# RepLDM: Reprogramming Pretrained Latent Diffusion Models for High-Quality, High-Efficiency, High-Resolution Image Generation

**Boyuan Cao**[1] **Jiaxin Ye**[1] **Yujie Wei**[1] **Hongming Shan**[1]*

[1]Institute of Science and Technology for Brain-Inspired Intelligence &
MOE Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence &
MOE Frontiers Center for Brain Science, Fudan University
{caoby23, jxye22, yjwei22}@m.fudan.edu.cn, hmshan@fudan.edu.cn

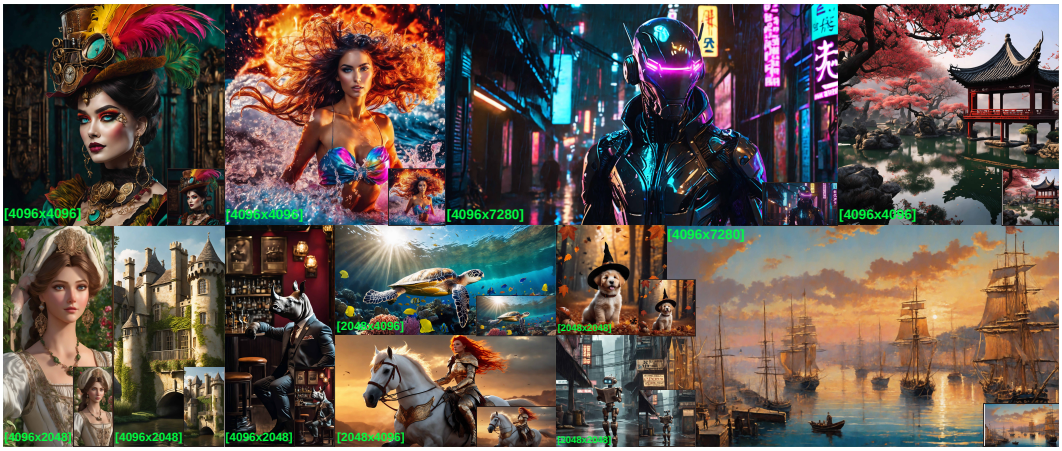


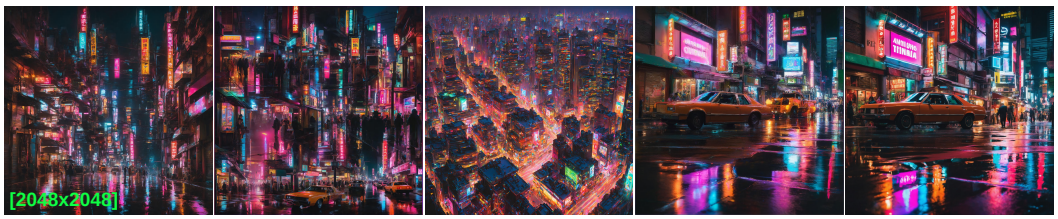

Figure 1: **High-resolution images generated by our RepLDM using a single consumer-grade 3090 GPU.** The corresponding thumbnails are generated by SDXL [32] at their training resolution.

## Abstract

While latent diffusion models (LDMs), such as Stable Diffusion, are designed for high-resolution (HR) image generation, they often struggle with significant structural distortions when generating images at resolutions higher than their training one. Instead of relying on extensive retraining, a more resource-efficient approach is to reprogram the pretrained model for HR image generation; however, existing methods often result in poor image quality and long inference time. We introduce RepLDM, a novel reprogramming framework for pretrained LDMs that enables *high-quality, high-efficiency, high-resolution* image generation; see Fig. 1. RepLDM consists of two stages: (**i**) an attention guidance stage, which generates a latent representation of a higher-quality training-resolution image using a novel training-free self-attention mechanism to enhance the structural consistency; and (**ii**) a progressive upsampling stage, which progressively performs upsampling in pixel space to mitigate the severe artifacts caused by latent space upsampling. The effective initialization from the first stage allows for denoising at higher resolutions with significantly fewer steps, improving the efficiency. Extensive experimental results demonstrate that RepLDM significantly outperforms state-of-the-art methods in both quality and efficiency for HR image generation, underscoring its advantages for real-world applications. Codes: https://github.com/kmittle/RepLDM.

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).




(a) SDXL (b) MultiDiffusion (c) Upsample Guidance (d) DemoFusion (e) RepLDM (**Ours**)

Figure 2: **Comparison of our RepLDM with prior work in generating 2048×2048 image**. The prompt is *Neon lights illuminate the bustling cityscape at night, casting colorful reflections on the wet streets*. Zoom-in for a better view.

## 1 Introduction

Diffusion models (DMs) have demonstrated impressive performance in visual generation tasks, particularly in text-to-image generation [6, 7, 13, 29, 30, 32, 41, 44–47, 50]. One notable variant of DMs is the latent diffusion model (LDM), which performs diffusion modeling in latent space to reduce training and inference costs, enabling HR generation up to $1024 \times 1024$. While it is plausible to modify the input size for higher-resolution generation, this often results in severe structural distortions, as illustrated in Fig. 2(a). Therefore, a recent research focus is on adapting trained LDMs for HR image generation without the need for additional training or fine-tuning (*i.e.* training-free manner), which can inherit the strong generation capacities of existing LDMs, especially open-sourced versions like Stable Diffusion.

Existing training-free approaches for HR image generation can be roughly categorized into three types: sliding window-based, parameter rectification-based, and progressive upsampling-based. Sliding window-based methods first divide the HR image into several overlapping patches and use sliding window strategies to perform denoising [1, 9, 22]. However, these methods could result in repeated structures and contents due to the lack of communication between windows; see Fig. 2(b). Parameter rectification-based methods attempt to correct models' parameters for better structural consistency through the entropy of attention maps, signal-to-noise ratio, and dilation rates of the convolution layers [11, 17–19, 51]. Though efficient, they often lead to the degradation of texture details; see Fig. 2(c). Unlike the two types mentioned above, progressive upscaling-based methods are to iteratively upsample the image resolution, which maintains better structural consistency and shows state-of-the-art (SOTA) performance [5, 24, 25, 33]. Unfortunately, these methods require fully repeating the denoising process multiple times, leading to an unaffordable computational burden; *e.g.*, AccDiffusion [25] takes 26 minutes to generate a $4096 \times 4096$ image. In addition, their upsampling operation in the latent space may introduce artifacts; see Fig. 2(d). To sum up, existing methods fail to ensure the fast, high-quality HR image generation.

In this paper, we propose RepLDM, a novel reprogramming framework for pretrained LDMs that is capable of generating high-quality, high-resolution images while keeping high-efficiency; see Fig. 2(e). Specifically, RepLDM decomposes the denoising process of LDMs into two stages: (**i**) an attention guidance stage, and (**ii**) a progressive upsampling stage. The first stage aims to generate a latent representation of a high-quality image at the training resolution through the proposed attention guidance, which is implemented via a novel training-free self-attention mechanism (TFSA) to improve structural consistency[2]. The second stage aims to progressively upsample the resolution in the pixel space rather than latent space, which can alleviate the severe artifacts caused by the latent space upsampling. By leveraging the effective initialization from the first stage, RepLDM can perform denoising in the second stage with significantly fewer steps, enhancing the overall efficiency with $5\times$ speedup. Extensive experimental results demonstrate the effectiveness and efficiency of RepLDM in generating HR images over the SOTA baselines.

The contributions of this work are summarized as follows. (**i**) We propose RepLDM, a novel framework for high-quality, high-efficiency, high-resolution image generation through reprogramming pretrained LDMs. (**ii**) We propose attention guidance, which can utilize a novel training-free self-

[2]In this paper, *structural consistency* refers to the plausibility of the overall scene layout and the realism of object structures within an image. Specifically, a reasonable layout should follow logical spatial relationships—for example, the sky should appear above the ground—while realistic object structures should conform to common sense, such as a cat having four legs rather than five.

attention to improve the structural consistency of the latent representation towards high-quality images at the training resolution. (**iii**) We propose progressively upsampling the resolution of latent representation in the pixel space, which can alleviate the artifacts caused by the latent space upsampling. (**iv**) Extensive experimental results demonstrate that the proposed RepLDM significantly outperforms the SOTA models in terms of image quality and inference time, emphasizing its great potential for real-world applications.

## 2 Related Work

**HR image generation with super-resolution.** An intuitive approach to generating HR images is to first use a pre-trained LDM to generate training-resolution[3] (TR) images and then apply a super-resolution model to perform upsampling [23, 27, 42, 43, 49]. Although one can obtain structurally consistent HR images in this way, super-resolution models are primarily focused on enlarging the image, and shown to be unable to produce the details that users expect in HR images [5, 24, 25].

**HR image generation with additional training.** Existing additional training methods either fine-tune existing LDMs with HR images [8, 16, 52] or train cascaded diffusion models to gradually synthesize higher-resolution images [14, 39]. Though effective, these methods require expensive training resources that are unaffordable for regular users.

