Attention Distillation: A Unified Approach to Visual Characteristics Transfer

Supplementary Material

A. Algorithm

We build our method on the pretrained Stable Diffusion models. Algorithm 1, using style transfer as an example, outlines our content-preserving optimization approach with attention distillation loss. For attention distillation guided sampling, we take style-specific text-to-image generation as an example and describe our approach in Algorithm 2. We denote the encoder and decoder of VAE as $\mathcal{E}(\cdot)$ and $\mathcal{D}(\cdot)$, respectively, and use $\epsilon_\theta(\cdot)$ to represent the denoising network. In Algorithm 2, $\mathrm{Sampling}(\cdot)$ refers to a diffusion sampling step from z_t to z_{t-1}, and $\mathrm{AdaIN}(\cdot,\cdot)$ [27] refers to modulate the variance and mean of the features to boost stylization.

B. Implementation Details

We implemented our approach using the PyTorch framework, applying mixed precision to save time and memory costs. For style-specific text-to-image generation, we use SDXL [46]; for other tasks, we employ Stable Diffusion v1.5 [48]. Following recent works [7, 29, 70], we extract attention features from the last six self-attention layers of U-Net to compute attention distillation loss. For comparison, we use the publicly available implementations of all baseline methods and adhere to their suggested configurations. All experiments are conducted on a single NVIDIA RTX 6000 Ada GPU. We use a fixed learning rate (0.05) for the Adam optimizer, except for style-specific text-to-image generation (0.015). In the following, we specify the detailed configurations for each task.

Style/Appearance Transfer. We initialize the target latent using the content/structure image. The content loss is computed with the Q features from the last 6 self-attention layers. The content loss weight, λ, is set to 0.25 for style transfer and 0.2 for appearance transfer, respectively. By default, We optimize the target latent over 200 iterations. All experiments are conducted to generate images at a resolution of 512x512. The time to synthesize an image takes about 30 seconds, with our optimization in latent space.

Style-Specific Text-to-Image Generation. We generate images at a resolution of 1024x1024 using SDXL. The sampling is conducted over 50 steps using DDIM sampling, with a scale set to 7 for classifier-free guidance. At each sampling step, We perform 2 iterations of latent optimization utilizing attention distillation loss. The whole process takes no more than 30 seconds. The learning rate of the Adam optimizer is set to 0.015 by default.

Controlled Texture Synthesis. For the mask-controlled texture synthesis, images are resized to resolution 512×512, and synthesized in the optimization manner. The optimization performs 200 iterations by default. We adopted the same initialization strategy as GCD [69], where we fill the target segmentation map with random pixels drawn from the semantically corresponding region of the source texture. However, the low spatial resolution of features from U-Net makes the Masked AD loss inadequate for precise spatial control, as shown in Fig. 15. To address this, we utilize the Q features from the initialization image to compute the content loss with a content weight λ of 0.15. Introducing content loss leads to precise spatial alignment without compromising texture quality. For the layout control task as Self-Rectification [70], we directly use the color layout as the content image to compute content loss.

Texture Expansion. In this task, the example textures are resized to 512×512. The results are generated using attention distillation guided sampling for efficiency. We use MultiDiffusion [5] to synthesize ultra-high resolution textures, achieving remarkable results; see an example of size 4096×4096 in Fig. 26. The sampling is conducted over 50 steps using DDIM sampling without classifier-free guidance. At each sampling step, We perform 3 iterations of latent optimization utilizing our attention distillation loss.

C. Additional Experiments

Time Efficiency. For texture synthesis, either optimization or sampling can be utilized. We record the time consumed by different methods (excluding the time for model loading, compilation, and image encoding/decoding). Specifically, the sampling method employs the DDIM sampler with 50 steps without classifier-free guidance. The Adam optimizer is set with a fixed learning rate of 0.05 for both methods. Typically, non-stationary textures require more iterations to produce a reasonable spatial structure. The detailed results are presented in Table 1 and Fig. 14.

A DEEP COMPARISON between optimization and sampling with attention distillation. The primary distinction between our optimization-based and sampling-based approaches with attention distillation lies in the nature of the extracted features. As illustrated in Algorithms 1 and 2, sampling-based methods extract features from

Algorithm 1 Content-preserving Optimization (For Style and Appearance Transfer)

Input: Style image I^s, content image I^c, learning rate η, content loss weight λ.
Output: Optimized image I.
$z^s, z^c \leftarrow \mathcal{E}(I^s), \mathcal{E}(I^c)$ ▷ Convert the input images to latent space
Initialize $z \leftarrow z^c$ ▷ Start with the content latents
for $t = T, T-1, ..., 1$ **do**
  $\{Q_c, K_c, V_c\} \leftarrow \epsilon_\theta(z^c, t, \emptyset)$ ▷ Extract self-attention features from the UNet
  $\{Q_s, K_s, V_s\} \leftarrow \epsilon_\theta(z^s, t, \emptyset)$
  $\{Q, K, V\} \leftarrow \epsilon_\theta(z, t, \emptyset)$
  $\mathcal{L}_{\text{content}} = \|Q - Q_c\|_1$ ▷ Calculate the content loss
  $\mathcal{L}_{\text{AD}} = \|\text{Self-Attn}(Q, K, V) - \text{Self-Attn}(Q, K_s, V_s)\|_1$ ▷ Calculate the style loss
  $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{AD}} + \lambda \mathcal{L}_{\text{content}}$ ▷ Total loss
  $z \leftarrow z - \eta \nabla_z \mathcal{L}_{\text{total}}$ ▷ Gradient descent step
end for
$I \leftarrow \mathcal{D}(z)$ ▷ Decode the latents to image space
Return: I.

