SyncSDE: A Probabilistic Framework for Diffusion Synchronization

Supplementary Material

Hyunjun Lee[1*] Hyunsoo Lee[1*] Sookwan Han[1,2†]

[1]ECE, Seoul National University [2]Republic of Korea Air Force

{hjl1013, philip21, jellyheadandrew}@snu.ac.kr

A. Task-specific experimental details

In this section, we provide the experimental details of each diffusion synchronization task.

A.1. Mask-based Text-to-Image generation

We use the pretrained Stable Diffusion v2 checkpoint [11] for image generation, resulting in 512×512 resolution image. Using 10 prompts, we generate 250 images per prompt with a fixed background mask for each. KID [4] and FID [5] we use 2,000 images per prompt using the same pretrained model. We use 50 steps for DDIM [13] sampling.

A.2. Text-driven real image editing

Firstly, we explain the details of soft mask generation. Note that we follow CSG [7] to generate the soft mask $\tilde{\mathbf{B}}$ which indicates the background region of the source image. Following paragraph summarizes the procedure introduced in [7].

We extract the self-attention and cross-attention map of the source image using the pretrained Stable Diffusion [11], each denoted as $\mathbf{M}_{\text{self}} \in \mathbb{R}^{H\times W\times H\times W}$ and $\mathbf{M}_{\text{cross}} \in \mathbb{R}^{N\times H\times W}$, where N denotes the number of word tokens defined in the pretrained Stable Diffusion model. Then we generate the background mask $\tilde{\mathbf{B}}$ as follows:

$$\tilde{\mathbf{B}} = \mathbf{1} - \mathbf{M}_{\text{fg}}, \tag{24}$$

where each element of $\mathbf{M}_{\text{fg}} \in \mathbb{R}^{H\times W}$ is defined as

$$\mathbf{M}_{\text{fg}}[h,w] = \text{tr}(\mathbf{M}_{\text{self}}[h,w]\mathbf{M}_{\text{cross}}^{\top}[u]). \tag{25}$$

Note that u denotes the index of the word token corresponds to the object that we want to manipulate.

We use the pretrained Stable Diffusion v1-4 model for experiments, generate images in 512×512 resolution. Also, we use four image editing tasks for evaluation: cat $\rightarrow$ dog, dog $\rightarrow$ cat, horse $\rightarrow$ zebra, and zebra $\rightarrow$ horse. For each task, we sample 250 real images from the LAION-5B dataset [12]. To find the most relevant images for the source word (*e.g.* 'cat' in cat-to-dog task) within the dataset, we leverage CLIP retrieval [3]. The source prompt is generated using the pretrained BLIP model [8], while the target prompt is made by replacing the source word with the target word. For instance, in the 'horse $\rightarrow$ zebra' task, we swap the word 'horse' in the source prompt with 'zebra' to generate the target prompt. We use DDIM [13] sampling with 50 steps.

A.3. Wide image generation

We use the pretrained Stable Diffusion v2 checkpoint [11] for wide image generation. With four different text prompts, we generate 250 images per prompt at a resolution of 2048×512. To measure KID [4], FID [5], and CLIP-S score [10], we randomly crop the generated wide images to a resolution of 512×512. We generate 2,000 images per prompt to construct the reference image set using the same pretrained model. We use 50 steps for DDIM [13] sampling.

A.4. Ambiguous image generation

We use the pretrained DeepFloyd v1.0 checkpoint [1] for experiments, synthesizing images at 256×256 resolution. The DeepFloyd-IF model employs a two-stage sampling process for image generation. Note that we apply the proposed synchronization startegy only to the 1^{st} stage, while the 2^{nd} stage's sampling is performed without synchronization. We use 5 prompt pairs, where each pair consists of two prompts describing the semantics to be modeled in resulting ambiguous image. For each prompt pair, we set f_1 as identity mapping and choose f_2 from one of 4 visual transforms: $\pm90^\circ$ rotation, 180° rotation, vertical flip, and skew transformation. We then generate 250 images per prompt pair. In case of reference images for measuring KID [4] and FID [5], we generate 2,000 images per prompt in each prompt pair with the same pretrained model. Total 50 timesteps are used for DDIM [13] sampling.