**HR image generation in training-free manner.** Current training-free methods can be roughly classified into three categories: sliding window-based, parameter rectification-based, and progressive upsampling-based methods. Sliding window-based methods consider spatially splitting HR image generation [1, 9, 22]. Specifically, they partition an HR image into several patches with overlap, and then denoise each patch. However, due to the lack of communication between windows, these methods result in structural disarray and content duplication. While enlarging the overlaps of the windows mitigates this issue, it can result in unbearable computational costs. For the parameter rectification-based methods, some researchers discovered that the collapse of HR image generation is due to the mismatches between higher resolutions and the model's parameters [11, 17–19, 51]. These methods attempt to eliminate the mismatches by rectifying the parameters such as the dilation rates of some convolutional layers. While mitigating the structural inconsistency, they often lead to the degradation of image details. Different from the aforementioned two types, the progressive upsampling-based methods show SOTA performance in some recent studies [5, 24, 25, 33]. Though promising, they require fully repeating the denoising process multiple times, which incurs unbearable computational overhead. Additionally, these methods perform upsampling in the latent space, which may introduce artifacts.

Although their remarkable results, these methods fail to improve the quality of HR images and computational efficiency at the same time. In contrast, RepLDM aims to generate HR images with high quality and high efficiency, towards practical applications.

## 3 Method

### 3.1 Overview of RepLDM

Fig. 3 presents the overview of RepLDM, which reprograms a pre-trained LDM to generate HR images without further training. Formally, a pre-trained LDM utilizes a denoising U-Net model $\mathcal{F}$ to iteratively denoise the latent representation of size $h \times w \times c$, which is then converted back to the pixel space for final image generation through the decoder $\mathcal{D}$ of a variational autoencoder (VAE). We note that the initial latent representation is sampled from a Gaussian distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$, and for inference the encoder $\mathcal{E}$ of VAE is not involved.

Our RepLDM extends pre-trained LDMs for higher-resolution image generation in a training-free manner; *i.e.*, $\mathcal{E}$, $\mathcal{D}$ and $\mathcal{F}$ are fixed. RepLDM achieves this by decomposing the standard denoising process in the latent space into two stages: (**i**) attention guidance stage, and (**ii**) progressive upsampling stage. In the first stage, RepLDM aims to generate a latent representation of a higher-quality TR

[3] In this paper, *training resolution* refers to the resolution used during model training, while *high resolution* denotes a resolution that substantially exceeds the training resolution—beyond the level at which the model can directly produce satisfactory results.

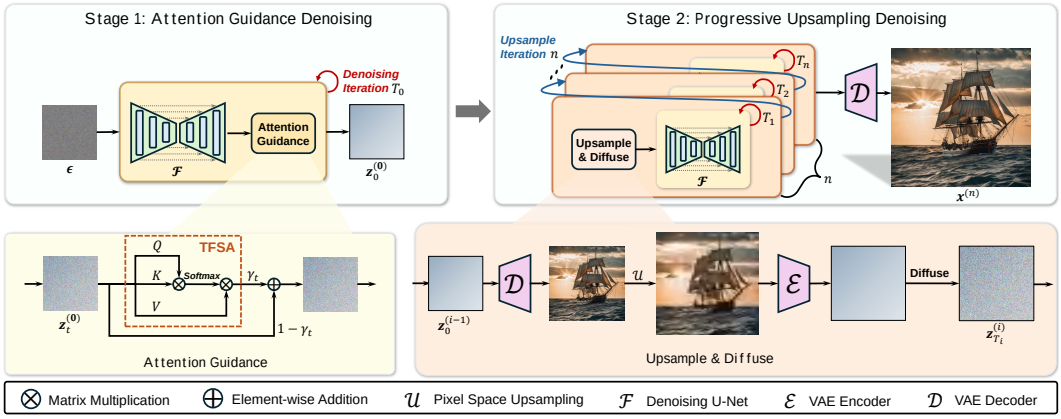



Figure 3: **Overview of RepLDM**. RepLDM divides the denoising process of a pre-trained LDM into two stages. The first stage leverages the introduced attention guidance to enhance the structural consistency by utilizing a novel training-free self-attention mechanism (TFSA). The second stage iteratively upsamples the latent representation in pixel space to eliminate artifacts.

image through the proposed attention guidance. The attention guidance is implemented as linearly combining the novel training-free self-attention mechanism (TFSA) and original latent representation to improve the structural consistency. In the second stage, RepLDM uses the latent representation provided by the first stage as a better initialization, and iteratively obtains higher-resolution images via the pixel space upsampling and diffusion-denoising refinement.

We detail the attention guidance stage in §3.2, followed by the progressive upsampling stage in §3.3.

### 3.2 Attention Guidance Stage

**Motivation.** Enhancing the structural consistency helps improve image quality [38]. However, it is challenging to do this in a training-free manner. We observe that the self-attention mechanism presents powerful global spatial modeling capability [4, 10, 26, 40], and this capability is parameter-agnostic. It is determined by the paradigm of global similarity calculation inherent to the self-attention mechanism [40, 53]. These insights motivate us to consider designing a novel training-free self-attention mechanism to elegantly enhance the global structural consistency of the latent representation.

**Denoising with attention guidance.** To improve the structural consistency of the latent representation at the training resolution $\boldsymbol{z} \in \mathbb{R}^{h \times w \times c}$, we propose a simple yet effective training-free self-attention mechanism for attention guidance, termed TFSA, formulated as:

$$\text{TFSA}(\boldsymbol{z}) = f^{-1}\left(\text{Softmax}\left(\frac{f(\boldsymbol{z})f(\boldsymbol{z})^{\text{T}}}{\lambda}\right) f(\boldsymbol{z})\right), \tag{1}$$

where the operation $f$ reshapes the latent representation into shape $(hw) \times c$ and $f^{-1}$ reshapes it back; $\lambda$ is the scaling factor, with a default value of $\lambda = \sqrt{c}$.

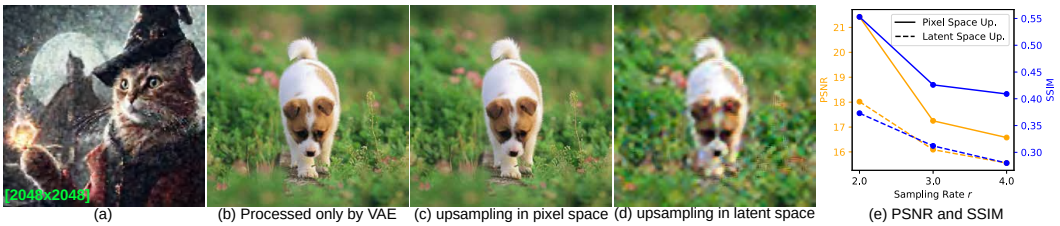
However, we empirically observe that directly using the TFSA in Eq. (1) to improve the structural consistency of the latent representation could lead to unstable denoising. Therefore, we propose linearly combining the outputs of TFSA and the original latent representation as attention guidance, which is formulated as:

$$\tilde{\boldsymbol{z}} = \gamma \text{TFSA}\left(\boldsymbol{z}\right) + (1-\gamma)\,\boldsymbol{z}, \tag{2}$$

where $\tilde{\boldsymbol{z}}$ is the structurally enhanced latent representation and $\gamma$ is the guidance scale. In Appendix A, we demonstrate that TFSA functions by modulating the distribution of latent representations. The term $(1-\gamma)\boldsymbol{z}$ in Eq. (2) serves as a statistical anchor, helping to keep the guided latent representations on the data manifold and ensuring smooth transitions in their distribution.

As shown in Fig. 3, we append the attention guidance in Eq. (2) to denoising U-Net model $\mathcal{F}$ and repeat the denoising process for a total of $T_0$ times for the first stage. We note that the denoising process starts from step $T_0$ to 1, and the final output of the first stage is denoted as $\boldsymbol{z}_0^{(0)}$.

**Adaptive guidance scale.** Considering that the latent representation is mostly non-semantic noise in the first few steps of denoising, we delay $k$ steps in introducing attention guidance. Moreover,




(a) (b) Processed only by VAE (c) upsampling in pixel space (d) upsampling in latent space (e) PSNR and SSIM

Figure 4: **Comparison between upsampling in pixel space and latent space.** (a) RepLDM with latent space upsampling leads to severe artifacts. (b)-(e): Qualitative and quantitative comparisons of different upsampling methods.

during the denoising process, the image structure is generated first, followed by local details [28, 39, 48]. Therefore, we primarily employ attention guidance in the early to mid-steps of denoising to focus on enhancing the structural consistency of the latent representation. Specifically, we introduce the adaptive guidance scale $\gamma_t$ by applying a decay to a given guidance scale $\gamma$, formulated as:

$$\gamma_t = \begin{cases} \gamma \left[ \frac{1}{2} \left[ \cos\left( \frac{T_0 - k - t}{T_0 - k} \pi \right) + 1 \right] \right]^{\beta} & \text{if } t \leq T_0 - k, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $\beta$ is the decay factor. In practice, considering that $k$ depends on $T_0$ for different resolutions, we use a delay rate $\eta_1 = \frac{k}{T_0}$ to control the number of steps for delaying attention guidance.