Algorithm 2 Attention Distillation Guided Sampling (For Style-specific Text-to-Image Generation)

Input: Style image I^s, text prompt y, learning rate η, optimization steps M.
Output: Generated image I.
$z^s \leftarrow \mathcal{E}(I^s)$ ▷ Convert the input images to latent space
Initialize $z_T \sim \mathcal{N}(0, 1)$ ▷ Start with random noise
for $t = T, t-1, ..., 1$ **do**
  $z_{t-1} \leftarrow \text{Sampling}(z_t, t, \epsilon_\theta(z_t, t, y))$ ▷ Diffusion Sampling
  $z^s_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} z^s + \sqrt{1 - \bar{\alpha}_{t-1}} \epsilon, \epsilon \sim \mathcal{N}(0, 1)$ ▷ Add noise to the style image latents
  $\{Q, K_s, V_s\} \leftarrow \epsilon_\theta(z^s_{t-1}, t-1, \emptyset)$ ▷ Extract self-attention features from the UNet
  $z_{t-1} = \text{AdaIN}(z_{t-1}, z^s_{t-1})$ ▷ Modulate the variance and mean
  for $m = 1, ..., M$ **do**
    $\{Q, K, V\} \leftarrow \epsilon_\theta(z_{t-1}, t-1, \emptyset)$
    $\mathcal{L}_{\text{AD}} = \|\text{Self-Attn}(Q, K, V) - \text{Self-Attn}(Q, K_s, V_s)\|_1$ ▷ Calculate the style loss
    $z_{t-1} \leftarrow z_{t-1} - \eta \nabla_{z_{t-1}} \mathcal{L}_{\text{AD}}$ ▷ Gradient descent step
  end for
end for
$I \leftarrow \mathcal{D}(z_0)$ ▷ Decode the latents to image space
Return: I

Figure 14. Comparison between optimization-based and sampling-based approaches with attention distillation.

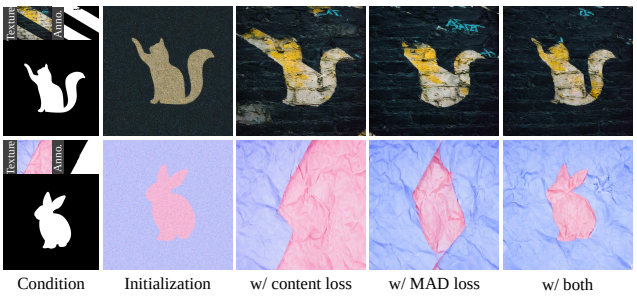


Figure 15. Ablation of losses used for controlled texture synthesis.

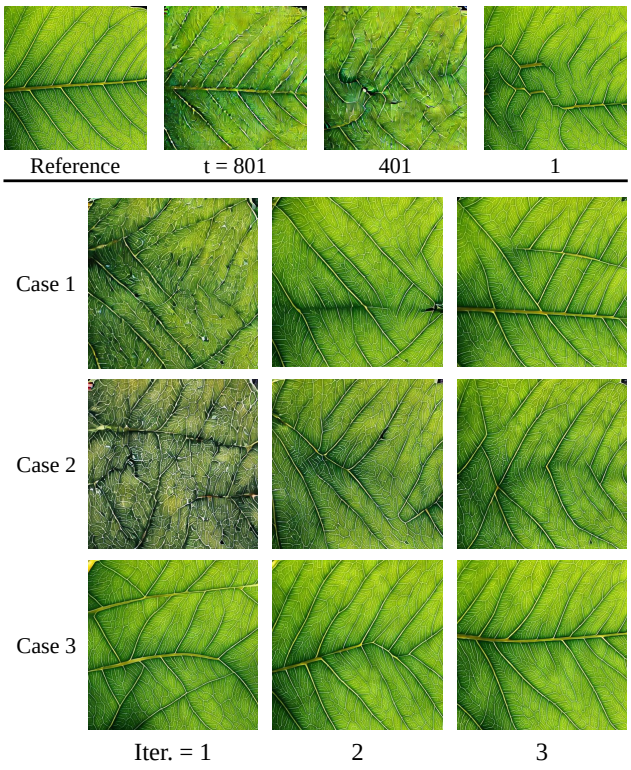


Figure 16. Digging deeper into the difference between our optimization and sampling-based methods. Top: optimization using differently fixed timesteps (fixed during optimization). Bottom: optimization with clean latents (Case 1), optimization with noised latents (Case 2), and sampling with noised latents (Case 3). For a fair comparison, we add the iteration number at each timestep for the optimization-based method (Case 1 & 2). See text for details.

stochastically inverted images, where scheduled noise relating to timesteps is added, which prevents the latents from being optimized outside the data distribution. In contrast, our optimization-based method extracts features from clean images (*i.e.*, z_0 encoded by VAE encoder). Experimental results reveal that by adjusting the timestep t, it is possible to extract features at varying levels of granularity, ranging from coarse to fine. As shown in Fig. 16 top, features

Table 1. Time efficiency of our optimization-based and sampling-based approaches using Stable Diffusion v1.5. The sample-based approach performed a total of 50 sampling steps.

