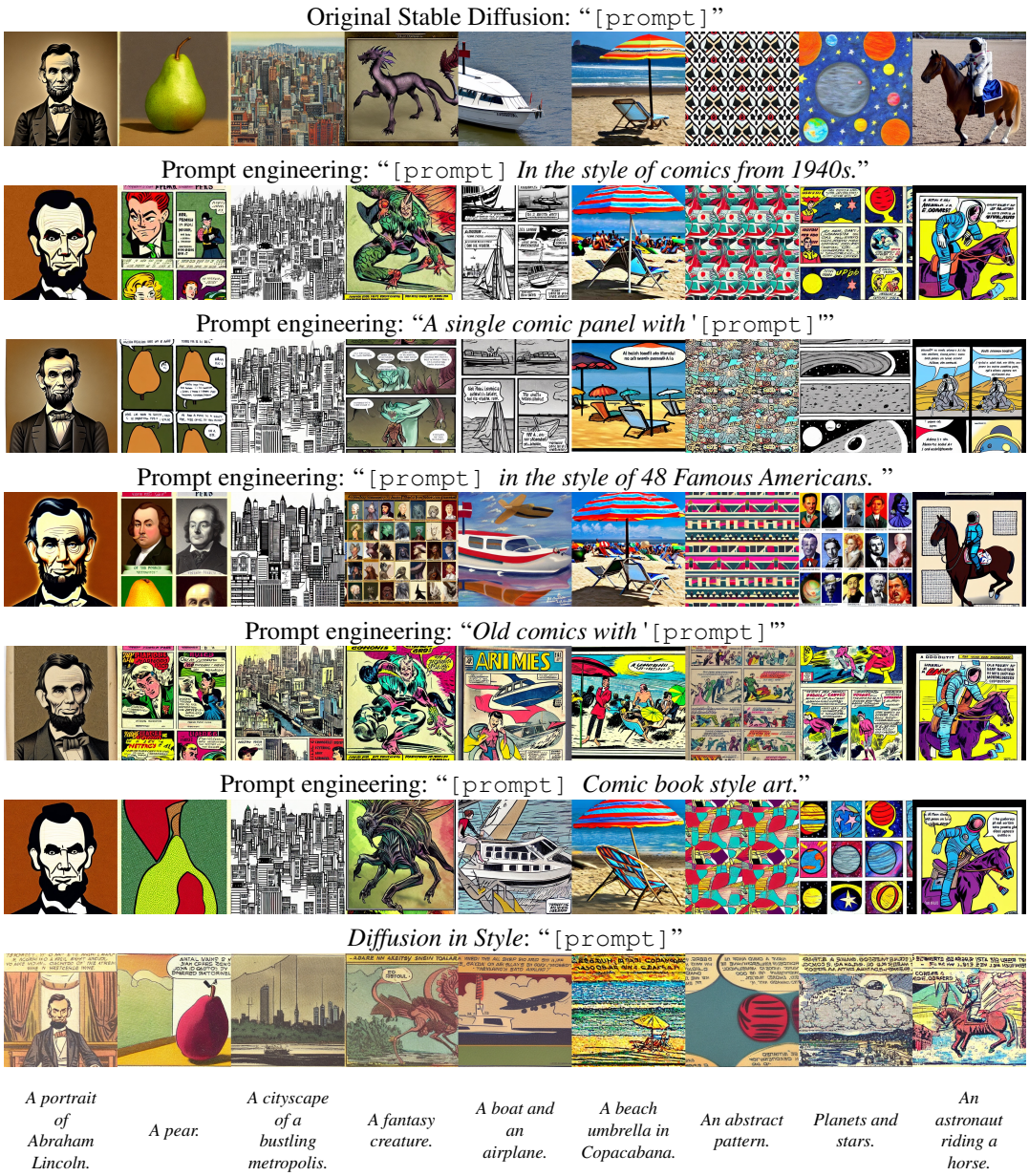


Figure 26. **Visual comparison of images generated with the original Stable Diffusion, with prompt engineering and with *Diffusion in Style*, for the *48 Famous Americans* style.** The top row contains images generated with the original Stable Diffusion, the five middle rows contain images generated with the five indicated prompt engineering templates, and the bottom row contains images generated with *Diffusion in Style*. Each column is generated from the textual prompt indicated at the bottom. For quantitative evaluation of prompt engineering, Section 5 and Figure 8, we use the first template: "[prompt] *In the style of comics from 1940s.*".