### 3.3 Progressive Upsampling Stage

**Motivation.** Fig. 2(a) shows that pre-trained LDMs still retain some ability to generate high-frequency information when directly used to synthesize HR images, although they exhibit structural disarray. Therefore, intuitively, we can utilize the latent representation produced by the first stage as a structural initialization, and generate the HR images through the "upsample-diffuse-denoise" iteration in the latent space. However, this pipeline leads to severe artifacts, as shown in Fig. 4(a). We speculate that this is due to *the upsampling of latent representations in the latent space*.

**Pilot study.** To examine this hypothesis, we conduct the following experiments. Specifically, we randomly select 10k images from ImageNet [3] to create an image set $\mathcal{P}$. For each image $\boldsymbol{x} \in \mathcal{P}$, we perform the following operations to obtain three additional image sets: **(i)** $\hat{\boldsymbol{x}} = \mathcal{D} \circ \mathcal{E}(\boldsymbol{x})$, which use VAE to obtain the reconstructed image set $\mathcal{P}_{\text{ref}}$; **(ii)** $\hat{\boldsymbol{x}} = \text{up} \circ \mathcal{D} \circ \mathcal{E} \circ \text{down}(\boldsymbol{x})$, which performs upsampling in pixel space to obtain the image set $\mathcal{P}_{\text{pix}}$; and **(iii)** $\hat{\boldsymbol{x}} = \mathcal{D} \circ \text{up} \circ \mathcal{E} \circ \text{down}(\boldsymbol{x})$, which performs upsampling in latent space to obtain the image set $\mathcal{P}_{\text{lat}}$. Both upsampling up and downsampling down are performed using bicubic interpolation. Fig. 4(e) reports the quantitative results, where $r$ represents the upsampling or downsampling rate. We calculate the PSNR and SSIM for pixel space upsampling set $\mathcal{P}_{\text{pix}}$ and latent space upsampling set $\mathcal{P}_{\text{lat}}$ with respective to the reference set $\mathcal{P}_{\text{ref}}$. It can be clearly observed that the latent space upsampling leads to a significant performance decline compared to pixel space upsampling. Fig. 4(b-d) shows upsampling in the pixel space produces images close to the reference while upsampling in latent space leads to severe artifacts and detail loss.

**Progressive denoising with pixel space upsampling.** Based on the above conclusion, we propose performing upsampling in the pixel space rather than latent space and utilize diffusion and denoising to refine the upsampled higher-resolution image. Specifically, the second stage consists of $n$ sub-stages to progressively upsample the training resolution to target resolution, each corresponding to one upsampling operation. For $i$-th sub-stage, $i = 1, \ldots, n$, we prepend an upsample and diffuse operation before the denoising process, which can be defined as:

$$\begin{aligned} \hat{\boldsymbol{z}}_0^{(i-1)} &= \mathcal{E} \circ \mathcal{U} \circ \mathcal{D}(\boldsymbol{z}_0^{(i-1)}), \\ \boldsymbol{z}_{T_i}^{(i)} &= \sqrt{\bar{\alpha}_{T_i}} \hat{\boldsymbol{z}}_0^{(i-1)} + \sqrt{1 - \bar{\alpha}_{T_i}} \boldsymbol{\epsilon}, \end{aligned} \tag{4}$$

where $\mathcal{U}$ represents upsampling operation, $\bar{\alpha}_{T_i}$ is the noise schedule hyper-parameter of the $T_i$-th diffusion time step, and $\boldsymbol{z}_0^{(i-1)}$ is the output of the $(i-1)$-th sub-stage; we use $\boldsymbol{z}_0^{(0)}$ to denote the output from the first stage. Then, $\mathcal{F}$ is used to iteratively denoise $\boldsymbol{z}_{T_i}^{(i)}$ from time step $T_i$ to obtain $\boldsymbol{z}_0^{(i)}$.

After completing all sub-stages, we obtain $z_0^{(n)}$, which is then decoded to produce the final output $\boldsymbol{x}^{(n)} = \mathcal{D}(\boldsymbol{z}_0^{(n)})$.

We empirically found that generating higher-resolution images requires more sub-stages. Additionally, when refining images using diffusion and denoising, higher resolutions demand larger time steps [39]. In practice, for flexibility, RepLDM allows users to customize the number of sub-stages $n$, and the diffusion time steps $T_i$ for each sub-stage by a pre-specified variable-length progressive scheduler $\eta_2 = \left[\frac{T_1}{T_0}, \frac{T_2}{T_0}, \ldots, \frac{T_n}{T_0}\right]$. The elements of $\eta_2$ represent the denoising steps of each sub-stage, normalized by $T_0$.

# 4 Experiments

## 4.1 Implementation Details

**Experimental settings.** We use SDXL [32] as the pre-trained LDM and conduct inference using a single NVIDIA 4090 GPU. To ensure consistency when testing inference speed, we use a single 3090 GPU, aligning with other methods. We randomly sample 33k images from the segment anything model (SAM) [21] dataset as the benchmark. Following the released code from DemoFusion [5], we use the EulerDiscreteScheduler [20] setting $T_0 = 50$ and the classifier-free guidance [15] scale to 7.5. Pixel space upsampling is performed using bicubic interpolation, and the decay factor $\beta$ is fixed at 3.

**Evaluation metrics.** The widely recognized metrics Frechet Inception distance (FID) [12], Inception score (IS) [35], and contrastive language-image pre-training (CLIP) score [34] are used to evaluate model performance. Additionally, since calculating FID and IS requires resizing images to $299 \times 299$, which may not be suitable for evaluating HR images, we follow the experimental settings of [5, 25] to perform ten $1024 \times 1024$ window crops on each image to calculate $\text{FID}_c$ and $\text{IS}_c$. Since FID is known to be sensitive to small implementation details [31], we employ a widely recognized implementation from a publicly available repository [37].

## 4.2 Quantitative Results

We compare RepLDM with the following models: (1) SDXL [32]; (2) MultiDiffusion [1]; (3) ScaleCrafter [11]; (4) DemoFusion [5]; (5) Upsample Guidance (UG) [18]; (6) AccDiffusion [25]; and (7) HiDiffusion [51]. For fair comparisons, we disabled the FreeU trick [38] in all experiments.

Table 1: **Quantitative comparison results**. The best results are marked in **bold**, and the second best results are marked by underline.

| Method | 2048 × 2048 | | | | | 2048 × 4096 | | | | | 4096 × 2048 | | | | | 4096 × 4096 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FID | IS | $\text{FID}_c$ | $\text{IS}_c$ | CLIP | FID | IS | $\text{FID}_c$ | $\text{IS}_c$ | CLIP | FID | IS | $\text{FID}_c$ | $\text{IS}_c$ | CLIP | FID | IS | $\text{FID}_c$ | $\text{IS}_c$ | CLIP |
| SDXL [32] | 99.9 | 14.2 | 80.0 | 16.9 | 25.0 | 149.9 | 9.5 | 106.3 | 12.0 | 24.4 | 173.1 | 9.1 | 108.5 | 11.5 | 23.9 | 191.4 | 8.3 | 114.1 | 12.4 | 22.9 |
| MultiDiff. [1] | 98.8 | 14.5 | 67.9 | 17.1 | 24.6 | 125.8 | 9.6 | 71.9 | 15.7 | 24.6 | 149.0 | 9.0 | 70.5 | 14.4 | 24.4 | 168.4 | 6.5 | 76.6 | 14.4 | 23.1 |
| ScaleCrafter [11] | 98.2 | 14.2 | 89.7 | 13.3 | 25.4 | 161.9 | 10.0 | 154.3 | 7.5 | 23.3 | 175.1 | 9.7 | 167.3 | 8.0 | 21.6 | 164.5 | 9.4 | 170.1 | 7.3 | 22.3 |
| UG [18] | 82.2 | 17.6 | 65.8 | 14.6 | **25.5** | 155.7 | 8.2 | 165.0 | 6.6 | 21.7 | 185.3 | 6.8 | 175.7 | 6.2 | 20.5 | 187.3 | 7.0 | 197.6 | 6.3 | 21.8 |
| HiDiff. [51] | 81.0 | 16.8 | 64.1 | 14.2 | 24.9 | 120.7 | 12.2 | 93.0 | 13.6 | 24.2 | 128.4 | 12.8 | 98.3 | 11.3 | 23.1 | 144.1 | 12.5 | 147.0 | 7.4 | 21.2 |
| DemoFusion [5] | 72.3 | **21.6** | 53.5 | **19.1** | 25.2 | 96.3 | 17.7 | 62.3 | 15.0 | **25.0** | 99.6 | 16.4 | 61.9 | 14.7 | 24.4 | 101.4 | 20.7 | 63.5 | 13.5 | **24.7** |
| AccDiff. [25] | 71.6 | 21.0 | 52.7 | 17.0 | 25.1 | **95.5** | 16.4 | 62.9 | 11.1 | 24.5 | 102.2 | 15.2 | 65.4 | 11.5 | 24.2 | 103.2 | 20.1 | 65.9 | 13.3 | 24.6 |
| RepLDM | **66.0** | 21.0 | **47.4** | 17.5 | 25.1 | **89.0** | **20.3** | **56.0** | **19.0** | **25.0** | **93.2** | **19.5** | **56.9** | **16.5** | **24.9** | **90.6** | **21.1** | **59.0** | **14.8** | 24.6 |

