Style-Friendly SNR Sampler for Style-Driven Generation

Supplementary Material

A. Experimental Details

A.1. Style Prompts

We conduct all quantitative evaluations using the 18 reference styles shown in the appendix of the StyleDrop paper [17]. The style prompts for these 18 styles can also be found in the StyleDrop appendix.

A.2. Evaluation Prompts

We present 23 evaluation prompts collected from StyleDrop paper [17] used for our quantitative and qualitative comparisons:

- An Opera house in Sydney in {style prompt} style
- A fluffy baby sloth with a knitted hat trying to figure out a laptop, close up in {style prompt} style
- A Golden Gate bridge in {style prompt} style
- The letter 'G' in {style prompt} style
- A man riding a snowboard in {style prompt} style
- A panda eating bamboo in {style prompt} style
- A friendly robot in {style prompt} style
- A baby penguin in {style prompt} style
- A moose in {style prompt} style
- A towel in {style prompt} style
- An espresso machine in {style prompt} style
- An avocado in {style prompt} style
- A crown in {style prompt} style
- A banana in {style prompt} style
- A bench in {style prompt} style
- A boat in {style prompt} style
- A butterfly in {style prompt} style
- An F1 race car in {style prompt} style
- A Christmas tree in {style prompt} style
- A cow in {style prompt} style
- A hat in {style prompt} style
- A piano in {style prompt} style
- A wood cabin in {style prompt} style

In Fig. S1, we present the detailed style descriptions generated by GPT-4o, which serve as text prompts for the *GPT-4o Prompt* baseline. Specifically, for each of the 18 reference styles, we obtained comprehensive textual descriptions using GPT-4o. These detailed prompts were directly used for text-to-image generation.

A.3. User Study

In this section, we provide detailed information about the setup of our user study. Our user study aims to measure human preferences in two key objectives of style-driven image generation: style alignment and text alignment. To assess these preferences, we conduct pairwise comparisons between our method and each baseline for each objective. Participants are shown the reference image, target text prompt, and two generated images (one from each method) and are asked to choose the image that better satisfies the objective. We collect three responses from each of the 150 participants, resulting in a total of 450 responses for each comparison. The full instructions used in our questionnaires are as follows.

For style alignment objective,

- Given a reference image and two machine-generated images, select which machine-generated output better matches the style of the reference image for each pair.
- Please focus only on the style including color schemes, layouts, illumination, and brushstrokes.
- If it's difficult to determine a preference, please select "Cannot Determine / Both Equally".

For text alignment objective,

- Given a reference image and two machine-generated images, select which machine-generated output better matches the target text for each pair.
- Please focus only on the text, without regard for the reference image.
- If it's difficult to determine a preference, please select "Cannot Determine / Both Equally".

A.4. Implementation

To ensure reproducibility, we provide pseudo-code implementations of Style-friendly SNR samplers in Fig. S2 and the addition of LoRA [7] parameters to MM-DiT for training in Fig. S3. All experiments are conducted on a NVIDIA A40 GPU with 48GB memory.

Optimizer and Learning Rate. We use the Adam optimizer [9] at a learning rate of 10^{-4} for 300 steps. The batch size is set to 1, with gradient accumulation over 4 steps.

Guidance Scale and Inference Steps. During inference, we use a guidance scale [6, 11] of 7.0. The number of denoising steps is set to 28.

Model-Specific Details. FLUX-dev [1] is a 12B guidance-distilled model [11] that takes guidance scale as input. For fine-tuning, we fix the guidance scale to 1.0 to match standard diffusion training, enable gradient checkpointing for memory efficiency, and use BF16 quantization for both fine-tuning and inference. We use 8B parameter model of SD3.5 [21] and Sana-1600M-1024px-BF16 model of SANA [22]. We do not use gradient checkpointing when fine-tuning these models.

Disabling Timestep Shifting. SD3 [3] uses a timestep shifting mechanism. However, we disable this shifting for our Style-friendly SNR sampler to isolate the effect of our proposed SNR sampling strategy.

Baseline Implementation. We use the Hugging Face Diffusers library (version 0.31.0) for consistent training and inference across methods.

B. Baselines

B.1. Direct Consistency Optimization

Direct Consistency Optimization (DCO) [10] is a fine-tuning method inspired by direct preference optimization [14] commonly used in large language models (LLMs). Instead of directly minimizing the diffusion loss, DCO aims to ensure that the diffusion loss of the fine-tuned model is lower than that of the pre-trained model on the reference data. The objective function is defined as:

$$\mathcal{L}_{DCO}(x_0) = \mathbb{E}_{t,\epsilon}\Bigg[- \log\sigma(-\beta T \\ ||v_\theta(x_t,t) - v(x_t,t)||^2 - ||v_\phi(x_t,t) - v(x_t,t)||^2)\Bigg], \quad (1)$$

where $v(x_t, t)$ is target velocity field, v_θ, is fine-tuning model, and v_ϕ is frozen pre-trained model.

In this objective, the parameter βT controls the strength of the preference towards the fine-tuned model over the pre-trained model. DCO increases the relative likelihood of the fine-tuned model over the pre-trained model, penalizing less when the fine-tuned model's loss is smaller. This helps preserve the text-to-image alignment of the pre-trained model.

However, DCO requires computations involving both the fine-tuned and pre-trained models, making it computationally more intensive than directly fine-tuning using the standard diffusion loss. In our experiments, we observed that using a large value of $\beta T = 1000$ resulted in slower convergence and suboptimal performance. Therefore, we set $\beta T = 1$ to achieve better results.

B.2. IP-Adapter

IP-Adapter [23] is designed to enable text-to-image models to generate identity-preserving images by training a compact adapter that encodes CLIP image embeddings [13]. This adapter introduces the CLIP image embedding as an additional input by concatenating its output with the text embeddings. The parameter-efficient nature of IP-Adapter allows for easy training and deployment across various text-to-image models. However, a notable limitation is its restricted style alignment due to the expressive constraints of CLIP embeddings, which may result in generated images that do not fully capture detailed stylistic characteristics. IP-Adapter allows adjusting the conditioning strength by scaling the embeddings with a factor between 0 and 1; we use a scale of 0.6 in all experiments. Using a scale of 1 can lead to content leakage beyond the style.

B.3. RB-Modulation

RB-Modulation [15] is a zero-shot approach using Stable Cascade [12], a model accepting both CLIP image embeddings and text embeddings as inputs. During the denoising process, RB-Modulation employs gradient guidance of a CSD [18], a model fine-tuned from CLIP to measure style similarity, resembling classifier guidance [2]. At each denoising step, CSD computes the similarity between the approximated x_0 and the reference image, guiding the generation process to enhance this similarity. RB-Modulation also aggregates multiple attention features.