	Optimization-based			Sampling-based		
Iterations	100	200	300	1	2	3
Run Time	10 s	21 s	32 s	7 s	10 s	13 s
GPU Memory	4 GB					

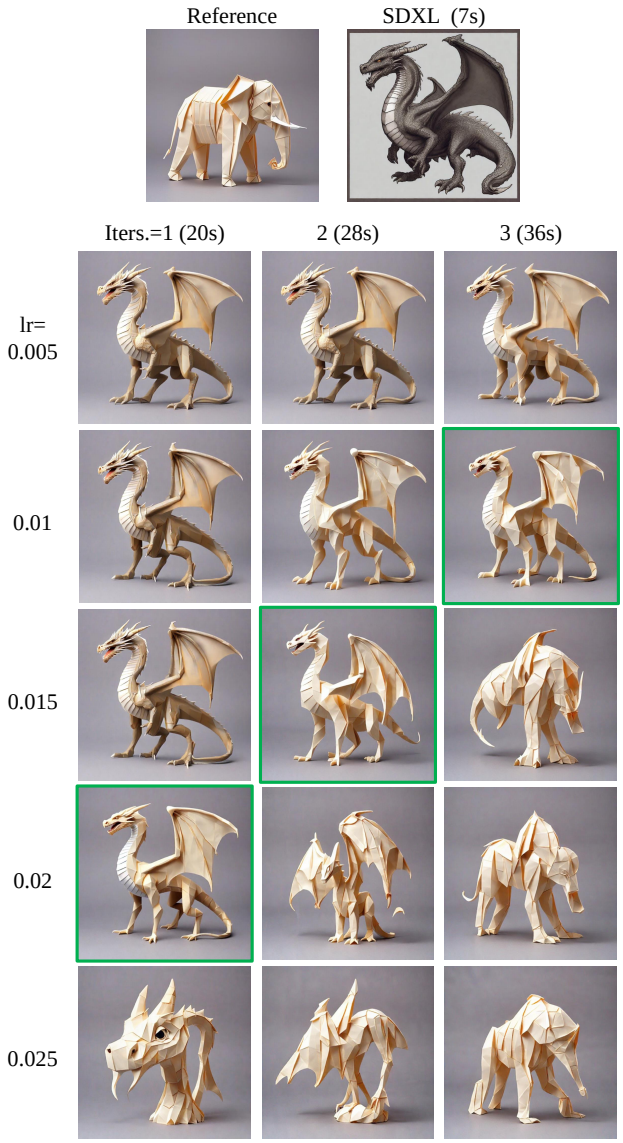


Figure 17. Impact of different learning rates and optimization iterations in style-specific text-to-image generation.

extracted from different timesteps were used to compute the AD loss to optimize the same Gaussian noise. Using the Adam optimizer with a learning rate of 0.05 and 200 iterations, the results indicate that features correspond-

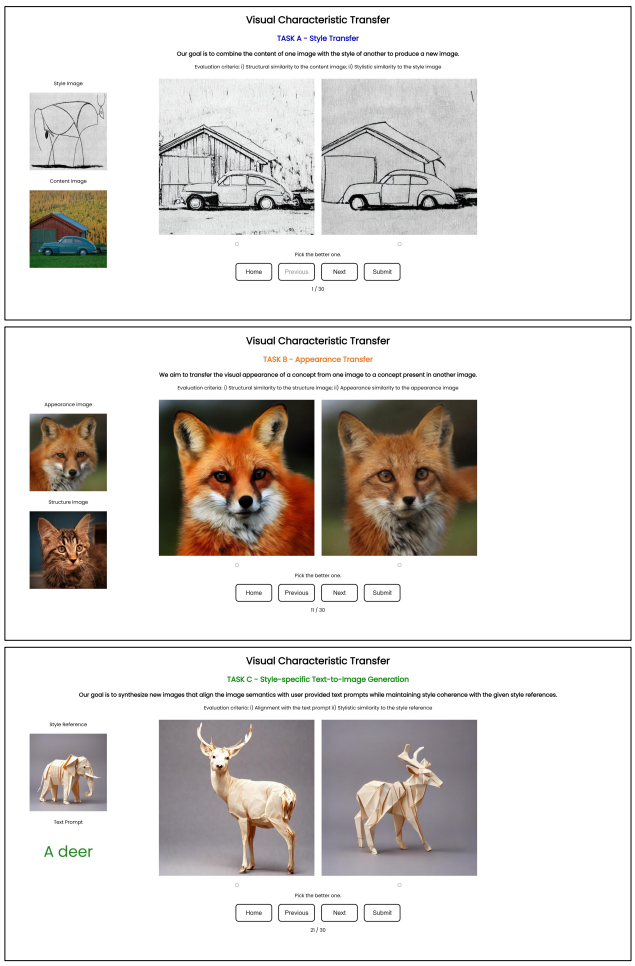
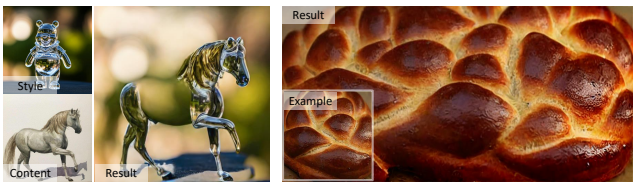