We report the performance of all methods on four different resolutions (Height × Width): $4096 \times 4096$, $4096 \times 2048$, $2048 \times 4096$, and $2048 \times 2048$. Considering that the generation time for HR images far exceeds that for low-resolution images, we used 2k prompts at the resolution of $2048 \times 2048$, and 1k prompts for resolutions greater $2048 \times 2048$. For all resolutions, we set $\gamma = 0.004$, $\beta = 3$ and $\eta_1 = 0.06$ for RepLDM. Given that the $4096 \times 4096$ resolution is significantly larger than other resolutions, we set $\eta_2 = [0.1, 0.2]$ (*i.e.*, $T_0 = 50$, $T_1 = 5$, $T_2 = 10$) for $4096 \times 4096$, and $\eta_2 = [0.2]$ (*i.e.*, $T_0 = 50$ and $T_1 = 10$) for other resolutions. When generating images with an aspect ratio of $r'$, we reshape the initially sampled Gaussian noise $\boldsymbol{\epsilon}$ in the first stage to match $r'$. This process keeps the number of tokens in $\boldsymbol{\epsilon}$ unchanged, preventing drastic fluctuations in the entropy of the attention maps in the transformer [19] leading to higher-quality images.

Table 1 manifests that RepLDM significantly outperforms previous SOTA models, AccDiffusion and DemoFusion. This indicates that RepLDM generates images with higher quality. For more comprehensive analyses, we repeat the experiments of Table 1 with different random seeds to perform error analyses and conduct a further comparison of the models on the LAION-5B benchmark [36]; see Appendix G.

Table 2 indicates that RepLDM demonstrates remarkable advantage in inference speed compared to the SOTA models. On a single 3090 GPU, RepLDM requires only about one-fifth of the inference time needed by SOTA models such as DemoFusion and AccDiffusion.

Table 2: **Model inference time**. The best results are marked in **bold**. Unit of Time: minute.

| Resolutions | SDXL [32] | MultiDiff. [1] | ScaleCrafter [11] | UG [18] | DemoFusion [5] | AccDiff. [25] | HiDiff. [51] | RepLDM |
|---|---|---|---|---|---|---|---|---|
| 2048 × 2048 | 1.0 | 3.0 | 1.0 | 1.8 | 3.0 | 3.0 | <u>0.8</u> | **0.6** |
| 2048 × 4096 | 3.0 | 6.0 | 6.0 | 4.0 | 11.0 | 12.7 | **1.9** | <u>2.0</u> |
| 4096 × 4096 | 8.0 | 15.0 | 19.0 | 11.1 | 25.0 | 26.0 | **3.4** | <u>5.7</u> |

### 4.3 Qualitative Results

In Fig. 5, RepLDM is qualitatively compared with AccDiffusion, DemoFusion, and MultiDiffusion. MultiDiffusion fails to maintain global semantic consistency. As indicated by the red boxes, DemoFusion and AccDiffusion tend to result in chaotic content repetition and severe artifacts, which we speculate are caused by upsampling in the latent space (as analyzed in §3.3). In contrast, RepLDM not only preserves excellent global structural consistency but also synthesizes images with more details. More qualitative comparison results can be found in Appendix B.

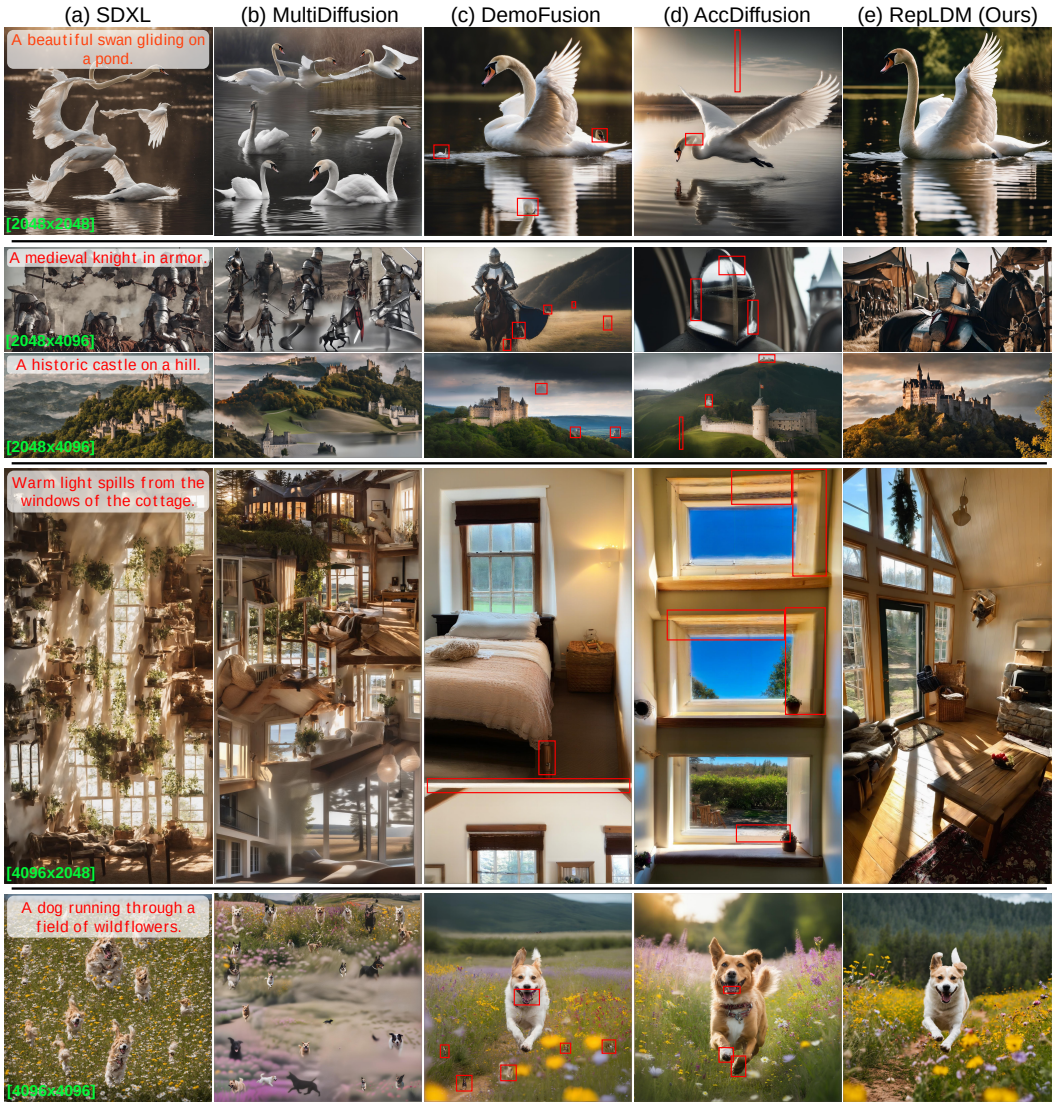



Figure 5: **Qualitative comparison with other baselines.** The prompts used to generate the images are presented in the white boxes. MultiDiffusion fails to maintain global semantic consistency. DemoFusion and AccDiffusion exhibit severe artifacts and content repetition. The red boxes indicate some synthesis errors. Zoom-in for a better view.

### 4.4 User Study

We invite 16 volunteers to participate in a double-blind experiment to further evaluate the performance of the models. Each volunteer is required to answer 35 questions. In each question, three images generated by AccDiffusion, DemoFusion, and RepLDM are presented. The volunteer needs to rate each image from 1 to 10 in terms of structural consistency, color abundance, and detail richness. We calculate the average of their scores. Moreover, to eliminate bias in each volunteer's ratings for each metric in each question, we subtract the minimum value among the three scores given by each volunteer for each metric in each question. The rectified score is denoted as score*. Table 3 shows that RepLDM surpasses previous SOTA models across all metrics.

