Stylecodes: Encoding Stylistic Information For Image Generation

Ciara Rowles

Corresponding author: crowles98@gmail.com

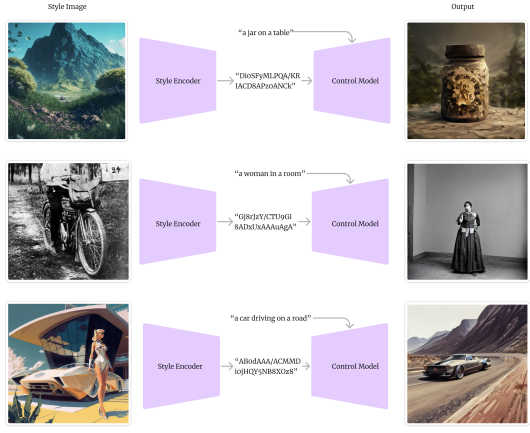

Fig. 1: Our Style Encoder compresses image styles into compact strings for style-conditioned generation.

Abstract. Diffusion models excel in image generation, but controlling them remains a challenge. We focus on the problem of style-conditioned image generation. Although example images work, they are cumbersome: srefs (style-reference codes) from MidJourney solve this issue by expressing a specific image style in a short numeric code. These have seen widespread adoption throughout social media due to both their ease of sharing and the fact they allow using an image for style control, without having to post the source images themselves. However, users are not able to generate srefs from their own images, nor is the underlying training procedure public. We propose StyleCodes: an open-source and open-research style encoder architecture and training procedure to express image style as a 20-symbol base64 code. Our experiments show that our encoding results in minimal loss in quality compared to traditional image-to-style techniques.

Keywords: Image Generation, Diffusion Models, Image Conditioning

1 Introduction

The field of image generation has received fresh impetus from diffusion model theory, where it is seen as an iterative process that reverses the diffusion of images into pure noise. Diffusion models have achieved state-of-the-art performance in image generation tasks and have proven to be more robust to train than the previous GAN-based models. They are also more widely applicable in practice, in large part because of the classifier-free guidance technique, which offers unprecedented control over the output image using a text prompt [7].

However, the common proverb "An image is worth a thousand words" is as relevant as ever: crafting text prompts that result in exactly the desired image is a non-trivial task, referred to as *prompt engineering*. Even when the desired image is clear in the mind of the user, it is difficult to express it in such a way that the model output replicates it accurately; expressing intent through images is often far more intuitive than through text. This concept has spurred the development of image-based conditioning such as ControlNet [23] and IPAdapter [16, 21] approaches. From scribbles and sketches or stylistic examples to many other modalities: these methods allow users to express their intent in the image domain, where spatial information and stylistic cues can be more easily conveyed.

Yet in practice, these control methods lack controllability and the ability to collaborate broadly without extensive shared workflows. We tackle this problem by first creating "stylecodes", 20 digit base64 encoded codes with a combination of a custom encoder and a custom Stylecode-conditioned model for control of the UNet, in this case with Stable Diffusion 1.5 [15]. We outline how to both encode and decode a stylecode and how this can be used to generate stylized images in normal contexts, as well as how the complete architecture is trained.

2 Related Work

2.1 Diffusion Models

Diffusion models generate images conditioned on text by learning to reverse a gradual diffusion process [6, 17, 18], typically in a latent-pixel domain for its efficiency and low-level prior [4, 14]. Unfortunately, these diffusion models can involve significant training costs. While much work is focused on inference speed or distillation into smaller models, we consider those efforts orthogonal to our work and leave out the relevant literature for brevity's sake.

However, the text prompts that condition these models are finicky and inaccurate for conveying user intent [20], especially in terms of style. Although negative prompts provide additional control, they can interfere with the original prompt or even be ignored [1], while still being restricted in their expressiveness. These difficulties imply a need for more expressive control, which we believe to be in images. Therefore, we now discuss both image-based conditioning for diffusion models and diffusion models for image-to-image translation tasks.

2.2 Image-based Control for Diffusion Models

InstantStyle [19] conditions the output on the style of an input image without training. To do so, the CLIP space embedding of its textual description is subtracted from that of the image to obtain a "style direction vector" in CLIP space: the text prompt cross attention layers in some blocks are then extended to also attend to this style vector. While this model is neither explicitly trained to model the conditioning process, its success is undeniable. In our research, we have found that the InstantStyle method produces the best style conservation process using conditioned images.