However, this approach relies on models that accept CLIP image embeddings, limiting model selection. Additionally, using gradient guidance of CSD increases inference costs, making the generation process more computationally intensive.

B.4. Style-Aligned

Style-Aligned [5] generates consistent sets of images with the same style by ensuring that features of each image attend to those of a reference image through shared key and value features in self-attention layers of image tokens. It first maps the reference image to noise using DDIM inversion [19] and shares self-attention features during denoising. The fidelity to the reference style can be controlled by amplifying the self-attention logits in the diffusion model. However, Style-Aligned is not directly applicable to MM-DiT [3] architecture that lacks image-only self-attention layers. Moreover, artificially amplifying self-attention logits can lead to artifacts and lower-quality images due to conflicting attention features.

B.5. Offset Noise

Offset noise [4] is a method proposed to fine-tune diffusion models for generating monochromatic images. During the diffusion process, a constant offset noise—identical across all pixel positions—is added to the standard Gaussian noise, scaled by a small factor (e.g., 0.1). This introduces an explicit bias toward monotonic noise patterns, encouraging the model to learn and reproduce solid colors. While offset noise aids in learning monotonous patterns, it can hinder the model's capacity to learn more complex styles.

Here, we additionally experiment with incorporating offset noise into our training process in Tab. S1. Offset noise with a scale of 0.1 improves the SD3 sampler's results in DINO and CLIP-I scores, as many reference styles from the StyleDrop paper [17] have monochromatic backgrounds, favoring this trick. However, it still does not reach the performance of our Style-friendly SNR sampler. Moreover, when we combine our Style-friendly approach with a smaller scale of offset noise (0.01), we observe a slight improvement in the style alignment of FLUX-dev.

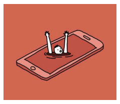
Soft washes of color with delicate shading and detailed linework emphasize the texture and character of the building. The use of light and shadow enhances depth, creating a classic, timeless watercolor illustration.

A loose and fluid approach with vibrant color transitions, blending warm and cool tones in an organic manner. The soft edges and bleeding pigments create a dreamy, almost ethereal effect.

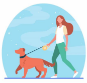
Simple, playful strokes with bright, unmixed colors give a naive and spontaneous feel. The use of bold outlines and uneven brushwork adds to the charm of a carefree, unrefined artistic approach.

Thick, visible brushstrokes with swirling, dynamic patterns create a sense of movement and depth. The use of bold, contrasting colors and a rich impasto texture gives the image a highly expressive and emotional atmosphere.

Expressive and bold brushstrokes create a sense of movement and texture. The use of vibrant yet earthy tones, along with swirling background patterns, adds a dynamic and emotional depth to the composition.



A simple and clean cartoon-like approach using bold outlines and flat colors. The exaggerated, symbolic composition conveys a strong message with minimal visual elements, making it both modern and impactful.



A clean and modern aesthetic with soft color palettes and smooth gradients. The absence of outlines and the use of simple geometric shapes contribute to a polished and contemporary design.

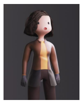
A monochrome composition created with flowing, dynamic strokes. The expressive lines and rhythmic movement give it an energetic and almost dreamlike atmosphere, reminiscent of classic pen-and-ink studies.

Smooth, flowing, and ethereal ribbons of vibrant color set against a dark background. The soft glow and gradient blending create a futuristic and dynamic visual effect.

A glowing, bioluminescent effect reminiscent of an X-ray scan, using bright cyan tones against a pitch-black background. The high contrast and fine details emphasize a surreal and otherworldly aesthetic.

Clean, modern, and minimalistic design using soft pastel tones and geometric shapes. The absence of outlines and the smooth gradients give it a polished, professional feel.

Bold, flat design with crisp, clean edges and a drop shadow effect, making it appear like a sticker or cutout. The smooth color transitions and simplified shapes add to the playful and modern aesthetic.



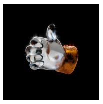
A smooth, polished, and cartoonish 3D model with simplified facial features. The soft shading, realistic hair texture, and subtle lighting create a blend of realism and stylization.

A simple, naive, and hand-drawn aesthetic with visible crayon strokes and uneven coloring. The rough texture and imperfect lines add to the charm of a playful, unpolished artwork.

A highly reflective, gold-like surface with smooth, liquid-like textures. The dripping effect and polished finish create a futuristic and surrealistic feel.

A detailed and rustic carving with deep grooves and textured patterns. The organic, handcrafted appearance gives it an ancient, folklore-inspired look.

Bright, exaggerated, and glossy 3D modeling with a soft, cartoonish aesthetic. The smooth reflections and bold, saturated colors create a fun and whimsical atmosphere.



A high-gloss, semi-transparent glass effect combined with a metallic base. The dramatic lighting and reflections enhance the sci-fi and high-tech aesthetic.

Figure S1. **Detailed style description.** These descriptions serve as the input prompts for our GPT-4o [8] prompt baseline, which generates images solely based on these textual style specifications.

```
# Inputs: mu, sigma, B, latent
# sample log-SNR
logsnr = torch.normal(mean=mu, std=sigma, size=(B,))
# compute timestep t
t = torch.nn.functional.sigmoid(-logsnr / 2).view(B, 1, 1, 1)
# sample noise
noise = torch.randn_like(latent)
# diffuse latent
noisy_latent = (1.0 - t) * latent + t * noise
```

Figure S2. **PyTorch implementation of a Style-friendly SNR sampler.**

```python
# Inputs: model_name, rank
# Configure LoRA for the specified model
if model_name == "FLUX":
    target_modules = [
        "to_k", "to_q", "to_v", "to_out.0",
        "add_k_proj", "add_q_proj", "add_v_proj", "proj_mlp", "proj_out"
    ]
elif model_name == "SD3":
    target_modules = [
        "to_k", "to_q", "to_v", "to_out.0",
        "add_k_proj", "add_q_proj", "add_v_proj", "to_add_out"
    ]
else:
    raise ValueError(f"Unsupported␣model:␣{model_name}")

# LoRA configuration
transformer_lora_config = LoraConfig(
    r=rank,
    lora_alpha=rank,
    init_lora_weights="gaussian",
    target_modules=target_modules,
)

# Add adapter to the transformer
transformer.add_adapter(transformer_lora_config)
```