Table 3: **Results of the user study.**

| Method | Structural Consistency | | Color Abundance | | Detail Richness | |
|---|---|---|---|---|---|---|
| | score ↑ | score* ↑ | score ↑ | score* ↑ | score ↑ | score* ↑ |
| AccDiff. [25] | 6.28 | 0.88 | 6.78 | 0.60 | 6.18 | 0.53 |
| DemoFusion [5] | 5.99 | 0.59 | 6.69 | 0.51 | 6.18 | 0.53 |
| RepLDM | **7.42** | **2.02** | **7.64** | **1.45** | **7.41** | **1.76** |

## 5 Ablation Study

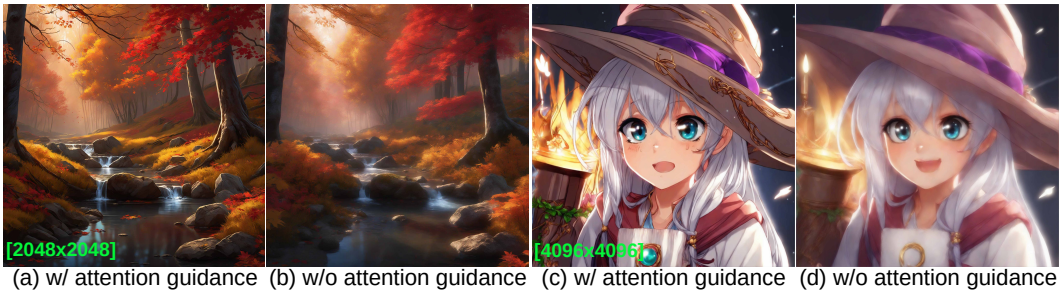
### 5.1 Attention Guidance

In this section, we first conduct ablation experiments on attention guidance, followed by ablation experiments on the hyper-parameters of attention guidance. In Appendix A, we provide a detailed analysis of how attention guidance improves latent structural consistency and image quality.

**Ablation on attention guidance.** We keep $\eta_2$ unchanged and analyze the effect of attention guidance through qualitative and quantitative experiments. Table 4 shows that attention guidance leads to improvements across various metrics, indicating that using attention guidance to enhance the consistency of latent encoding results in higher-quality images. The qualitative experiments in Fig. 6 demonstrate that using attention guidance eliminates image blurriness and enriches the image details. Note that FID and IS quantify the statistical differences between two distributions [2, 12, 35]. Since attention guidance mainly enhances visual quality by modifying the mid- and high-frequency components while preserving the low-frequency structure of the image, it has limited impact on the overall distributional statistics. Although attention guidance may not yield significant improvements in quantitative metrics, it provides a noticeable enhancement in human visual perception; see Table 3. Please refer to Appendix C.2 for additional qualitative ablation results.

Table 4: **Ablation on attention guidance (AG).** The best results are marked in **bold**.

| Method | 2048 × 2048 | | | | | 2048 × 4096 | | | | | 4096 × 2048 | | | | | 4096 × 4096 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FID | IS | $FID_c$ | $IS_c$ | CLIP | FID | IS | $FID_c$ | $IS_c$ | CLIP | FID | IS | $FID_c$ | $IS_c$ | CLIP | FID | IS | $FID_c$ | $IS_c$ | CLIP |
| w/o AG | 66.8 | **21.6** | 47.5 | 17.4 | **25.3** | 91.6 | **20.3** | 58.0 | 14.5 | **25.0** | 95.3 | **19.9** | 58.4 | 14.5 | **24.9** | 92.0 | **21.6** | 59.8 | 13.6 | 24.5 |
| w/ AG | **66.0** | 21.0 | **47.4** | **17.5** | 25.1 | **89.0** | **20.3** | **56.0** | **19.0** | **25.0** | **93.2** | 19.5 | **56.9** | **16.5** | **24.9** | **90.6** | 21.1 | **59.0** | **14.8** | **24.6** |




(a) w/ attention guidance (b) w/o attention guidance (c) w/ attention guidance (d) w/o attention guidance

Figure 6: **Ablation on attention guidance**. Zoom-in for a better view.

**Ablation on attention guidance with ControlNet.** To further demonstrate the generalization ability of attention guidance, in this section, we perform an qualitative ablation study of attention guidance with ControlNet [50]. Specifically, we conducted comparative experiments using two types of conditional guidance (canny and depth) across two resolution scales: $4096 \times 4096$ and $2048 \times 2048$. As shown in Fig. 7, the integration of attention guidance with ControlNet substantially enhances chromatic fidelity and structural granularity in synthesized images.

**Ablation on guidance scale $\gamma$.** We fix $\eta_1 = 0.06, \eta_2 = [0.2]$ and then explore the effect of the guidance scale $\gamma$ through both quantitative and qualitative experiments. For the quantitative

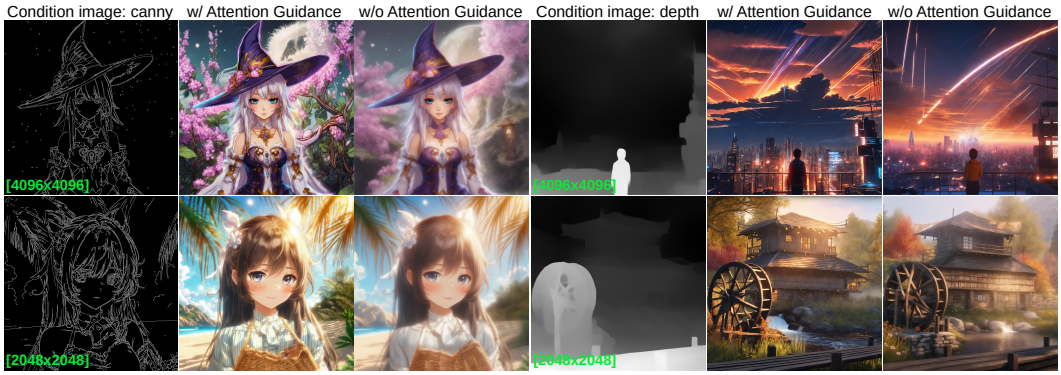



Figure 7: **Ablation on attention guidance with ControlNet.**

experiments, we find that $\gamma = 0.004$ performs better. Interestingly, when a larger $\gamma$ is used, the visual quality of the images can be further enhanced. As shown in Fig. 8, using a larger guidance scale results in richer image details. This allows users to generate images according to their preferences for detail richness and color contrast by adjusting the guidance scale. The setup and results of the quantitative experiments are detailed in Appendix C.2.


Figure 8: **Ablation on guidance scale**. Zoom-in for a better view.

**Ablation on delay rate $\eta_1$.** We fix $\gamma = 0.004, \eta_2 = [0.2]$ and then investigate the impact of the delay rate $\eta_1$ through both quantitative and qualitative experiments. The quantitative analysis results indicate that better generation results can be achieved when $\eta_1 = 0.06$, indicating that appropriately delaying the effect of attention guidance contributes to further improving the quality of the images. We conjecture that this is because, at the very beginning of the denoising process, the structural information in the latent encoding has not yet emerged, and thus attention guidance cannot effectively enhance structural consistency. As shown in Fig. 9, delaying the effect of attention guidance eliminates some generation errors, further improving image quality. The setup and results of the quantitative experiments are detailed in Appendix C.2.


Figure 9: **Ablation on delay rate.** Errors indicated by red boxes can be eliminated by delaying attention guidance. Zoom-in for a better view.

**Ablation on the time steps of attention guidance.** To explain why attention guidance needs to be applied during the early to middle steps of denoising, we apply attention guidance during different denoising steps of the first stage: (a) 47 to 33, (b) 32 to 17, and (c) 16 to 1. Fig. 10 shows that when attention guidance is applied during the early to middle steps of denoising, the image becomes clearer and more detailed; however, when attention guidance is applied during the later steps of denoising, it has negligible effect on the generated image. We speculate that this is because diffusion models tend to synthesize


Figure 10: **Applying attention guidance at different denoising steps.** Zoom-in for a better view.

structural information first [28, 39, 48], and once the structural information is generated, attention guidance may have a limited impact on structural consistency.

### 5.2 Progressive High-Resolution Denoising

In this section, we conduct ablation experiments on the progressive scheduler $\eta_2$ in the second stage of RepLDM. Specifically, we fixed $\gamma = 0, \eta_1 = 0$ and then explore the effect of the progressive scheduler $\eta_2$ through both quantitative and qualitative experiments. Quantitative experimental results indicate that an excessively large progressive scheduler value may result in a decline in image quality. This can also be observed in Fig. 11. It is evident that a too large progressive scheduler value may lead to structural misalignment and repetition issues observed in pre-trained SDXL. When the progressive scheduler value is sufficiently small, changing it yields similar visual effects. Therefore, we can choose a smaller progressive scheduler value (*e.g.*, 0.2) to accelerate inference. The setup and quantitative results are detailed in Appendix C.2.

(a) $\eta_2$ = [0.9] (b) $\eta_2$ = [0.7] (c) $\eta_2$ = [0.5] (d) $\eta_2$ = [0.1]

Figure 11: **Generated images using different $\eta_2$.** (a): When the value of progressive scheduler is too large, the structural repetition issue may reappear. (b) to (d): The visual effects are similar. Therefore, we can use a smaller progressive scheduler value to accelerate inference.