*Equal Contributions.

†Project Lead.

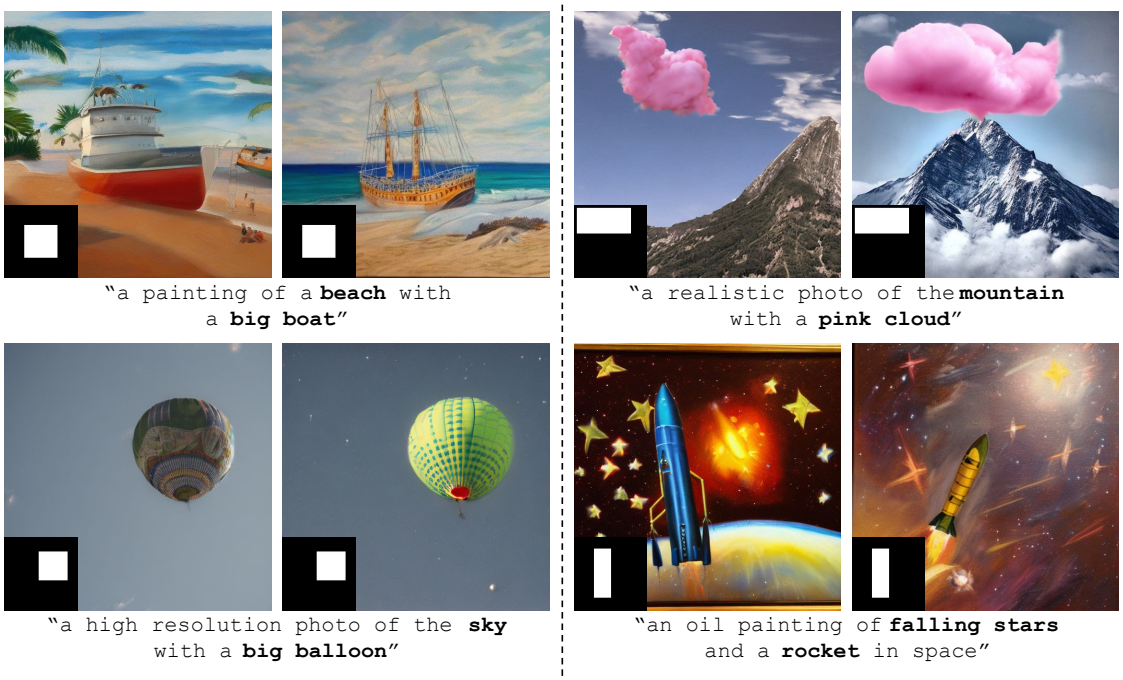

Figure 10. **Additional qualitative results of mask-based T2I generation.** SyncSDE shows strong performance on mask-based T2I generation task. We use the pretrained Stable Diffusion [11] for image generation.

A.5. 3D mesh texturing

We use the pretrained depth-conditioned ControlNet v1-1 [14] for mesh texturing. Using 6 meshes and a single prompt for each mesh, we generate 100 textures per mesh. Each generated texture is projected onto a fixed single view, resulting in a 768×768 resolution RGB image. To generate reference images, we use the same pretrained model and sample 2,000 images per prompt using the equivalent mesh map as depth condition. In addition, following SyncMVD [9] and SyncTweedies [6], we use the self-attention modification technique proposed in [9] along with Voronoi Diagram-guided filling [2]. We use 30 steps for DDIM [13] sampling. Like SyncTweedies, we do not use diffusion synchronization during the last 20% of the sampling steps.

Table 6. **Comparison of computational overhead between SyncSDE and SyncTweedies [6].**

Method	Time (s/image)	GPU memory (GB)
SyncTweedies [6]	7.721	2.44
SyncSDE (Ours)	5.664	2.78

B. Computational overhead

We analyze the computational overhead in terms of both time and GPU memory required to generate a single image. The measured computational overhead of the proposed method and SyncTweedies [6] is reported in Table 6. We use a single NVIDIA RTX 3090 GPU for measurement. Notably, SyncSDE exhibits a comparable computational overhead to SyncTweedies.