Figure 18. **User study interface.**

ing to larger timesteps focus on coarse structures, whereas those from small timesteps focus on fine details, demonstrating the necessity of linearly decreasing the timestep in our optimization-based method, as described in Sec.3.2 of our main paper.

To further investigate the differences between these two approaches, we designed three experimental cases for texture synthesis. Case 1 involves using features extracted from clean image latents to compute the AD loss for optimization. Case 2 uses features extracted from noisy latents for the same purpose. Case 3 also employs features from noisy latents but optimizes the latent after denoising with the UNet for each timestep, *i.e.*, our AD-guided sampling method. In these experiments, the same Gaussian noise was used as the initial latent, the Adam optimizer with a learning rate of 0.05 was employed, and the number of steps was set to 50. As shown in the bottom of Figure 16, the comparison between Cases 1 and 2 reveals that Case 2 produces noisier results and converges much more slowly. More importantly, the comparison between Cases 2 and 3 demonstrates



(a) (b)

Figure 19. Limitations of our method.

that our guided sampling (or, equivalently, optimizing the denoising UNet-sampled results with AD loss) significantly improves the quality and speed of texture synthesis.

Impact of hyperparameters on Style-Specific T2I Generation. We study the impact of two hyperparameters, optimization iteration number in sampling, and learning rate on style-specific T2I generation. As shown in Fig. 17, a lower learning rate or fewer optimization iterations results in insufficient stylization, while increasing the learning rate or the number of optimization iterations can lead to a loss of semantic structure derived from text prompts. According to this study, we set the number of optimization iterations to 2 and the learning rate to 0.015 as default values, balancing image quality, text alignment, and time.

D. Details of User Study

We conduct a user study on three transfer tasks, selecting two competitors for each. Specifically, we compare StyleID [10] and StyTR^2 [12] for style transfer, Cross-image Attention [2] and SpliceViT [57] for appearance transfer, and InstantStyle [59] and Visual Style Prompting [29] for style-specific text-to-image generation. For each task, the user interface, shown in Fig. 18, randomly presents a set of results from our pool, displaying our method's generated results alongside those of one competitor in the center of the screen side-by-side. Reference images or prompts are provided on the left, with a summary of the evaluation criteria at the top of the screen. Users are asked to pick the better one. The criteria for each task are summarized as follows:

Style transfer: i) structural similarity to the content image, and ii) stylistic similarity to the style image.

Appearance transfer: i) structural similarity to the structure image, and ii) appearance similarity to the appearance image.

Style-specific text-to-image generation: i) semantic alignment with the text prompt, and ii) stylistic similarity to the style reference.

E. Limitation and Discussion

While we have demonstrated the effectiveness of our attention distillation loss across a wide range of visual characteristic transfer tasks—such as artistic style and appearance transfer, style-specific text-to-image generation, and texture synthesis—several limitations should be noted. First, we observed that the results of texture expansion occasionally exhibit oversaturated colors. This issue arises because the AD loss does not explicitly constrain the consistency of the data distribution. Instead, it relies on the model's understanding of the reference image to reassemble visual elements. When the resolution of the generated image exceeds the model's training scope, the aggregation process may produce suboptimal results. Second, in style and appearance transfer tasks, the AD loss depends on the model's ability to establish semantic correspondences based on its understanding of images. When the content of two images differs significantly, the model's limitations may lead to incorrect semantic matches, negatively impacting the final output. See Fig. 19 for two examples.

F. Additional Results

Finally, in the below figures, we provide additional results:

(1) In Figs. 20 and 21, we display additional results of creative, text-guided generation with style-specific guidance.

(2) In Fig. 22, we show more style transfer outcomes on diverse content and style examples.

(3) In Fig. 23, we present the comparison on unconditioned texture synthesis to showcase the texture understanding capabilities of our attention distillation loss. We apply both optimization-based and sampling-based approaches with our method and compare them against state-of-the-art methods, including Self-Rectification [70], GCD [69], GPDM [16], and SWD [24].

(4) In Figs. 24 and 25, we present the additional results of stationary and non-stationary texture synthesis and expansion, all achieved through our guided-sampling approach.