## 6 Limitations And Future Work

RepLDM exhibits limitations in the following aspects: (**i**) Effectively controlling text in images is challenging, as demonstrated by examples in Fig. 12. This may be due to the inherent limitations of SDXL in generating textual symbols. Text, due to its more regular structure compared to other image content, is difficult to restore by directly enhancing the structural consistency of the latent representation. We speculate that the most reliable approach would be to fine-tune the model specifically on images containing text. (**ii**) When generating ultra-high resolution images, such as $12800 \times 12800$, the second stage of RepLDM inevitably needs to be decomposed into more sub-stages, which increases the model's inference time.


(a) SDXL (b) RepLDM (c) SDXL (d) RepLDM

Figure 12: **Limitations of RepLDM.** The generation results of SDXL at its training resolution and those of RepLDM at higher resolutions are provided. As indicated by the white boxes, RepLDM fails to address the text structure errors inherited from SDXL.

Developing a low-cost and effective fine-tuning method to correct text generation errors may be a promising topic. Moreover, adapting attention guidance to other tasks, such as video generation can be an interesting issue.

## 7 Conclusion

In this paper, we reprogram pretrained LDMs, unlock their potentials, and propose RepLDM for high-quality, high-efficiency, high-resolution image generation. RepLDM divides the denoising process of an LDM into two stages: (**i**) attention guidance stage, and (**ii**) progressive upsampling stage. The first stage generates structurally enhanced latent representations through the proposed attention guidance, employing a novel parameter-free self-attention mechanism. The second stage iteratively performs upsampling in the pixel stage, thus eliminating the artifacts caused by latent space upsampling. Extensive experiments show that our proposed RepLDM significantly outperforms SOTA models while achieving $5\times$ speedup in HR image generation.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 62471148) and the computations in this research were supported by the CFFF platform of Fudan University.

## References

[1] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *ICML*, 2023.

[2] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[5] Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion: Democratising high-resolution image generation with no $$$. In *CVPR*, pages 6159–6168, 2024.

[6] 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 *ICML*, 2024.

[7] Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiaxiang Liu, Weichong Yin, Shikun Feng, et al. Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. In *CVPR*, pages 10135–10145, 2023.

[8] Lanqing Guo, Yingqing He, Haoxin Chen, Menghan Xia, Xiaodong Cun, Yufei Wang, Siyu Huang, Yong Zhang, Xintao Wang, Qifeng Chen, et al. Make a cheap scaling: A self-cascade diffusion model for higher-resolution adaptation. *arXiv preprint arXiv:2402.10491*, 2024.

[9] Moayed Haji-Ali, Guha Balakrishnan, and Vicente Ordonez. Elasticdiffusion: Training-free arbitrary size image generation. *arXiv preprint arXiv:2311.18822*, 2023.

[10] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *NIPS*, 34:15908–15919, 2021.

[11] Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In *ICLR*, 2023.

[12] 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. *NIPS*, 30, 2017.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NIPS*, 33:6840–6851, 2020.

[14] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

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

[16] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, pages 13213–13232. PMLR, 2023.

[17] Linjiang Huang, Rongyao Fang, Aiping Zhang, Guanglu Song, Si Liu, Yu Liu, and Hongsheng Li. Fouriscale: A frequency perspective on training-free high-resolution image synthesis. In *ECCV*, pages 196–212. Springer, 2024.

[18] Juno Hwang, Yong-Hyun Park, and Junghyo Jo. Upsample guidance: Scale up diffusion models without training. *arXiv preprint arXiv:2404.01709*, 2024.

[19] Zhiyu Jin, Xuli Shen, Bin Li, and Xiangyang Xue. Training-free diffusion model adaptation for variable-sized text-to-image synthesis. *NIPS*, 36, 2024.

[20] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NIPS*, 35:26565–26577, 2022.

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023.

[22] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. *NIPS*, 36:50648–50660, 2023.

[23] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, pages 1833–1844, 2021.

[24] Zhihang Lin, Mingbao Lin, Wengyi Zhan, and Rongrong Ji. Accdiffusion v2: Towards more accurate higher-resolution diffusion extrapolation. *arXiv preprint arXiv:2412.02099*, 2024.

[25] Zhihang Lin, Mingbao Lin, Meng Zhao, and Rongrong Ji. Accdiffusion: An accurate method for higher-resolution image generation. *arXiv preprint arXiv:2407.10738*, 2024.

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

[27] Xiaotong Luo, Zekun Ai, Qiuyuan Liang, Ding Liu, Yuan Xie, Yanyun Qu, and Yun Fu. Adaformer: Efficient transformer with adaptive token sparsification for image super-resolution. In *AAAI*, volume 38, pages 4009–4016, 2024.

[28] Yang Luo, Yiheng Zhang, Zhaofan Qiu, Ting Yao, Zhineng Chen, Yu-Gang Jiang, and Tao Mei. Freeenhance: Tuning-free image enhancement via content-consistent noising-and-denoising process. In *ACMMM*, pages 7075–7084, 2024.

[29] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI*, volume 38, pages 4296–4304, 2024.

[30] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171. PMLR, 2021.

[31] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, pages 11410–11420, 2022.

[32] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[33] Haonan Qiu, Shiwei Zhang, Yujie Wei, Ruihang Chu, Hangjie Yuan, Xiang Wang, Yingya Zhang, and Ziwei Liu. Freescale: Unleashing the resolution of diffusion models via tuning-free scale fusion. *arXiv preprint arXiv:2412.09626*, 2024.

[34] 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*, pages 8748–8763. PMLR, 2021.

[35] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NIPS*, 29, 2016.

[36] 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.

[37] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.3.0.

[38] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. In *CVPR*, pages 4733–4743, 2024.

[39] Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. *arXiv preprint arXiv:2309.03350*, 2023.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017.

[41] Chenhui Wang, Tao Chen, Zhihao Chen, Zhizhong Huang, Taoran Jiang, Qi Wang, and Hongming Shan. Fldm-vton: Faithful latent diffusion model for virtual try-on. *arXiv preprint arXiv:2404.14162*, 2024.

[42] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015*, 2023.

[43] Yang Wang and Tao Zhang. Osffnet: Omni-stage feature fusion network for lightweight image super-resolution. In *AAAI*, volume 38, pages 5660–5668, 2024.

[44] Yujie Wei, Shiwei Zhang, Zhiwu Qing, Hangjie Yuan, Zhiheng Liu, Yu Liu, Yingya Zhang, Jingren Zhou, and Hongming Shan. Dreamvideo: Composing your dream videos with customized subject and motion. In *CVPR*, pages 6537–6549, 2024.

[45] Yujie Wei, Shiwei Zhang, Hangjie Yuan, Biao Gong, Longxiang Tang, Xiang Wang, Haonan Qiu, Hengjia Li, Shuai Tan, Yingya Zhang, et al. Dreamrelation: Relation-centric video customization. *arXiv preprint arXiv:2503.07602*, 2025.

[46] Yujie Wei, Shiwei Zhang, Hangjie Yuan, Xiang Wang, Haonan Qiu, Rui Zhao, Yutong Feng, Feng Liu, Zhizhong Huang, Jiaxin Ye, et al. Dreamvideo-2: Zero-shot subject-driven video customization with precise motion control. *arXiv preprint arXiv:2410.13830*, 2024.

[47] Jiaxin Ye, Boyuan Cao, and Hongming Shan. Emotional face-to-speech. *arXiv preprint arXiv:2502.01046*, 2025.

[48] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *ICCV*, pages 23174–23184, 2023.

[49] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, pages 4791–4800, 2021.

[50] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023.

[51] Shen Zhang, Zhaowei Chen, Zhenyu Zhao, Yuhao Chen, Yao Tang, and Jiajun Liang. Hidiffusion: Unlocking higher-resolution creativity and efficiency in pretrained diffusion models. In *ECCV*, pages 145–161. Springer, 2025.

[52] Qingping Zheng, Yuanfan Guo, Jiankang Deng, Jianhua Han, Ying Li, Songcen Xu, and Hang Xu. Any-size-diffusion: Toward efficient text-driven synthesis for any-size hd images. In *AAAI*, volume 38, pages 7571–7578, 2024.

[53] Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. *arXiv preprint arXiv:2405.01434*, 2024.

# Appendix



## A How Does TFSA/Attention Guidance Work?

In this section, we further elaborate on the working mechanism of attention guidance. Our attention guidance enhances the structural consistency of the latent representation by integrating the output of TFSA. Therefore, we conduct a detailed analysis of TFSA. Specifically, the functionality of TFSA can be described in two aspects: (**i**) *clustering the related tokens* in the latent representations; (**ii**) *adjusting the amplitude of the high-frequency and low-frequency components* in the latent representations.