Figure S3. **PyTorch implementation of LoRA integration.**

Method	Model	Metrics		
		DINO ↑	CLIP-I ↑	CLIP-T ↑
SD3 Sampler [3]	SD3.5	0.424	0.670	0.350
w/ offset 0.1	SD3.5	0.452	0.678	0.353
Style-friendly	SD3.5	0.489	0.698	0.349
w/ offset 0.01	SD3.5	0.476	0.697	0.350
SD3 Sampler [3]	FLUX-dev	0.373	0.645	0.350
w/ offset 0.1	FLUX-dev	0.451	0.679	0.349
Style-friendly	FLUX-dev	0.461	0.686	0.344
w/ offset 0.01	FLUX-dev	0.500	0.704	0.341

Table S1. **Incorporating offset noise.** Offset noise improves SD3 sampler but still does not reach the performance of our Style-friendly SNR sampler; combining our Style-friendly approach with Offset Noise at a smaller scale (0.01) slightly enhances the style alignment of FLUX-dev. Here, we use $\sigma = 2$ for Style-friendly.

This quantitative evaluation is based on the monochromatic backgrounds prevalent in the StyleDrop [17] references. Our qualitative comparisons in Fig. S6 show that offset noise struggles with complex references, failing to capture intricate stylistic details. This indicates that while offset noise can help with simple, uniform styles, it is vulnerable to complex styles.

Method	Model	Metrics		
		DINO ↑	CLIP-I ↑	CLIP-T ↑
SD3 Sampler [3]	FLUX-dev	0.373	0.645	0.350
w/ rank 128	FLUX-dev	0.426	0.668	0.345
Style-friendly	FLUX-dev	0.461	0.686	0.344

Table S2. **Comparison to increasing LoRA rank.**

Method	DINO	CLIP-I	CLIP-T
Style-friendly	0.489	0.698	0.349
w/o Text attn	0.462	0.693	0.349

Table S3. **Ablation study on trainable parameters.**

C. Additional Results

C.1. Quantitative Results

CLIP Scores. In the main paper, we presented analyses of the mean μ, standard deviation σ, and LoRA rank using the DINO similarity score. In Fig. S4a, we provide the corresponding CLIP image similarity (CLIP-I) scores to further validate our findings. The CLIP-I scores exhibit a similar trend to the DINO scores, where decreasing μ enhances style alignment. Varying σ affects the CLIP-I scores consistently with the DINO results. Our Style-friendly SNR sampler with $\mu = -6$ and a rank of 4 still outperforms the SD3 sampler with a rank of 32 (dotted lines).

Effectiveness Compared to Increasing Model Capacity. To demonstrate that our method is more effective than increasing model capacity, we conduct an additional experiment where we fine-tune the model using the SD3 sampler with a higher LoRA rank of 128. As shown in Tab. S2, our Style-friendly SNR sampler with a rank of 32 achieves higher DINO and CLIP-I scores compared to the SD3 sampler with a rank of 128. This indicates that focusing on the critical noise levels where styles emerge has a more significant impact than increasing the number of trainable parameters.

Trainable Parameters. To validate the importance of fine-tuning both transformer blocks of MM-DiT [3], we conduct an ablation study on SD3.5-8B, comparing the results of training LoRA adapters on only the image-transformer blocks versus training on both the image and text-transformer blocks. As shown in Tab. S3, fine-tuning both the image and text-transformer blocks leads to higher DINO and CLIP-I scores compared to fine-tuning only the image-transformer blocks, while the CLIP-T scores are identical. This indicates that including the text-transformer blocks in the fine-tuning process enhances the model's ability to learn stylistic features without compromising text alignment. These results suggest that to effectively capture new styles, it is beneficial to fine-tune both the visual and linguistic components of MM-DiT.

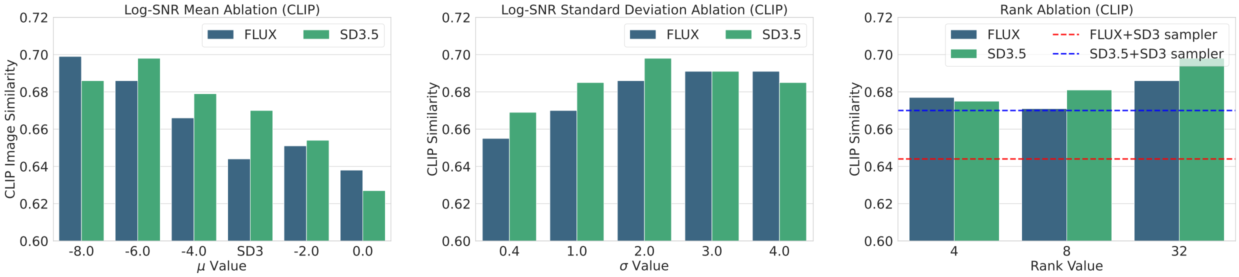
C.2. Qualitative Results

SD3.5 Samples. We extend our qualitative comparison by evaluating our Style-friendly SNR sampler using the SD3.5-8B model [21], comparing it against previous fine-tuning methods, namely the SD3 sampler [3] and DCO [10]. As shown in Fig. S5, the results are consistent with the qualitative comparisons using FLUX-dev presented in the main paper.

Additional Comparison. We further demonstrate the effectiveness of a Style-friendly SNR sampler in learning complex style templates, such as multi-panel images. As shown in Fig. S6, our method captures the given multi-panel style, generating images that closely resemble the reference. In contrast, previous fine-tuning approaches, SD3 sampler [3] and DCO [10], fail to learn the multi-panel concept, producing images without the panel structure. The offset noise [4] method attempts to reflect the style but still generates images with a single panel or fewer panels than the reference. Zero-shot approaches including IP-Adapter [23], RB-Modulation [15], and Style-Aligned [5] also attempt to generate multi-panel images but often produce outputs with structures different from the reference, as shown in Fig. S7. This highlights the capability of our method to handle challenging styles that other approaches struggle with.

Object fine-tuning using Style-friendly SNR sampler. We further evaluate whether our proposed Style-friendly SNR sampler—designed specifically for style learning—affects object-driven generation performance. As shown in Fig. S8, our Style-friendly SNR sampler and the original SNR sampler produce qualitatively similar results when fine-tuning on object references. Our approach successfully captures critical details, including shapes, colors, and prominent text or patterns (e.g., on the bowl and can). However, it occasionally omits subtle features, such as the small teeth of the monster toy. These findings support our hypothesis that distinct approaches are required for object-centric and style-driven fine-tuning; our Style-friendly sampler, while slightly suboptimal for object-centric fine-tuning, excels in capturing nuanced style characteristics.