C. Additional qualitative results

We visualize additional qualitative results of SyncSDE in Figure 10, 11, 12, 13, and 14. As shown in the figures, SyncSDE shows outstanding performance in multiple image generation tasks, including mask-based T2I generation, text-driven real image editing, wide image generation, ambiguous image generation, and 3D mesh texturing. The experimental results demonstrate that the proposed method successfully models the correlation between multiple diffusion trajectories, thus smoothly blending the generated patches.

D. Limitations and social impacts

Since our method uses a pretrained text-to-image diffusion model [1, 11, 14], the proposed method may result in sub-optimal outcomes depending on the pretrained backbone model. For instance, due to the limitations of the pretrained diffusion model, it may struggle to synthesize images with complex structures or multiple fine details. Furthermore, the proposed method may generate harmful images due to the shortcomings of the pretrained diffusion model.

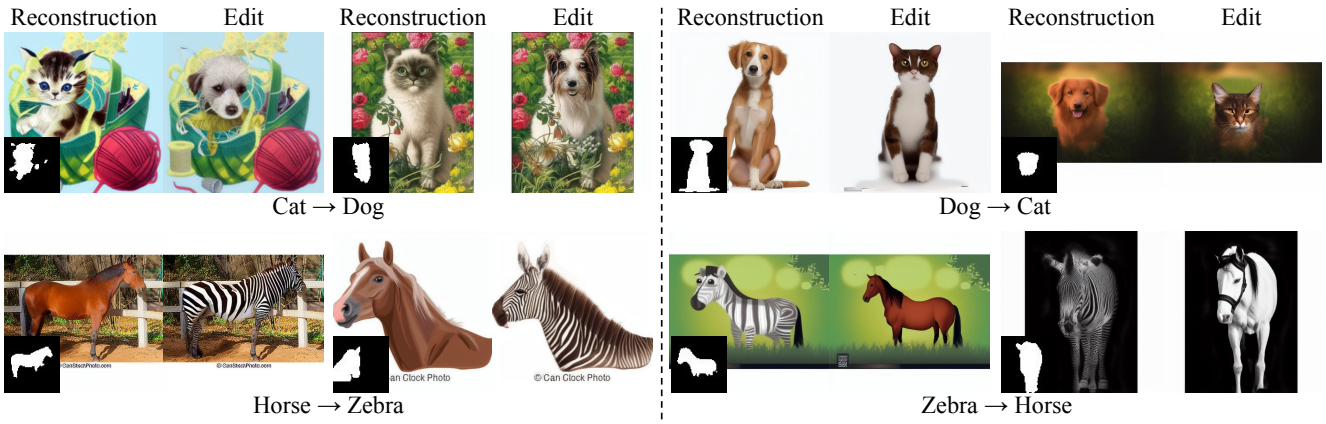

Figure 11. **Additional qualitative results of text-driven real image editing.** We edit the real images sampled from the LAION-5B dataset [12] by leveraging SyncSDE combined with the pretrained Stable Diffusion [11]. We also visualize the foreground region defined by the generated mask.