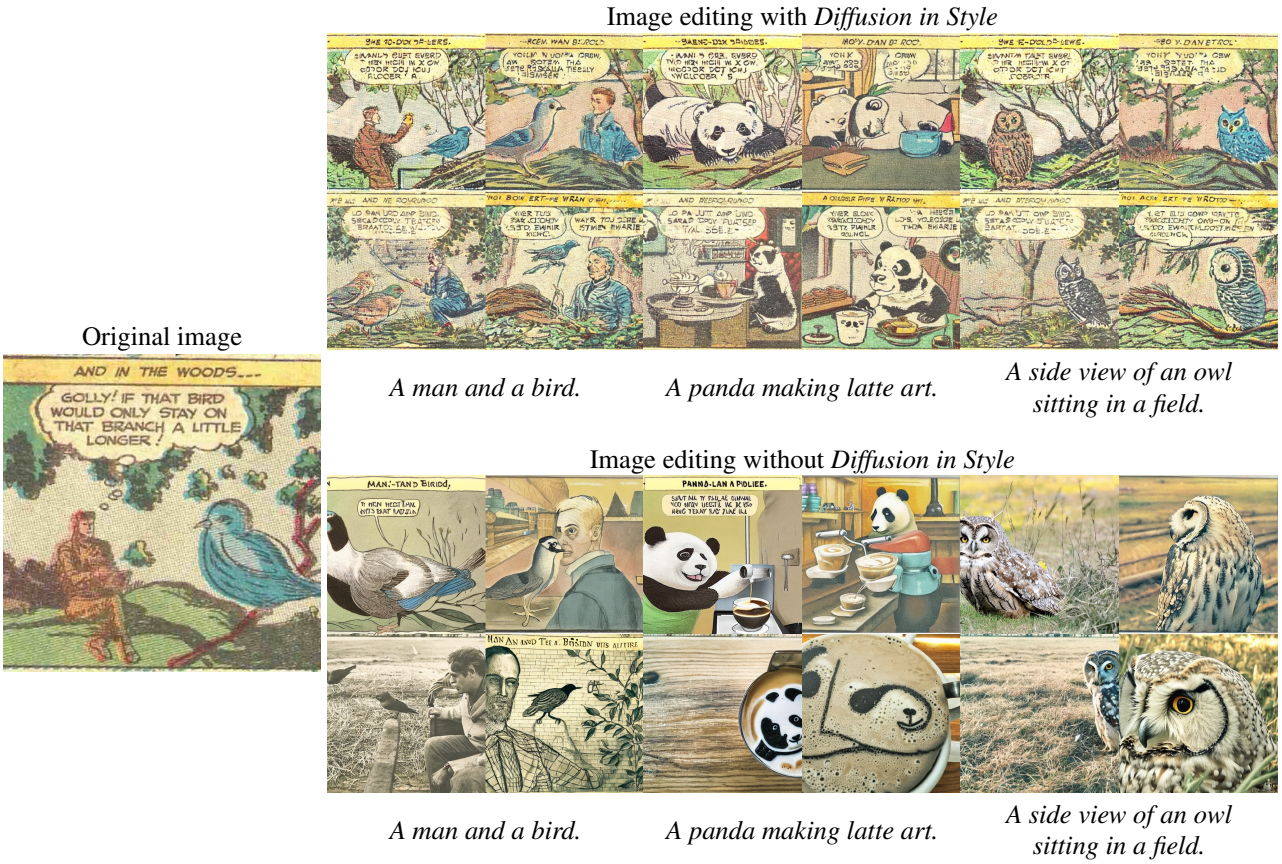


Figure 27. **Text-guided image editing with versus without *Diffusion in Style*, from an image of the *48 Famous Americans* style**. Comparison of text-guided image editing with and without *Diffusion in Style*. Given the original image (left), we generate 4 image edits for 3 different prompts, "*A man and a bird.*", "*A panda making latte art.*" and "*A side view of an owl sitting in a field.*". Image editing obtained with *Diffusion in Style* is shown in the top row, and image editing obtained with the original Image-to-Image pipeline is shown in the bottom row. The same seeds are used to generate image editing with and without *Diffusion in Style*. In this figure, image editing is generated with a strength of 70% (see the implementation of `StableDiffusionImg2ImgPipeline` in the Diffusers library [15] for details). As we see, Image-to-Image results without our technique drift from the expected style, while Image-to-Image results with *Diffusion in Style* stay faithful to the style.