Additional Samples. We present additional samples using the FLUX-dev [1] to demonstrate the versatility of our method. Fig. S9 shows that even when fine-tuned on square reference images, our model can generate images with different aspect ratios while maintaining the reference style.



(a) Varying μ.

(b) Varying σ.

(c) Varying LoRA Rank.

Figure S4. **SNR sampler analysis.** CLIP-I similarities with FLUX and SD3.5-8B. Dotted lines in (c) indicate the results of SD3 sampler [3].

For each prompt, we show results from two different random seeds to illustrate diversity across various aspect ratios. Fig. S10 provides additional typography samples in different aspect ratios, exhibiting our capability to produce stylized textual content.

D. Limitations and Discussions

Style Prompt Design. As shown in Fig. S11, using a different style prompt during fine-tuning can lead to emphasizing different stylistic features, such as child-like elements or background architectures (second row) instead of watercolor painting elements (first row), which may not align with the user's focus. Users should be mindful that variations in the style prompt can lead to different results. Nevertheless, our approach demonstrates effective style learning for style prompts given by the users.

Computational Cost. While fine-tuning diffusion models remains the most promising approach for achieving style alignment, it involves significant computational costs. Fine-tuning for a new style typically requires around 300 fine-tuning steps, and due to the iterative nature of diffusion models, generating a single image during inference can take several seconds. We anticipate that future work will explore applying our Style-friendly SNR sampler during the training of zero-shot models [23] or integrating it with models that offer faster inference speeds, such as Consistency Models [20] or Adversarial Diffusion Distillation models [16]. These developments could reduce both training and inference times, making style-driven generation more accessible and efficient.

E. Broader Impact

Our Style-friendly SNR sampler makes diffusion models successful in fine-tuning various style references. This advancement allows diffusion models to function effectively as digital art previewers, benefiting artists and non-expert users by simplifying the creative process. However, we note that it is important to be careful of copyright when using reference images for fine-tuning. Practitioners should ensure they have permissions to use reference images.

References

[1] BlackForestLabs. Flux. https://github.com/black-forest-labs/flux, 2024. 1, 5

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in neural information processing systems*, pages 8780–8794, 2021. 2

[3] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 1, 2, 4, 5, 6

[4] Nicholas Guttenberg. Diffusion with offset noise. https://www.crosslabs.org/blog/diffusion-with-offset-noise, 2023. 2, 5

[5] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4775–4785, 2024. 2, 5

[6] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1

[7] 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*, 2022. 1

[8] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 3

[9] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[10] Kyungmin Lee, Sangkyung Kwak, Kihyuk Sohn, and Jinwoo Shin. Direct consistency optimization for compositional text-to-image personalization. *arXiv preprint arXiv:2402.12004*, 2024. 2, 5

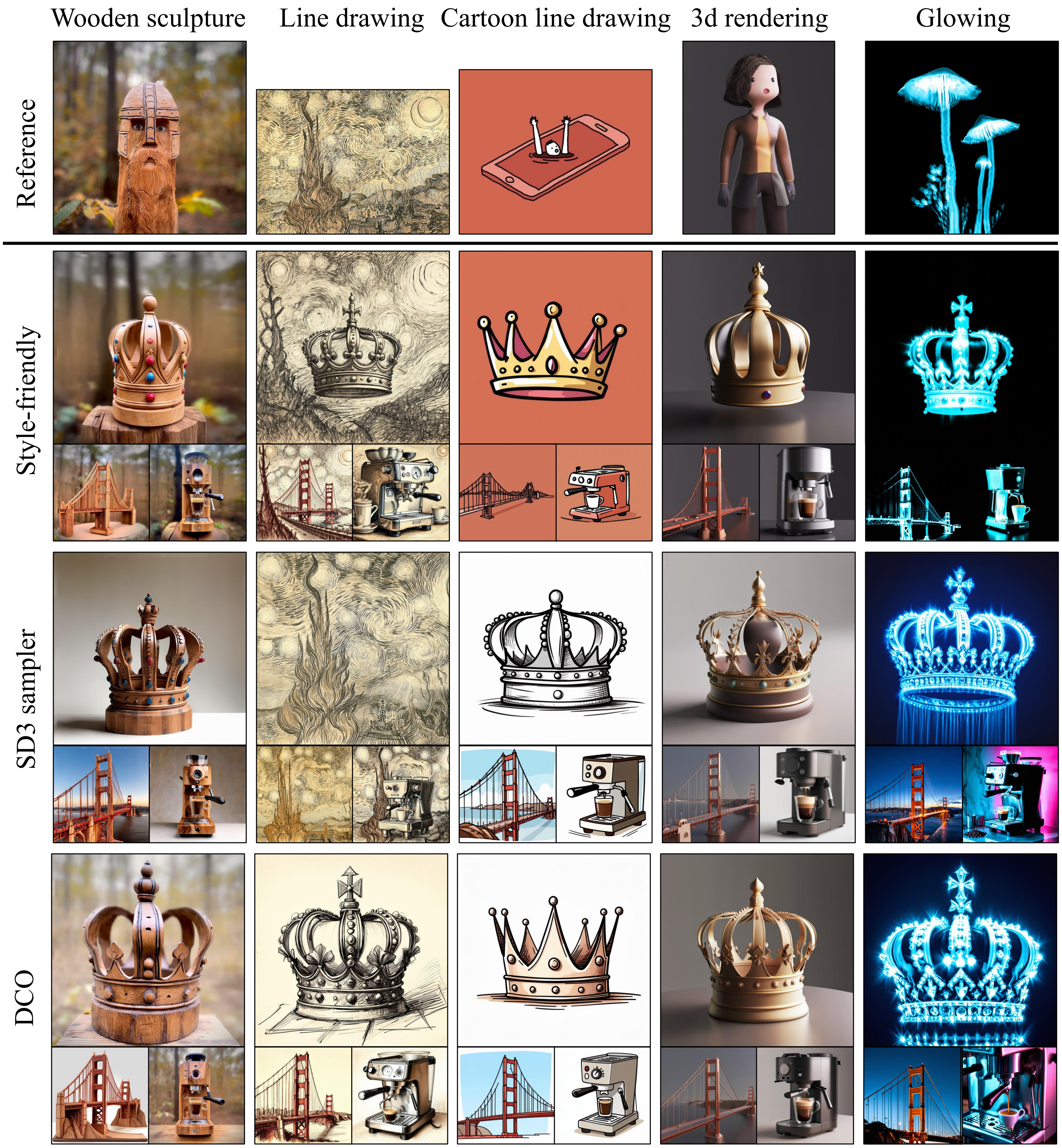


Figure S5. **Comparison of fine-tuning the SD3.5-8B.** We show 'A crown', 'A Golden Gate bridge', and 'An espresso machine' in various styles. The results with SD3.5-8B are consistent with the qualitative comparison based on FLUX-dev presented in the main paper.