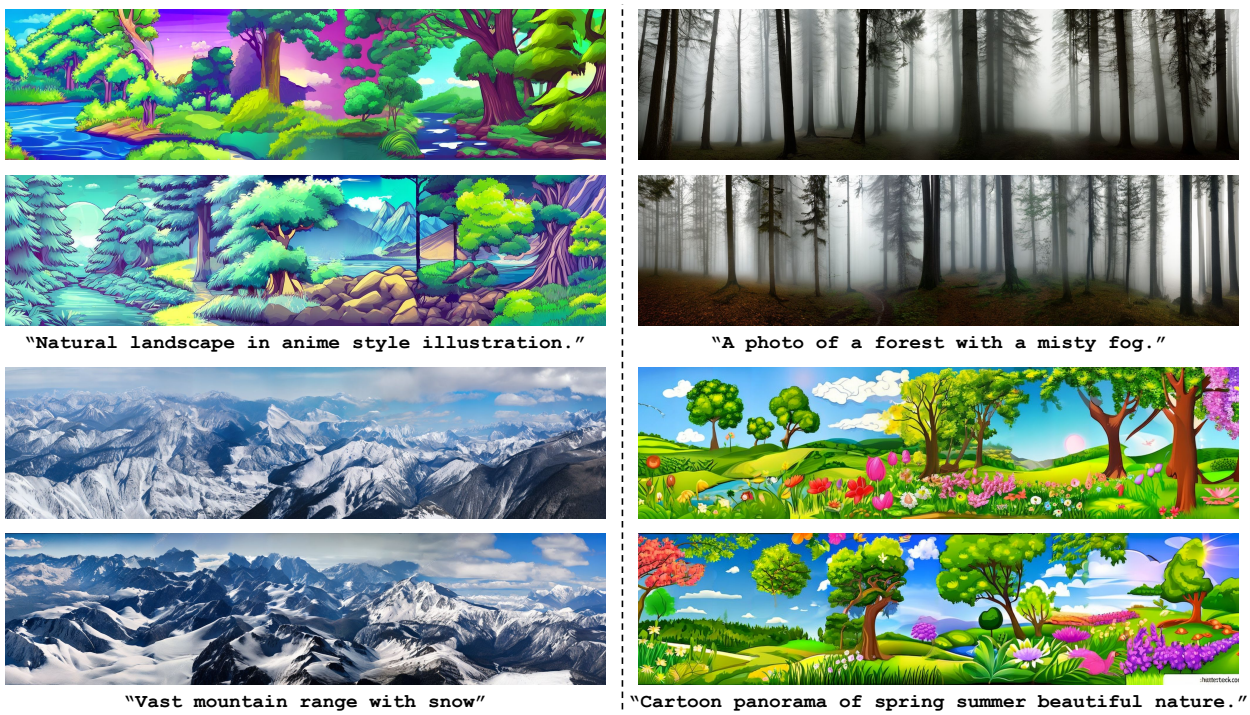

Figure 12. **Additional qualitative results of wide image generation.** We visualize wide images generated by SyncSDE using the pretrained Stable Diffusion [11] for image generation.