(5) Finally, in Fig. 26, we demonstrate an extreme texture expansion by generating a high-resolution image in size 4096×4096 using a 512×512 example.

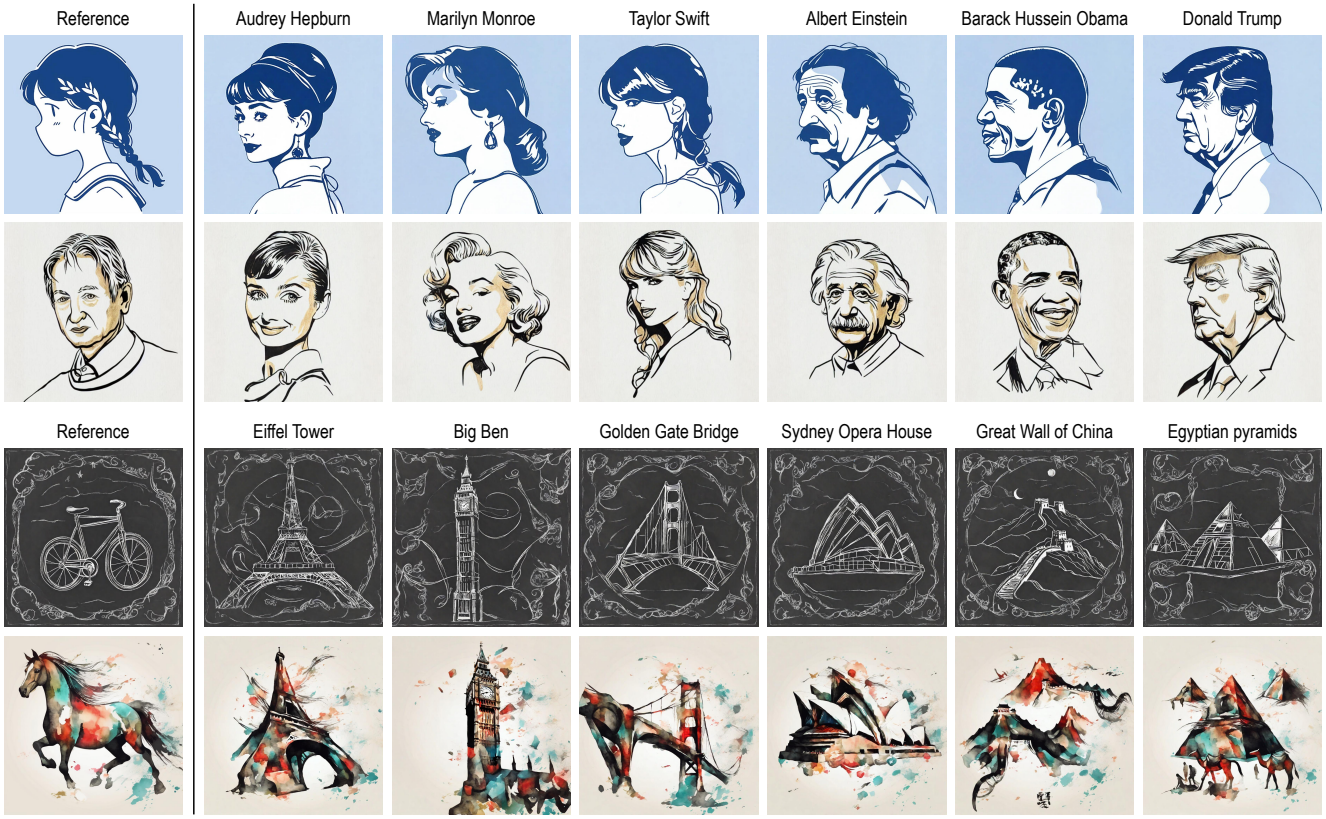


Figure 20. Additional results of our approach on style-specific text-to-image generation.

Figure 21. Additional results of our approach on style-specific text-to-image generation.