IPAdapter [21] comprises a small neural translator to project from the input image's embedding, such as from ViT-H/14 [2] CLIP [12], onto the embedding space used by the text encoder; the network cross-attends to these novel embeddings in additional cross-attention layers similar to those of the text prompt, effectively enabling it to use an image as prompt input. IPAdapter is trained to reproduce the input image exactly — only "by coincidence" is the emergent behaviour of the model to flexibly transfer the style and content of the condition image to the output images. Our system leverages the lessons learnt from IPAdapter and InstantStyle to make the conditioning explicit, but bottlenecked through the Encoder-Decoder.

UnCLIP-based Approaches [13] retrain the base model to reproduce an image based on its CLIP embedding, similar to IPAdapter, but replace the text prompt with the image condition completely, attaining a single mode of control over the output. Since the entire model is retrained for this purpose, it risks catastrophic forgetting of the original model's capabilities, and is incompatible with other residual changes to the base model such as LoRA's [8]: we instead prefer to use seperate control models in order to keep the base model intact.

ControlNet [23] functions by cloning the base model's encoder. The clone's inputs are replaced with the conditioning image, and its outputs are added as residuals to the original model's hidden states [22,23]. ControlNet excels at pixel-aligned conditioning of the output; it is not very succesful at style alignment. However, we have found that the residual way in which the ControlNet affects the base model's output *is* amenable to style alignment: just through conditioning on a stylecode rather than an input image.

3 Method

3.1 Preliminaries

Diffusion models [6, 17, 18] iteratively reverse a diffusion process that gradually transforms an image into noise (typically white Gaussian noise). We write the forward noise process as starting from the data distribution $\boldsymbol{z}_0 \sim p(\boldsymbol{z})$ and ending with pure noise samples $\boldsymbol{z}_T \sim \mathcal{N}(0, \mathbf{I})$, over the course of T time steps. The immediate forward process is formally specified as

$$\boldsymbol{z}_{t+1} \sim p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t) = \mathcal{N}\left(\sqrt{\alpha_{t+1}}\boldsymbol{z}_t, (1-\alpha_{t+1})\mathbf{I}\right), \tag{1}$$

where α_t denotes the so-called noise schedule. Given this forward process, the diffusion model is trained to model the immediate denoising distributions, noted as $\hat{p}(\boldsymbol{z}_t|\boldsymbol{z}_{t+1})$. During training, time steps are randomly sampled, and we directly supervise $\hat{p}(\boldsymbol{z}_t|\boldsymbol{z}_{t+1})$ by first sampling $p(\boldsymbol{z}_t|\boldsymbol{z}_0), \boldsymbol{z}_0 \sim p(\boldsymbol{z})$ followed by $\boldsymbol{z}_{t+1} \sim p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)$. Luckily, the exponential nature of additive white Gaussian noise implies a closed-form direct conditional $\boldsymbol{z}_t \sim p(\boldsymbol{z}_t|\boldsymbol{z}_0)$ which means that sampling $p(\boldsymbol{z}_t|\boldsymbol{z}_0)$ is constant-cost in terms of t. By iteratively running the diffusion model for subsequent time steps, we can sample from the full generative model, written as

$$\hat{p}(\boldsymbol{z}_0|\boldsymbol{z}_T) = \prod_T^1 \hat{p}(\boldsymbol{z}_t - 1|\boldsymbol{z}_t), \qquad \boldsymbol{z}_T \sim \mathcal{N}(0, \mathbf{I}). \tag{2}$$

As completely unconditional sampling is not very useful, diffusion models are trained to condition the generative distribution on an auxiliary input text prompt $\mathcal{T}$, modeling instead $p(\boldsymbol{z}_0|\boldsymbol{z}_T, \mathcal{T})$ (known as *Classifier-Free Guidance* [7]). In our case, we wish further controllability by additionally conditioning the generative model on a style $\mathcal{S}$ to model $p(\boldsymbol{z}_0|\boldsymbol{z}_T, \mathcal{T}, \mathcal{S})$. We wish to leverage a pre-trained text-conditioned model by reusing its weights and adding small modules. The base model is kept frozen to preserve its generative performance and expressivity.

IPAdapter In IPAdapter, for example, cross-attention layer that attend to (a CLIP [12] embedding of) the style image are added after every prompt cross-attention layer. The model is then trained by first sampling a data element $\boldsymbol{z}_0$ and then setting $\mathcal{S} = \text{CLIP}(\boldsymbol{z}_0)$. *I.e.* the model is only supervised to reproduce the condition image exactly — even though this is not the (only) intended use-case. Although the model is never trained to produce images with a different caption than the condition image, it shows emergent capabilities to do exactly that: it tends to generate images with the same style, composition, and identities as the condition. However, it lacks any controllability of these aspects, and sometimes fails any or all of these aspects, depending on the text prompt.