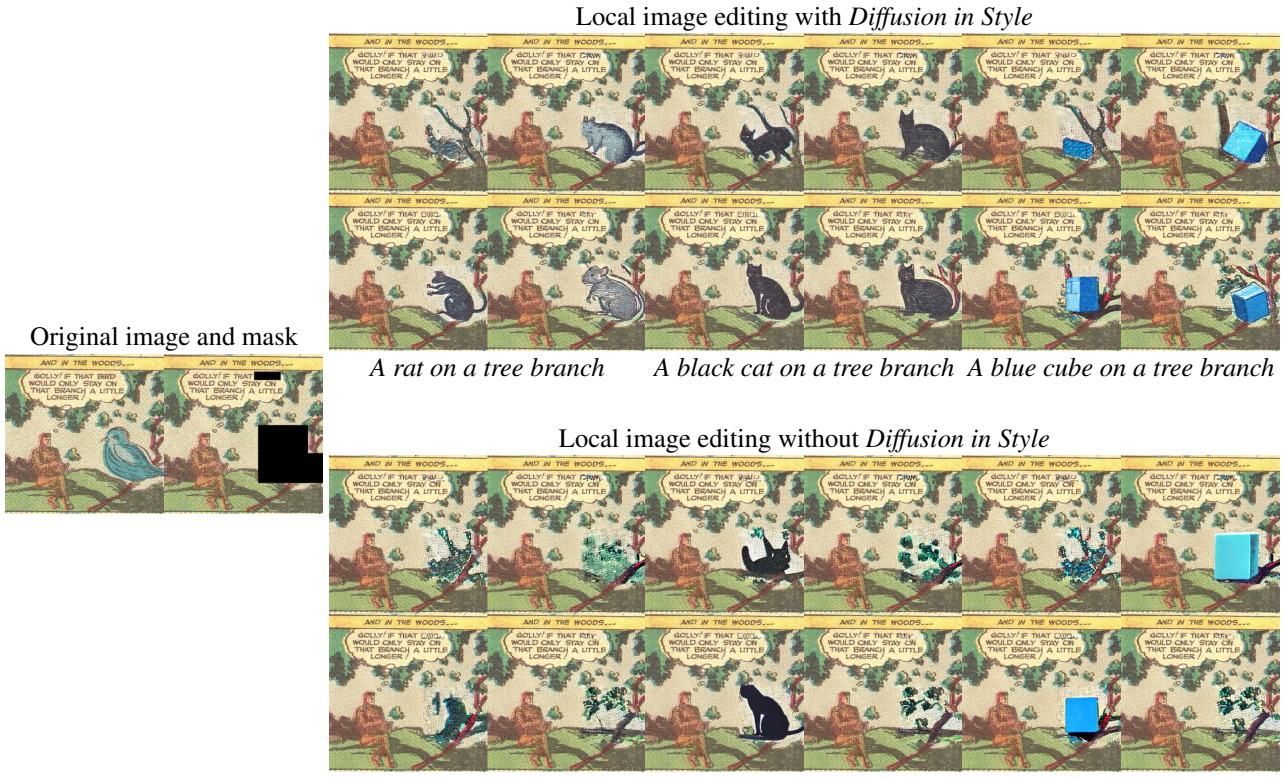


Figure 28. **Text-guided local image editing with versus without *Diffusion in Style*, from an image of the *48 Famous Americans* style**. Comparison of text-guided local image editing with and without *Diffusion in Style*. Given the original image with a mask (left), we generate 4 local image edits for 3 different prompts, "*A rat on a tree branch*", "*A black cat on a tree branch* " and "*A blue cube on a tree branch*". Local image editing obtained with *Diffusion in Style* is shown in the top row, and local image editing obtained with the original "legacy inpaint pipeline" is shown in the bottom row. The same seeds are used to generate image editing with and without *Diffusion in Style*. All local image edits are generated with the default strength of 80% (see the implementation of `StableDiffusionInpaintPipelineLegacy` in the Diffusers library [15] for details). As we see, local image editing results without our technique drift from the expected style, while local image editing results with *Diffusion in Style* stay faithful to the style.

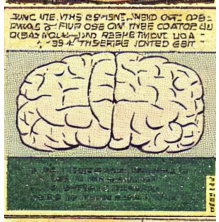
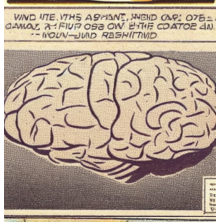
Positive prompt	Negative prompt	Generated image	Positive prompt	Negative prompt	Generated image
A side view of an owl sitting in a field.	Ø		*A side view of an owl sitting in a field.*	*Yellow.*	
A panda making latte art.	Ø		*A panda making latte art.*	*Details.*	
A cross-section view of a brain.	Ø		*A cross-section view of a brain.*	*Green description.*	
Rainbow coloured penguin.	Ø		*Rainbow coloured penguin.*	*Two characters.*	
A mouse using a mushroom as an umbrella.	Ø		*A mouse using a mushroom as an umbrella.*	*Pink.*	
A confused grizzly bear in calculus class.	Ø		*A confused grizzly bear in calculus class.*	*Teacher.*	

Figure 29. **Negative prompting with *Diffusion in Style***. Images generated *without* negative prompting are shown on the left, and images generated *with* negative prompting are shown on the right. In each row, the images are generated with and without negative prompting from the same, style-specific, initial latent tensor.