### A.1 TFSA Clusters Semantically Related Tokens

**Visualization of the clustering effect of TFSA.** TFSA reorganizes tokens based on their similarities. Intuitively, this enables TFSA to perform token clustering, which enhances the structural consistency of latent representations. To demonstrate the clustering effect of TFSA, we calculated the deviation of the tokens' mean (DTM) of the latent representations $\tilde{\boldsymbol{z}}_t$ and $\boldsymbol{z}_t$. Concretely, assuming $\boldsymbol{z}_t \in \mathbb{R}^{h\times w\times c}$, and $\boldsymbol{Z}_t = \text{Flatten}(\boldsymbol{z}_t) = [\boldsymbol{y}_{t1}, \ldots, \boldsymbol{y}_{tN}] \in \mathbb{R}^{N\times c}$, where $N = h \times w$, we calculate DTM as:

$$\text{DTM} = [\text{mean}(\boldsymbol{y}_{ti}) - \text{mean}(\boldsymbol{Z}_t) \text{ for } i = 1, \ldots, N] \tag{5}$$

To provide an intuitive illustration of the clustering effect of TFSA, we visualize the DTM based on token indices (*i.e.*, $i = 1, \ldots, N$) when $t$ is relatively large. As shown in columns (A) and (B) of Fig. 13, compared to the DTM of $\boldsymbol{z}_t$ (blue points), the DTM of $\tilde{\boldsymbol{z}}_t$ (red points) becomes more dispersed and exhibits distinct stripe patterns, indicating that TFSA indeed clusters the tokens of the latent representations. This clustering effect can be more directly demonstrated when $t$ is smaller. As shown in the heatmaps in columns (C) and (D) of Fig. 13, it is evident that TFSA clusters semantically related tokens.

**The clustering effect of TFSA leads to accelerated structural denoising.** Fig. 13 shows that the clustering effect of TFSA clarifies the semantic structures of objects, enabling the model to complete the denoising of low-frequency structures earlier. This early revelation of the overall image layout

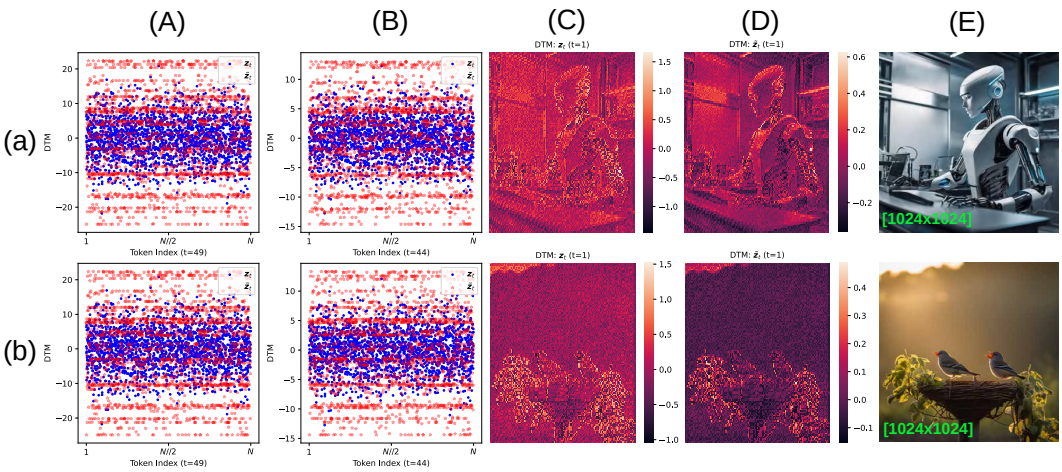



Figure 13: **The clustering effect of TFSA.** Columns (A), (B), (C), and (D) show the DTM of latent representations, while column (E) presents the corresponding generated RGB images.

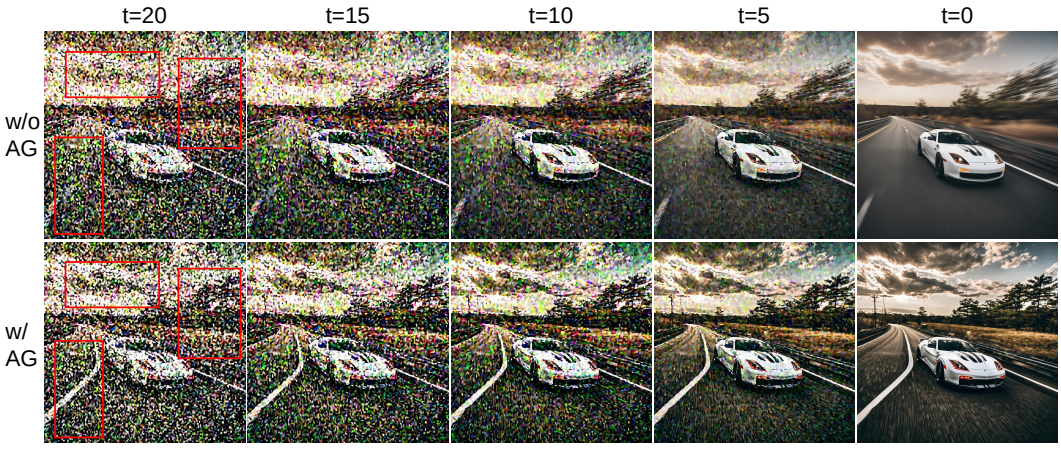



Figure 14: **Denoising visualization for the ablation of attention guidance.** As indicated by the red boxes, the clustering effect of TFSA prompts earlier structural emergence, delivering better prior for subsequent fine-detail generation. Resolution: $1024 \times 1024$.

provides a stronger prior for subsequent fine-detail generation. To illustrate this, Fig. 14 presents the denoising process for the ablation of attention guidance. Note the regions highlighted by red boxes. With the incorporation of attention guidance, these areas exhibit clearer structures, which facilitates the generation of more affluent details and more vivid colors in subsequent steps.

To quantitatively demonstrate that TFSA accelerates structural emergence, we calculate the SSIM between $\boldsymbol{z}_t$ and $\boldsymbol{z}_0$, where $t \in 1, 2 \ldots, T-1$, and $T = 50$. As shown in Fig. 15, compared to the naive denoising process, attention guidance consistently drives the latent representations closer to their final states at each step, indicating the structural foreseeability of TFSA.

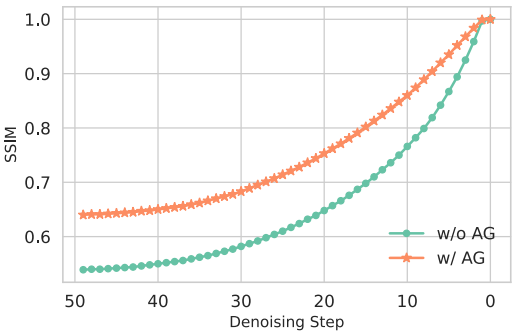



Figure 15: **Quantitatively analysis on the clustering effect of TFSA.** We calculate the SSIM between noised latents $\boldsymbol{z}_t$ $(1 \leq t \leq 49)$ and their corresponding clean latent $\boldsymbol{z}_0$.

### A.2 TFSA Adjusts the Amplitude of High- and Low-frequency Components

The aim of this experiment is to explain: (**i**) why appropriately delaying attention guidance can resolve structural deformation issues (as shown in Fig. 9); (**ii**) why attention guidance enhances the details and colors of the image (as shown in Fig. 6

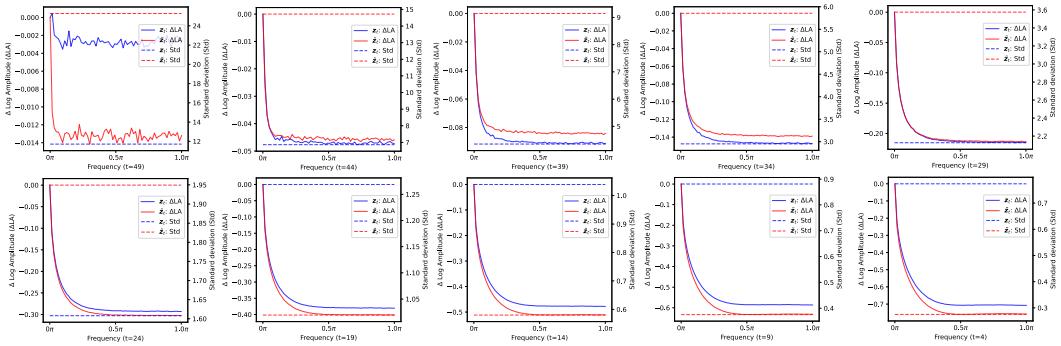


Figure 16: **The Fourier transform of the latent representation and the mean of the standard deviations across all channels.** $z_t$ is represented in blue, while $\tilde{z}_t$ is represented in red; the Fourier transforms are shown as solid lines, and the standard deviations are shown as dashed lines. The results are based on the generation process of 5k images.


Figure 17: **Visualization of attention maps in TFSA.** The query tokens are highlighted with red boxes, and the heatmap color ranges from blue to red, indicating increasing correlation strength between the key tokens and the query tokens. Resolution: $1024 \times 1024$. Zoom-in for a better view.

and 8); and (**iii**) why applying attention guidance in the later stages of denoising does not enhance the image details and colors (as shown in Fig. 10).