3.2 StyleCode / Image Conditioning

As discussed in Sec. 2 and Sec. 3.1, existing techniques that condition on images do not provide a clear way to easily share styles with others. To enable this, we encode the style-defining image as a 20 digit base64 stylecode, which can then be used to condition the image generation.

Model architecture Our model architecture is a combination of a basic attention based autoencoder pair and a controlnet-style UNet decoder for residually controlling the frozen base model. As the style codes are in embedding space, a decoder-based Stylecode-conditioned model suffices.

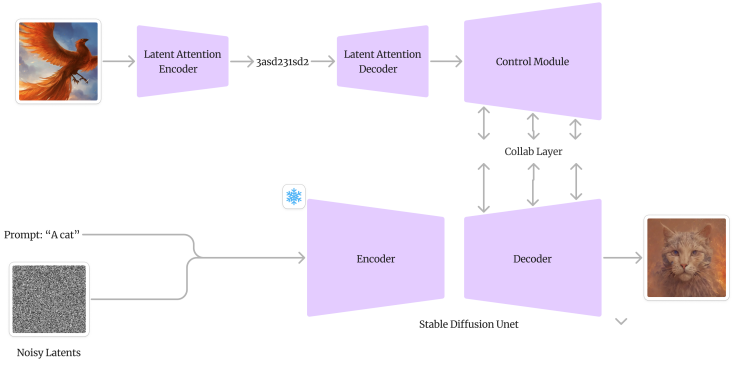

Fig. 2: Auto Encoder and Control Module Architecture

AutoEncoder For our system, we used an latent autoencoder, The Encoder works by using 3 attention layers to attend from a high channel representation of the latent to the embeddings, the latent is then projected down to the 20-dimensional latent size for the stylecode.

Later ,the Decoder takes this latent, makes a new output latent of the same shape as the midblock states for the Styleref Control Module and attends to the passed-in stylecode latent through its 3 layers.

During training to avoid discretization issues, the base64 encoding/decoding step is skipped, so the basic output latent is simply passed to the decoder. This is done because it allows you to use the model to train the latent used for the final stylecode, while ensuring a consistent flow of gradients during backpropagation. This is trained jointly with the Stylecode-conditioned model to ensure the closest match between desired results and outputs.

Stylecode For embedding the 20 dimensional latent, we chose to use base64 encoding due to it's widespread adoption and reliability, reducing any potential issues with character decoding later. To do so, we quantize each of the 20 dimensions independently. We also added a single number to the end of the stylecode, meant to denote the current stylecode encoder version. This way you can know if the stylecode you are looking at was made for a different decoder at a glance.

Stylecode-conditioned model For the Stylecode-conditioned model interaction with the main UNet, we used a decoder only version of the ControlNet scheme, residually affecting the internal hidden states of the base model. In order to maximize the performance of the Stylecode-conditioned model , it consumes the internal state of the base model at every communication point (much like Collaborative Control citeCollaborativeControl, except that our secondary model does not produce any output).

Image Encoder For our Image encoder, we chose to use SigLip [need citation] due to it's superior performance over CLIP. It provides effective and efficient image embeddings and in our experiments trained drastically faster than simply passing the image wholesale into the Style Encoder .

3.3 Dataset Generation

For our dataset generation, we relied on InstantStyle [19] , combined with source images from the MidJourney [9] dataset and [5]CommonCanvas, which were used as conditions for InstantStyle [19] running with SDXL [11] IP Adapter [21] and with prompts from a random image in the JourneyDB dataset [10]. This resulted in 35,000 condition, style and prompt dataset entries at a resolution of 1024x1024, which we downscaled to 512x512 for training (the native resolution of the base model).

3.4 Training Process

Our training process follows a simple training procedure with the base model frozen and the new layers initialized to white noise ($\sigma = 10^{-4}$). The residual layers are zero-initialized to ensure the base model is unaffected on initialization. The base model of choice is StableDiffusion 1.5 [14] for its excellent balance of output diversity, controllability and accessibility — it remains a staple model in the community for these reasons. We use a batch size of 25 and a learning rate of 10^{-6} for a total of 100000 steps, unless specified differently in the experiments.

4 Results

As shown in Fig. 3, style is effectively enforced from the stylecode with this method.