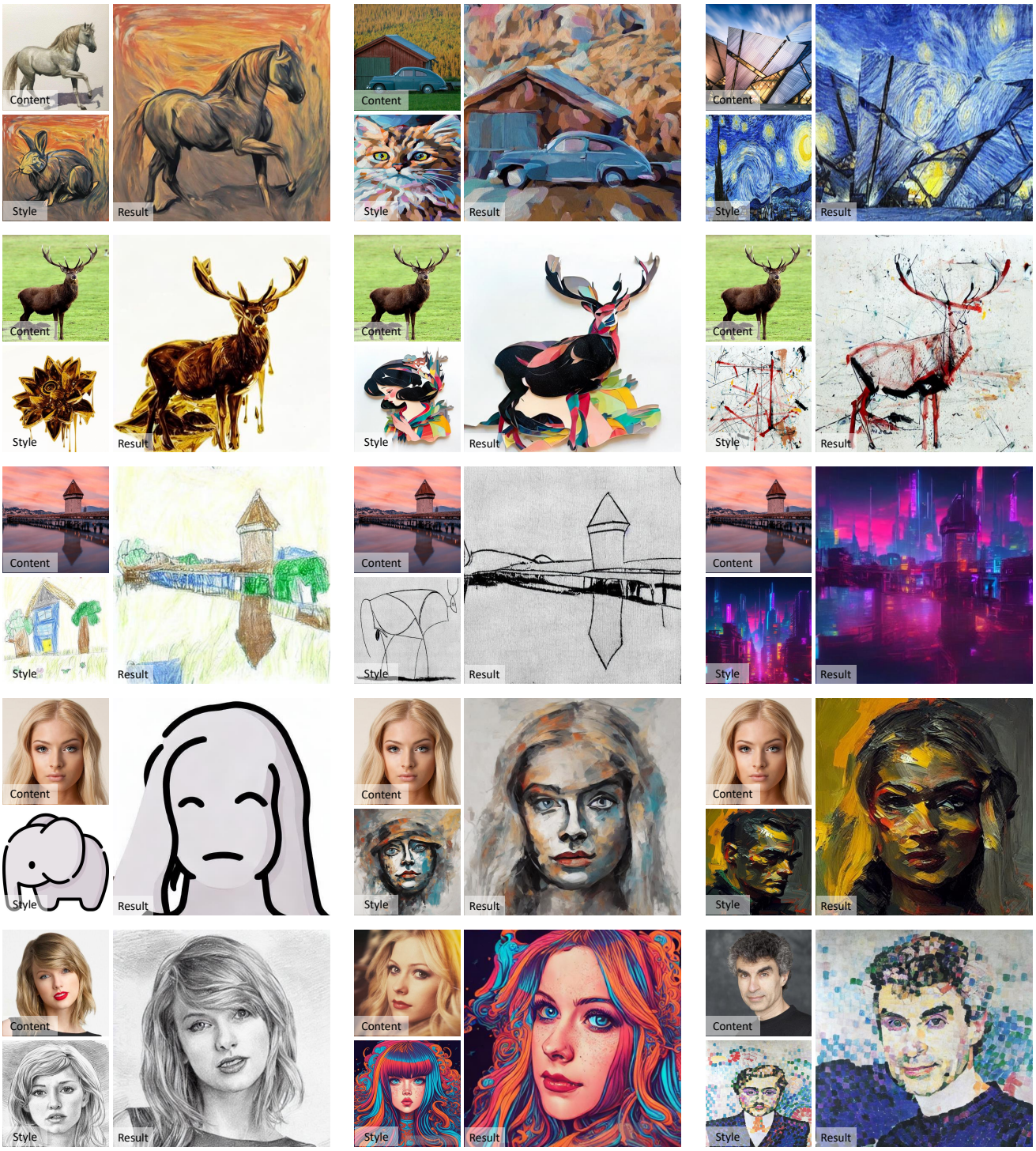


Figure 22. Additional results of our approach on artistic style transfer.