[11] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023. 1

[12] Pablo Pernias, Dominic Rampas, Mats L. Richter, Christopher J. Pal, and Marc Aubreville. Wuerstchen: An efficient architecture for large-scale text-to-image diffusion models,

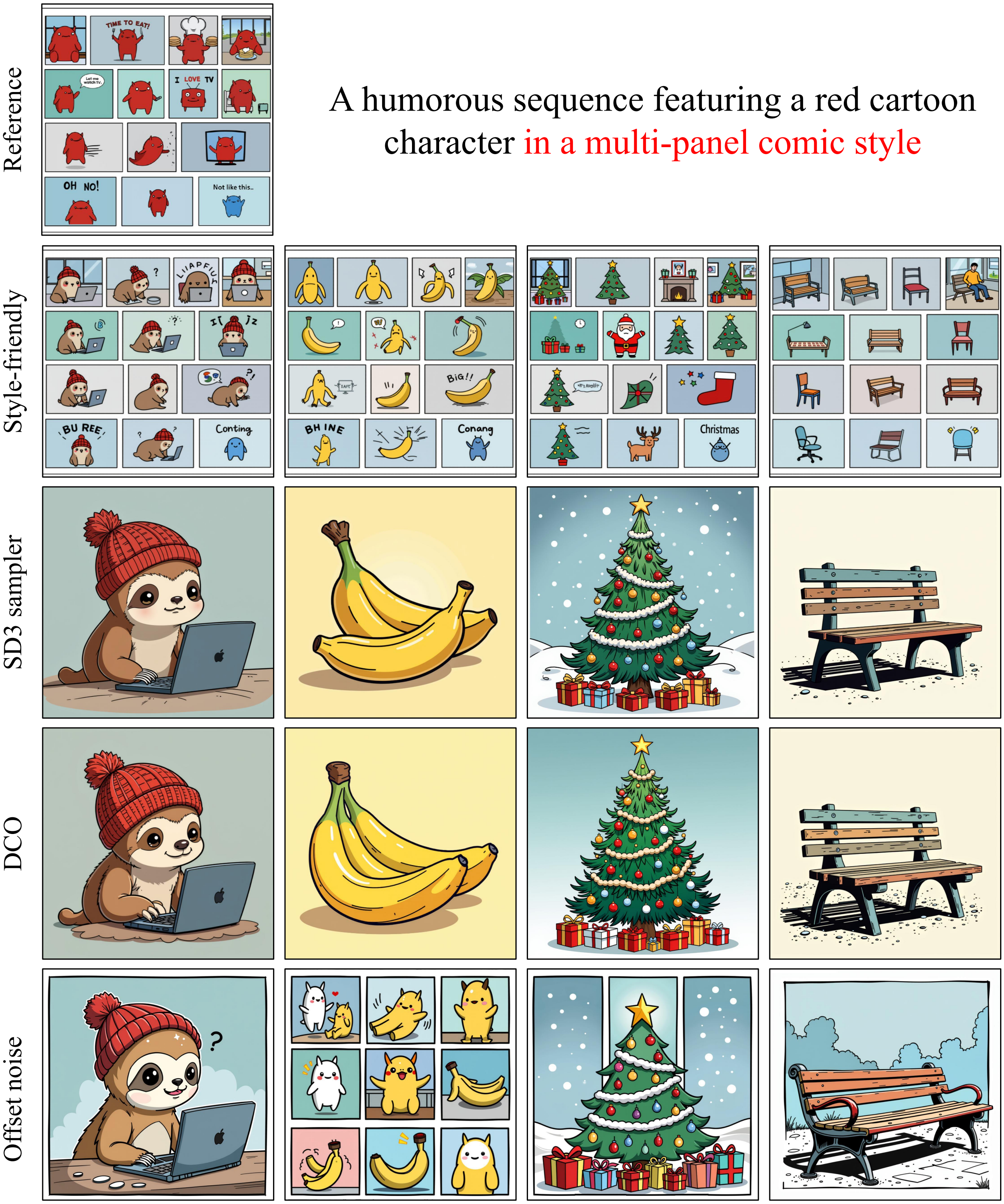


Figure S6. **Additional qualitative comparison.** Our Style-friendly approach successfully captures complex multi-panel styles, generating images that closely resemble the reference. The prompts used are “A fluffy baby sloth with a knitted hat trying to figure out a laptop, close up in {style prompt} style”, “A banana in {style prompt} style”, “A Christmas tree in {style prompt} style”, and “A bench in {style prompt} style”.