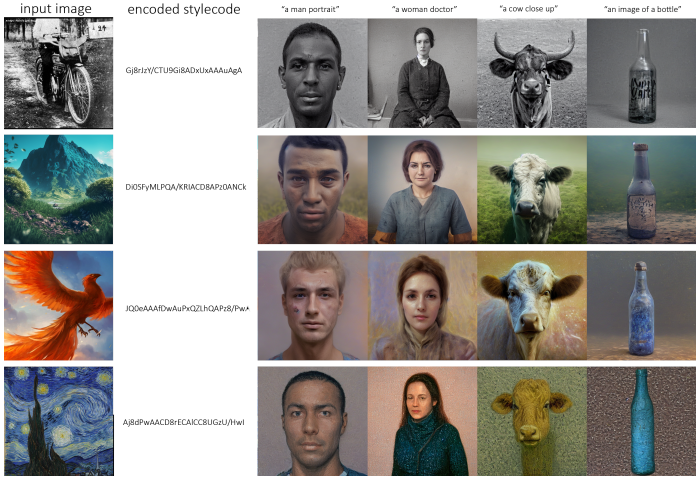

Fig. 3: Example results with the left-most column being the source image with prompts "a close up man", "a woman portrait", "a cow" and "a bottle on a desk" with the same seeds after passing through the encoder to a stylecode and then used to generate the images.

An additional benefit shown in Fig. 4 of the architecture is that due to the frozen base model, it can be switched out with fine-tuned variants with minimal performance degradation.

Fig. 4: Example of the results with various trained base models and the control module with the prompt "portrait of a man" and four different style images

5 Conclusion, Limitations, and Future Work

In this work, we have introduced stylecodes in order to allow social methods of control for image generation diffusion models conditioned on style images. By introducing this simple code condition, it is possible to easily share style information with friends about work done and use this to control local image generation models. This is all while fully preserving the functionality of the base model and it's compatibility with other control solutions like Loras and IP-Adapter. We note that this partially replicates, but significantly expands, the functionality of MidJourney's srefs. Furthermore, we openly publish our findings as well as source code, both for training and for inference.

We found that the main limitation was the training cost for the control model. While it is relatively low for Stable Diffusion 1.5, larger DiT based models quickly become unwieldy and costly. We also found that the dataset biased the distribution of the output model significantly, resulting in too narrow of a range of results, as well as issues with generations being reinforced in the final model. Using a combination of synthetic and real data may be a better solution to this in future models.

We feel that this model and architecture will lead to more practical and sociable methods of control, allowing collaborative image generation between multiple people. In terms of future work, we highlight the interplay with CFG, and specifically the option for multiple guidance [3]. Additionally, using larger models and a more varied dataset will almost certainly improve the diversity of styles that can be created.

References

1. Ban, Y., Wang, R., Zhou, T., Cheng, M., Gong, B., Hsieh, C.J.: Understanding the impact of negative prompts: When and how do they take effect? arXiv preprint arXiv:2406.02965 (2024)
2. Beaumont, R.: Vit-h/14 clip model (2023), https://huggingface.co/laion/CLIP-ViT-H-14-laion2B-s32B-b79K [Accessed: July 15th, 2024]
3. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023)
4. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first International Conference on Machine Learning (2024)
5. Gokaslan, A., Cooper, A.F., Collins, J., Seguin, L., Jacobson, A., Patel, M., Frankle, J., Stephenson, C., Kuleshov, V.: Commoncanvas: Open diffusion models trained on creative-commons images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8250–8260 (2024)
6. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
7. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021)
8. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022)
9. MidJourney: Midjourney – ai-based image generation tool. https://www.midjourney.com (2024), accessed: 2024-11-19
10. Pan, J., Sun, K., Ge, Y., Li, H., Duan, H., Wu, X., Zhang, R., Zhou, A., Qin, Z., Wang, Y., Dai, J., Qiao, Y., Li, H.: Journeydb: A benchmark for generative image understanding (2023)
11. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis (2023), https://arxiv.org/abs/2307.01952
12. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
13. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
14. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
15. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2022), https://arxiv.org/abs/2112.10752
16. Rowles, C., Vainer, S., Nigris, D.D., Elizarov, S., Kutsy, K., Donné, S.: Ipadapter-instruct: Resolving ambiguity in image-based conditioning using instruct prompts (2024), https://arxiv.org/abs/2408.03209
17. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Bach, F., Blei, D. (eds.)

Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 2256–2265. PMLR, Lille, France (07–09 Jul 2015), https://proceedings.mlr.press/v37/sohl-dickstein15.html
18. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2020)
19. Wang, H., Wang, Q., Bai, X., Qin, Z., Chen, A.: Instantstyle: Free lunch towards style-preserving in text-to-image generation. arXiv preprint arXiv:2404.02733 (2024)
20. Witteveen, S., Andrews, M.: Investigating prompt engineering in diffusion models. arXiv preprint arXiv:2211.15462 (2022)
21. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arxiv:2308.06721 (2023)
22. Zavadski, D., Feiden, J.F., Rother, C.: Controlnet-xs: Rethinking the control of text-to-image diffusion models as feedback-control systems (2024), https://arxiv.org/abs/2312.06573
23. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)