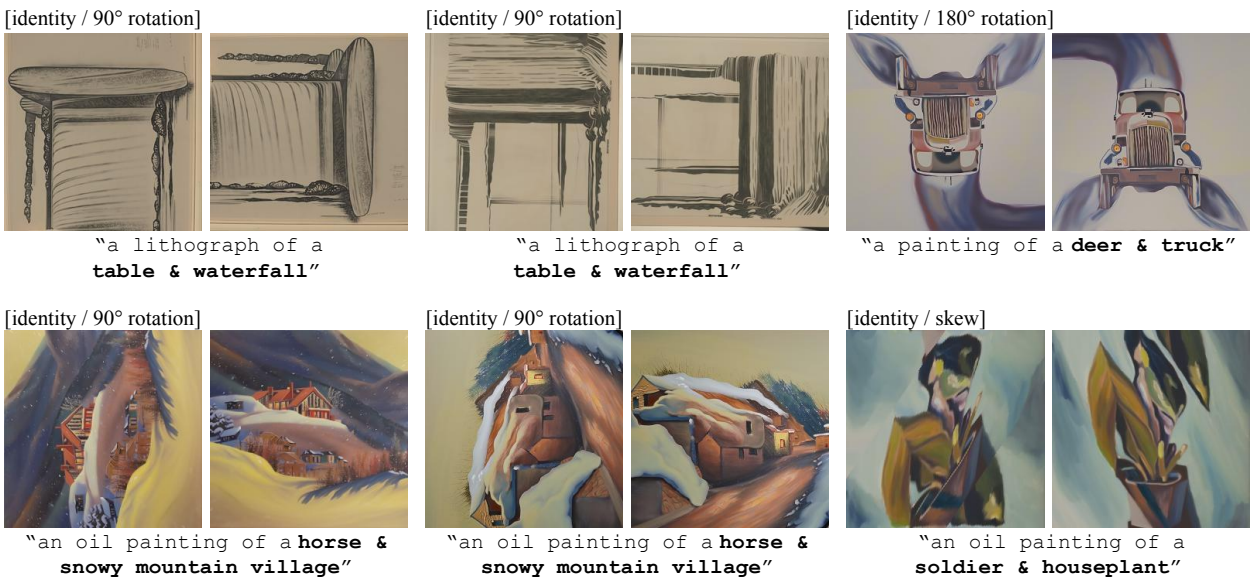

Figure 13. **Additional qualitative results of ambiguous image generation.** Using the pretrained Deepfloyd-IF model [1], we generate ambiguous image with various prompt pairs and visual transformations. SyncSDE generates high-quality ambiguous images.

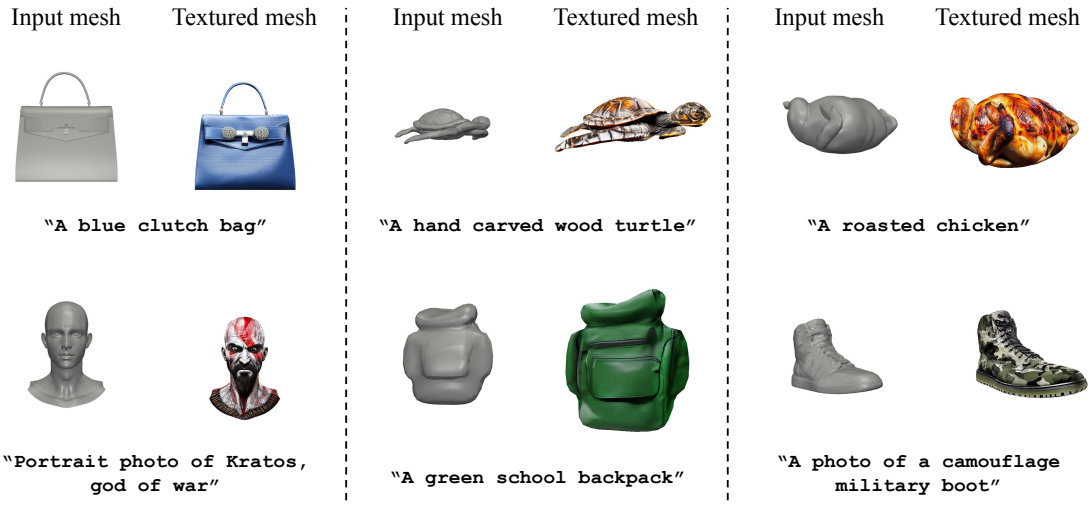

Figure 14. **Additional qualitative results of 3D mesh texturing.** We use the pretrained depth-conditioned ControlNet [14] for mesh texturing. Given an input mesh and the text prompt, SyncSDE generates remarkable texture images.

References

[1] DeepFloyd Lab at StabilityAI. Deepfloyd if. https://www.deepfloyd.ai/deepfloyd-if, 2023. 1, 2, 4

[2] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *CSUR*, 1991. 2

[3] Romain Beaumont. Clip retrieval: Easily compute clip embeddings and build a clip retrieval system with them. https://github.com/rom1504/clip-retrieval, 2022. 1

[4] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *ICLR*, 2018. 1

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 1

[6] Kim Jaihoon, Koo Juil, Yeo Kyeongmin, and Sung Minhyuk. Synctweedies: A general generative framework based on synchronized diffusions. *NeurIPS*, 2024. 2

[7] Hyunsoo Lee, Minsoo Kang, and Bohyung Han. Conditional score guidance for text-driven image-to-image translation. *NeurIPS*, 2023. 1

[8] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 1

[9] Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. In *SIGGRAPH Asia*, 2024. 2

[10] 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 *ICML*, 2021. 1

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 3

[12] 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. *NeurIPS*, 2022. 1, 3

[13] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021. 1, 2

[14] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 4