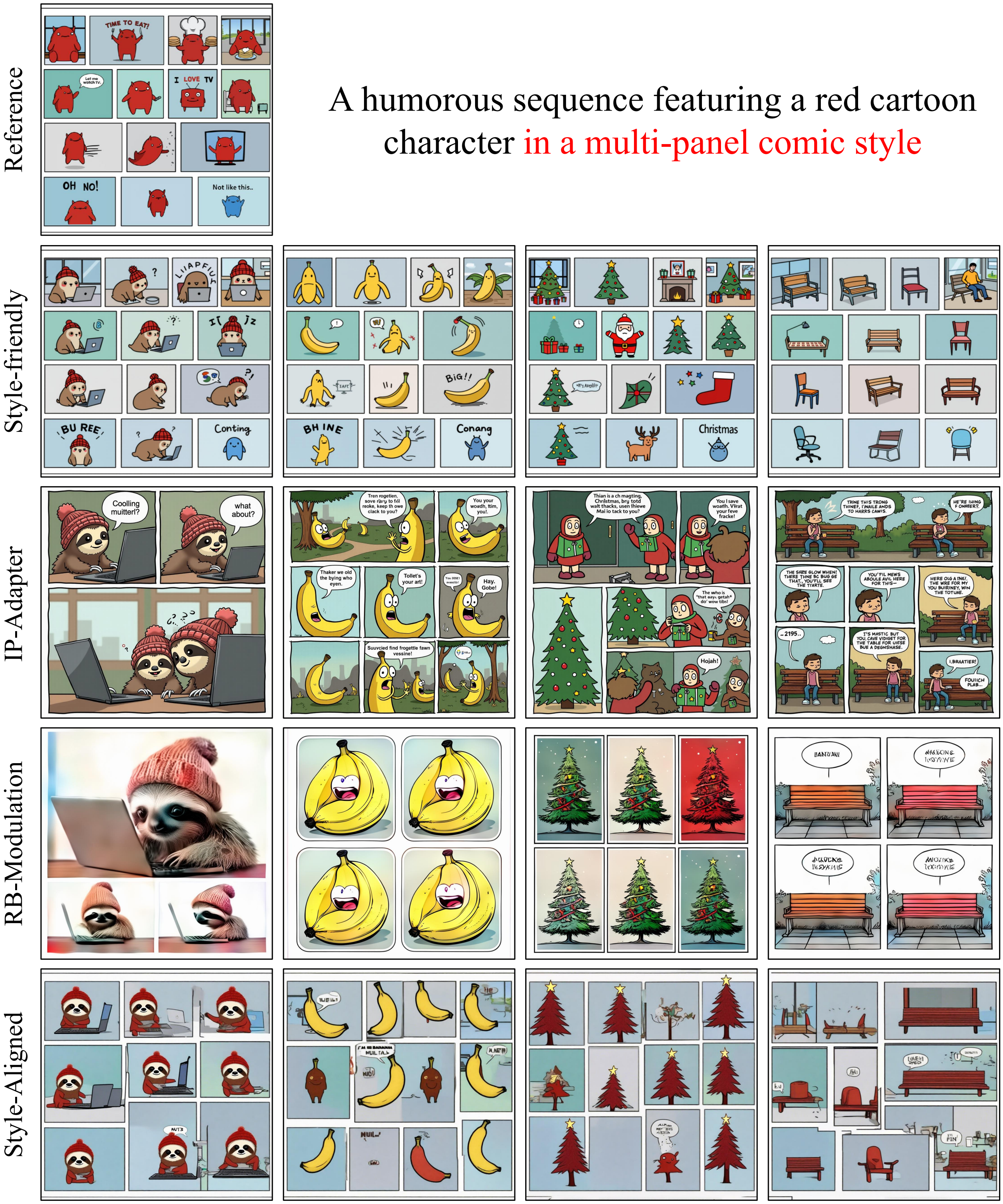


Figure S7. **Additional qualitative comparison.** Our method effectively captures the multi-panel style, whereas zero-shot methods generate images with different structures or introduce artifacts.

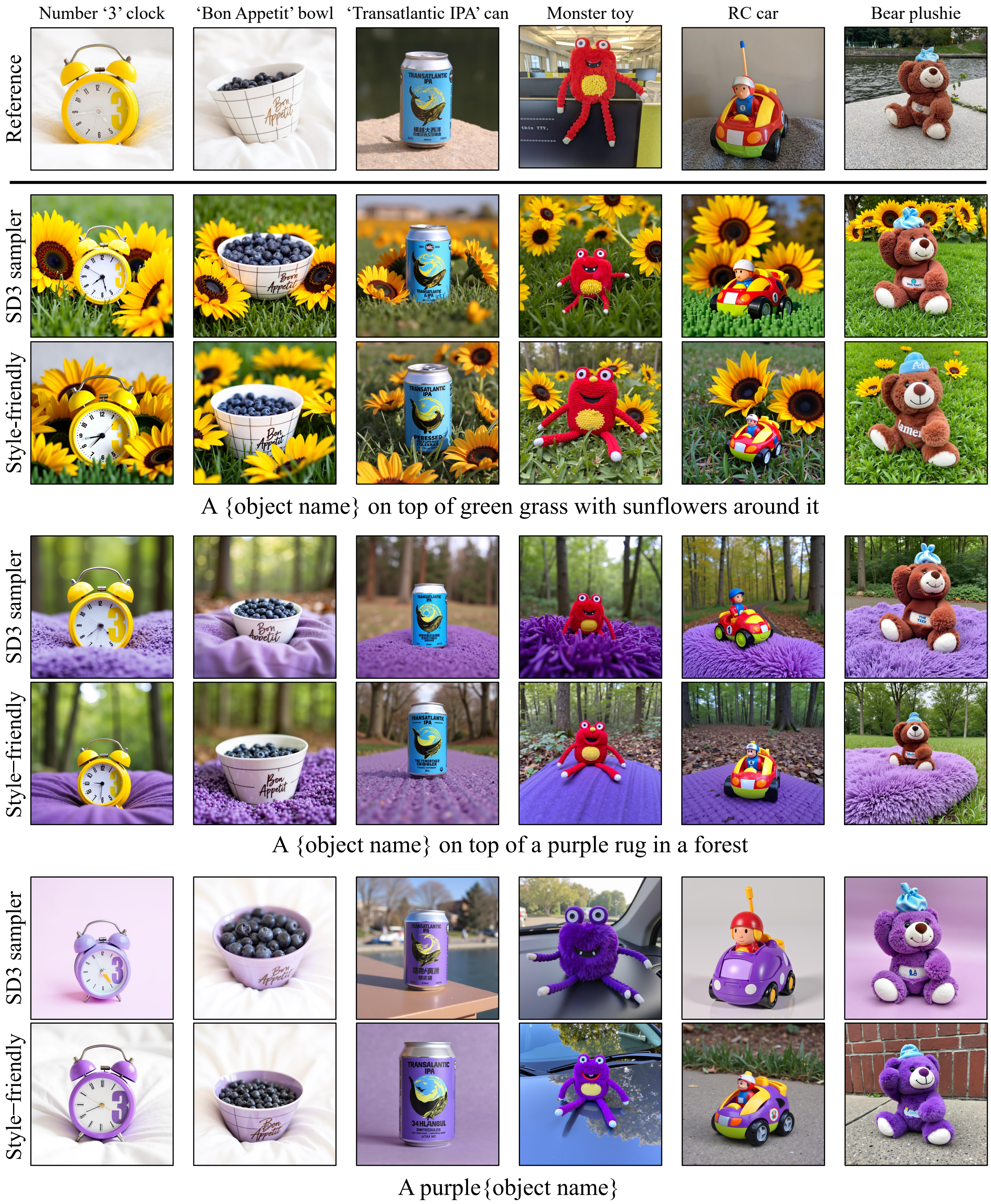


Figure S8. **Object fine-tuning comparison.** We compare our Style-friendly SNR sampler and the standard sampler on object-driven fine-tuning. Both approaches generate similar overall results, though our Style-friendly sampler occasionally misses minor details, such as the small teeth of the monster toy. Nevertheless, the Style-friendly sampler reliably captures the object’s overall shape, color, and key details such as text and patterns on the bowl and can. The object names are written at the top of the reference images.