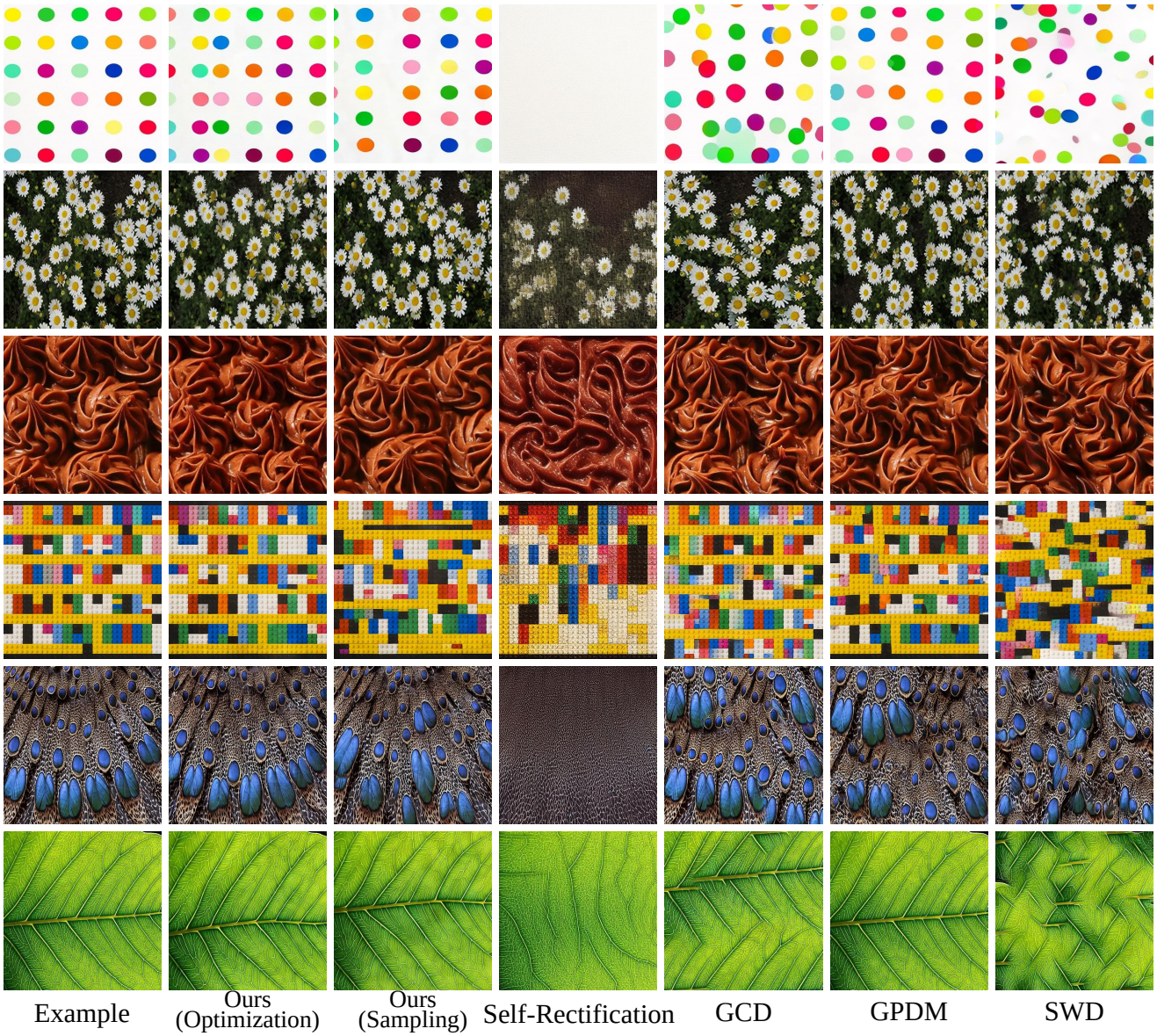


Figure 23. Comparison on unconditioned texture synthesis. Note that Self-Rectification needs a rough layout, but here, we only give it a random initialization as the target. In our results presented in the 4th and 6th rows, a fine-tuned VAE decoder is employed.

Figure 24. Additional results of our approach on stationary texture synthesis and expansion.