2023. 2

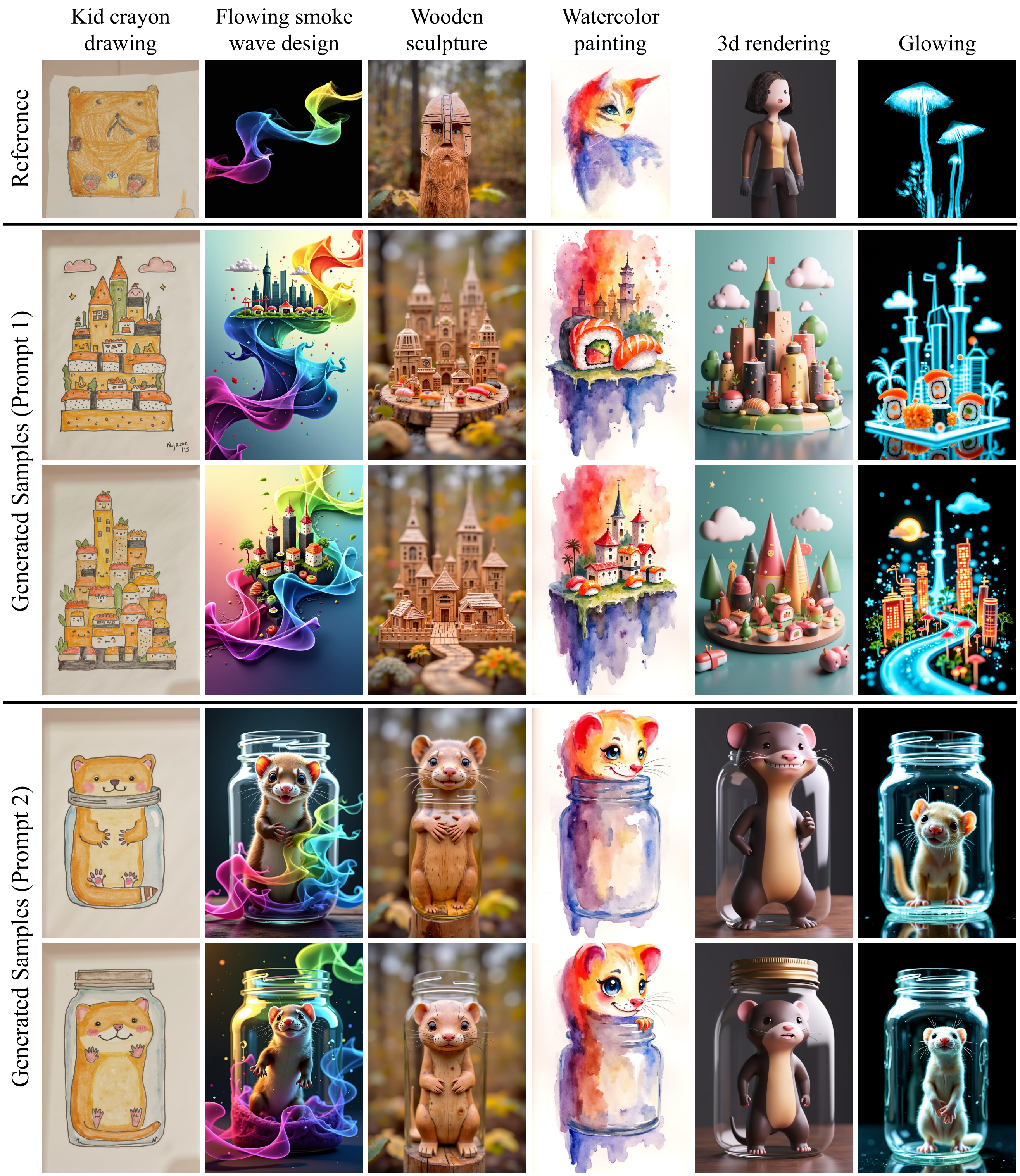

Kid crayon drawing
Flowing smoke wave design
Wooden sculpture
Watercolor painting
3d rendering
Glowing
Reference
Generated Samples (Prompt 1)
Generated Samples (Prompt 2)

Figure S9. **Additional samples.** Each row shows images generated with the same random seed at a resolution of 1216×832, using the prompts “a cute city made of sushi in {style prompt} style” and “mischievous ferret with a playful grin squeezes itself into a large glass jar, in {style prompt} style”.