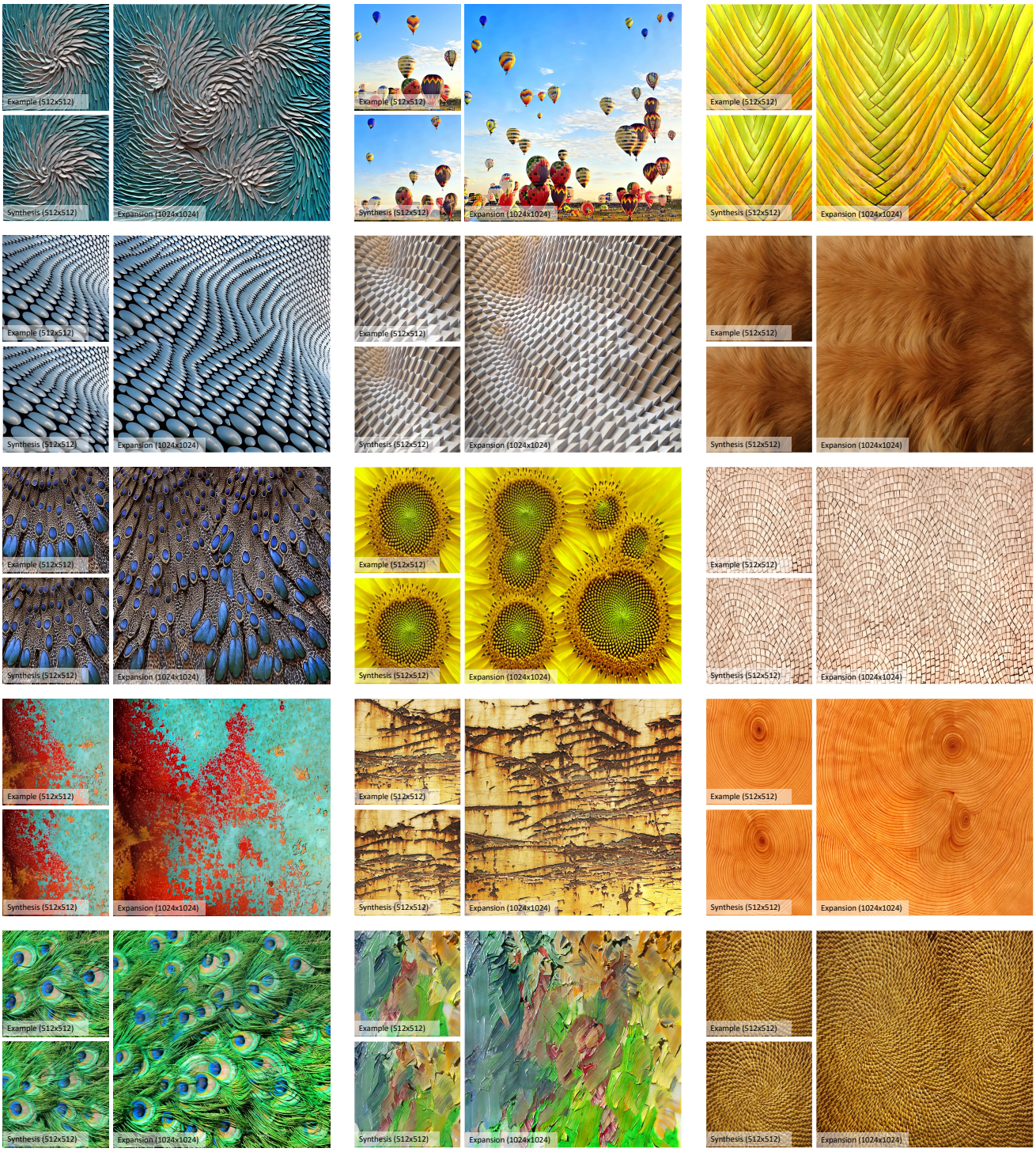


Figure 25. Additional results of our approach on non-stationary texture synthesis and expansion.

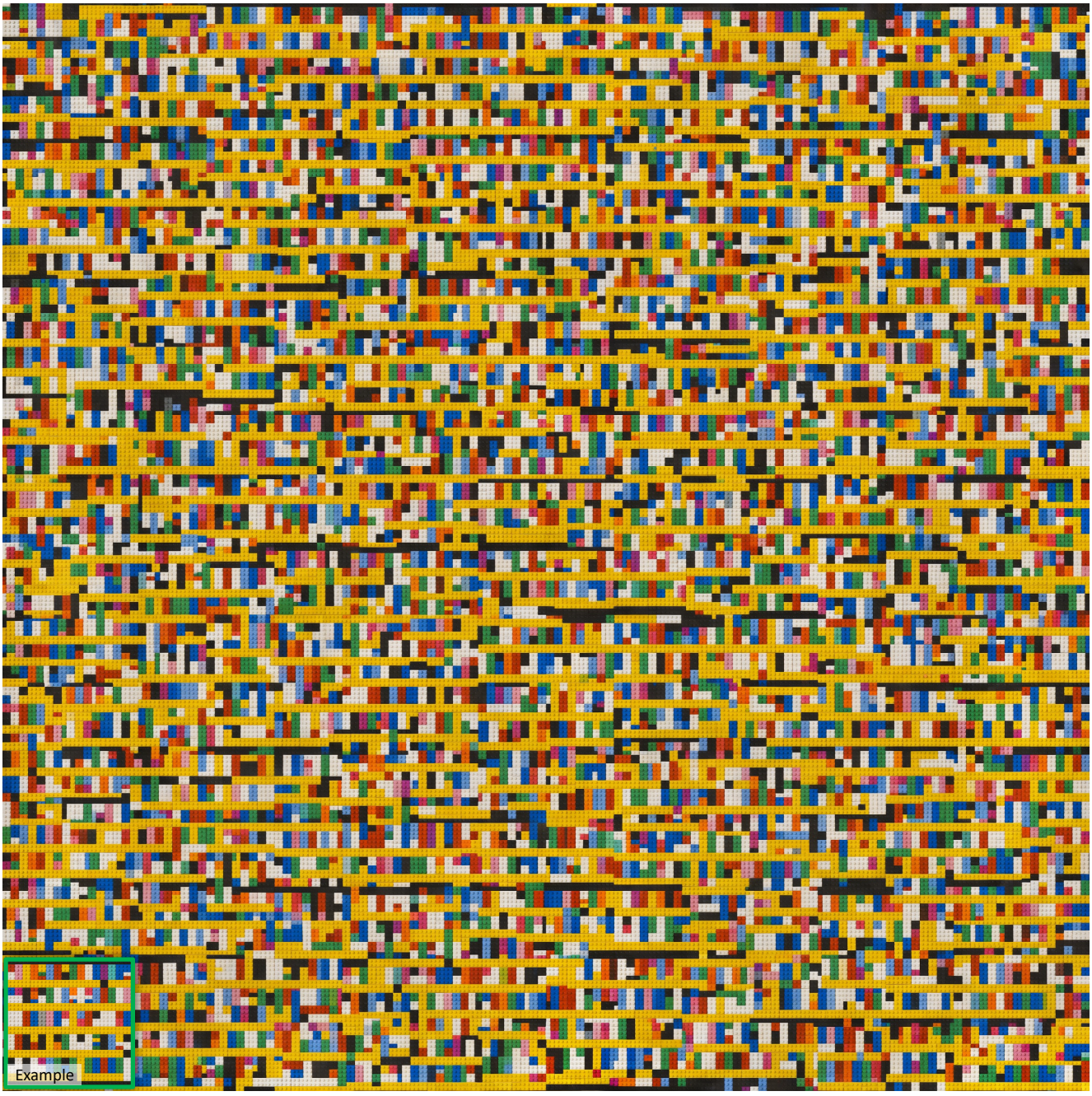


Figure 26. Texture synthesis with arbitrary resolution, where the above-generated image is in size of 4096×4096 pixels, synthesized from an example (bottom left) in size 512×512.