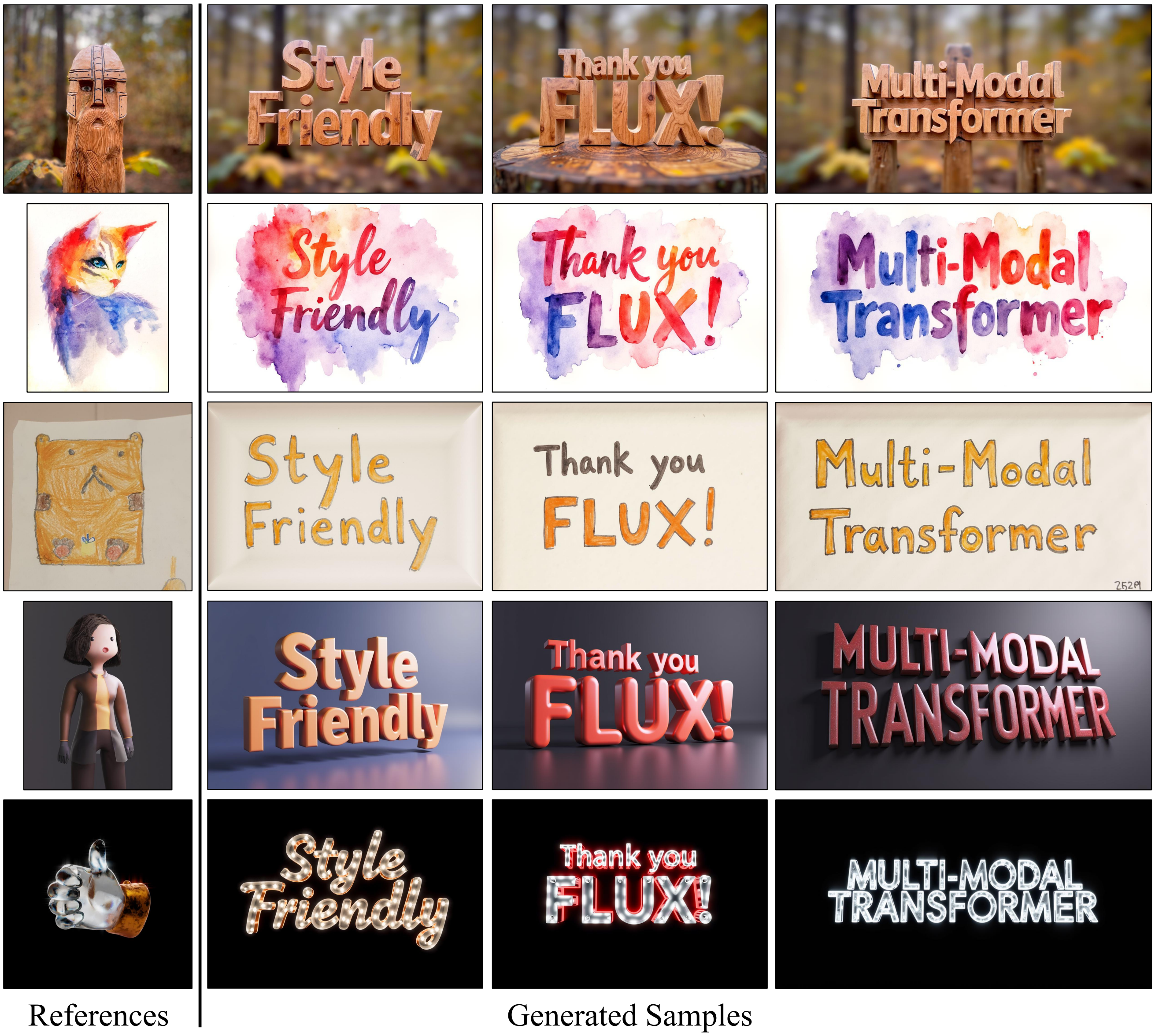
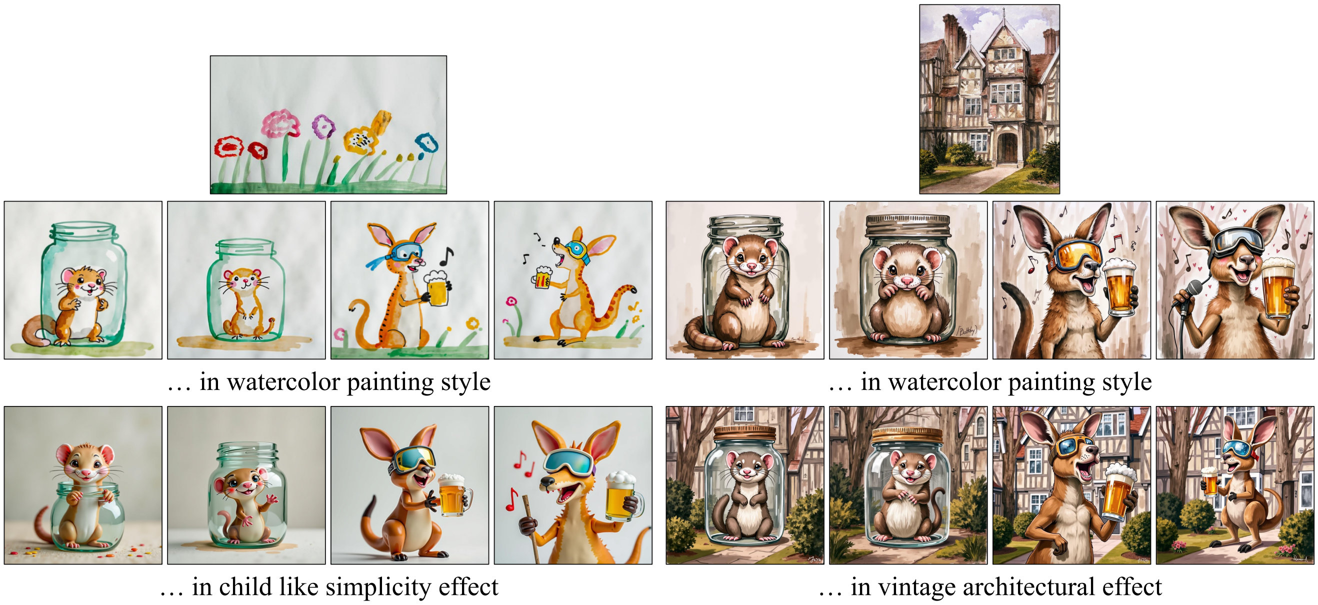


Figure S10. **Typography.** The first column shows reference images. The second and third columns display samples generated at a resolution of 832×1216, and the fourth column presents samples at 704×1408 resolution. The prompts used are "the words that says '{letters}' are written in English, in {style prompt} style", where '{letters}' represents the words synthesized in the samples.

[13] 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

[14] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[15] Litu Rout, Yujia Chen, Nataniel Ruiz, Abhishek Kumar,



… in watercolor painting style

… in watercolor painting style

… in child like simplicity effect

… in vintage architectural effect

Figure S11. **Effect of Style Prompt Design.** The first row shows images generated using style prompts from the StyleDrop paper [17] during both fine-tuning and generation. The second row shows images generated using a different style prompt during both fine-tuning and generation. Each column is generated using the same random seed. This demonstrates how varying the style prompt can lead to different stylistic elements being emphasized in the generated images.

Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Rb-modulation: Training-free personalization of diffusion models using stochastic optimal control. *arXiv preprint arXiv:2405.17401*, 2024. 2, 5

[16] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2025. 6

[17] Kihyuk Sohn, Lu Jiang, Jarred Barber, Kimin Lee, Nataniel Ruiz, Dilip Krishnan, Huiwen Chang, Yuanzhen Li, Irfan Essa, Michael Rubinstein, et al. Styledrop: Text-to-image synthesis of any style. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 4, 13

[18] Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Measuring style similarity in diffusion models. *arXiv preprint arXiv:2404.01292*, 2024. 2

[19] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2

[20] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023. 6

[21] Stability. stable-diffusion-3.5-large. https://huggingface.co/stabilityai/stable-diffusion-3.5-large, 2024. 1, 5

[22] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *The Thirteenth International Conference on Learning Representations*, 2025. 1

[23] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 2, 5, 6