To explain the aforementioned three points, as shown in Fig. 16, we calculate the Fourier transforms of $z_t$ (blue solid line) and $\tilde{z}_t$ (red solid line), along with the mean of the standard deviations for all their channels (dashed line). It can be observed that TFSA significantly alters the relative amplitudes of the high- and low-frequency components in the latent representations during the initial denoising steps (from $t = 49$ to $t = 47$), particularly affecting the low-frequency components, which results in structural deformation. During the early and middle stages of denoising (from $t = 44$ to $t = 29$), TFSA increases the amplitudes of high-frequency components in the latent representations, which explains why attention guidance leads to richer details and colors. In the later stages of denoising (from $t = 28$ to $t = 0$), TFSA slightly suppresses the high-frequency components of the latent representations while almost leaving the low-frequency components unchanged. This explains why applying attention guidance in the later stages of denoising cannot enrich details and colors of the generated images.

Additionally, Fig. 16 shows that TFSA increases the standard deviation of $\tilde{z}_t$ during the early and middle stages of denoising, while decreasing it in the later stages. The trend of the standard deviation changes is closely consistent with the variation in the amplitude of the high-frequency components. We conjecture that this is because the amount of information in the latent representations is positively correlated with the standard deviation, where a larger standard deviation corresponds to more image details and larger high-frequency components.

### A.3 Visualization of Attention Maps in TFSA

To further demonstrate the clustering effect of TFSA on related tokens, we visualize its attention maps. As shown in Fig. 17, without using projection matrices, the correlations between tokens are determined jointly by their represented colors and semantics. For example, in Fig. 17(a), the key tokens correlated with the query token at the selected car location are related not only to the car itself (*i.e.*, the concept of the car) but also to its color. TFSA leverages such correlations to fuse token information, thereby accelerating the formation of the overall image layout.

## B Supplementary Qualitative Comparison of §4.3

Fig. 18 presents additional qualitative comparison results. MultiDiffusion continues to struggle with maintaining global consistency; as indicated by the red boxes, DemoFusion tends to produce repetitive content, a problem somewhat alleviated in AccDiffusion but not fully resolved. As highlighted by the black boxes, another issue with AccDiffusion is the presence of noticeable streak artifacts in the images.

## C Supplementary Ablation Experiments of §5

### C.1 Further Qualitative Analysis of Attention Guidance

Fig. 19 provides additional qualitative ablation results on attention guidance. Individual preferences for contrast, color vividness, and detail richness may vary. attention guidance allows users to adjust parameters such as the guidance scale to synthesize images according to their preferences.

### C.2 Ablation on the hyper-parameters of Attention Guidance

**Quantitative analysis of guidance scale.** We sampled 1k prompts, fixed $\eta_1 = 0.06, \eta_2 = [0.2]$ and performed ablation studies for guidance scale $\gamma$. The quantitative results are shown in Table 5. Considering all metrics, we find that $\gamma = 0.004$ achieved better quantitative results.

Table 5: **Quantitative ablation experiments on the guidance scale** $\gamma$. The best results are marked in **bold**, and the second best results are marked by underline.

| Method | 1024 × 1024 | | | | | 1600 × 1600 | | | | | 2048 × 2048 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FID ↓ | IS ↑ | FID$_c$ ↓ | IS$_c$ ↑ | CLIP ↑ | FID ↓ | IS ↑ | FID$_c$ ↓ | IS$_c$ ↑ | CLIP ↑ | FID ↓ | IS ↑ | FID$_c$ ↓ | IS$_c$ ↑ | CLIP ↑ |
| $\gamma = 0.000$ | 90.85 | 58.18 | 21.21 | **17.69** | 25.09 | 90.91 | 54.74 | **21.45** | 15.41 | 24.93 | 91.78 | 59.08 | 21.57 | 17.36 | 24.86 |
| $\gamma = 0.001$ | 90.50 | 58.04 | **21.34** | 16.76 | 25.08 | 91.17 | 54.31 | 21.19 | 15.47 | 24.93 | 91.40 | 58.75 | **21.87** | 15.85 | 24.86 |
| $\gamma = 0.002$ | 89.82 | 57.54 | 21.28 | 17.04 | 25.08 | 90.39 | **53.71** | 21.26 | 15.00 | 24.97 | 90.81 | 58.34 | 21.45 | 17.16 | 24.90 |
| $\gamma = 0.003$ | 90.10 | 57.08 | 20.80 | 16.61 | 25.08 | 90.56 | 53.95 | 21.35 | 15.46 | 24.98 | 90.87 | 58.40 | 21.47 | **17.60** | 24.92 |
| $\gamma = 0.004$ | **89.40** | **56.64** | 20.96 | 16.63 | 25.09 | **89.91** | 54.23 | 20.91 | **15.54** | 25.01 | **90.11** | **58.11** | 21.18 | 16.78 | 24.94s |
| $\gamma = 0.005$ | 90.17 | 57.50 | 20.89 | 16.34 | 25.12 | 90.24 | 55.19 | 20.67 | 15.21 | 25.02 | 90.46 | 58.91 | 20.79 | 16.87 | 24.97 |
| $\gamma = 0.006$ | 89.79 | 58.18 | 20.33 | 15.93 | 25.16 | 90.36 | 56.71 | 20.33 | 14.59 | 25.06 | 90.32 | 59.86 | 20.37 | 16.12 | 25.00 |
| $\gamma = 0.007$ | 90.42 | 60.29 | 20.07 | 16.20 | 25.21 | 90.91 | 59.35 | 20.36 | 14.16 | 25.12 | 90.86 | 61.81 | 20.14 | 15.70 | 25.06 |
| $\gamma = 0.008$ | 91.64 | 63.63 | 19.66 | 14.25 | **25.25** | 91.98 | 63.93 | 19.13 | 13.71 | 25.13 | 92.16 | 64.82 | 19.59 | 14.24 | 25.08 |
| $\gamma = 0.009$ | 94.29 | 67.87 | 19.15 | 13.00 | 25.25 | 94.38 | 70.21 | 19.45 | 12.12 | **25.16** | 94.39 | 68.84 | 19.22 | 13.63 | **25.12** |

**Quantitative analysis of delay rate.** We sampled 1k prompts, fixed $\gamma = 0.004, \eta_2 = [0.2]$ and performed ablation studies for delay rate $\eta_1$. Table 6 presents the experimental results, indicating that better results can be achieved when $\eta_1 = 0.06$. This means that appropriately delaying the effect of attention guidance can further enhance the quality of the generated images.

Table 6: **Quantitative ablation experiments on the delay rate** $\eta_1$. The best results are marked in **bold**, and the second best results are marked by underline.

| Method | 1024 × 1024 | | | | | 1600 × 1600 | | | | | 2048 × 2048 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FID ↓ | IS ↑ | FID$_c$ ↓ | IS$_c$ ↑ | CLIP ↑ | FID ↓ | IS ↑ | FID$_c$ ↓ | IS$_c$ ↑ | CLIP ↑ | FID ↓ | IS ↑ | FID$_c$ ↓ | IS$_c$ ↑ | CLIP ↑ |
| $\eta_1 = 0.00$ | 89.98 | 58.29 | 20.74 | 16.48 | 25.06 | 90.89 | 55.54 | 21.00 | 14.42 | 24.98 | 90.75 | 59.41 | 20.54 | 16.99 | 24.91 |
| $\eta_1 = 0.02$ | 89.96 | 57.67 | 20.99 | **16.87** | 25.05 | 90.76 | 54.77 | 21.08 | 15.35 | 24.95 | 91.78 | 59.08 | **21.57** | **18.16** | 24.86 |
| $\eta_1 = 0.04$ | 89.47 | 57.28 | 20.98 | 16.63 | 25.07 | 90.22 | 54.14 | 20.86 | 15.43 | 24.98 | 90.52 | 58.47 | 20.76 | 17.02 | 24.91 |
| $\eta_1 = 0.06$ | 89.44 | **56.64** | 20.92 | 16.58 | **25.11** | 89.91 | 54.23 | 20.91 | 15.54 | 25.01 | **90.11** | **58.11** | 21.18 | 16.78 | 24.94 |
| $\eta_1 = 0.08$ | 89.95 | 56.97 | 21.05 | 16.76 | 25.09 | **89.87** | 54.10 | 21.22 | 15.65 | 24.98 | 90.74 | 58.45 | 20.99 | 17.06 | **24.92** |
| $\eta_1 = 0.10$ | **89.29** | 56.88 | **21.11** | 16.84 | 25.09 | 89.97 | 53.99 | 21.04 | 15.37 | 24.99 | 90.41 | 58.45 | 20.99 | 17.12 | 24.92 |
| $\eta_1 = 0.12$ | 89.84 | 57.32 | 21.05 | 16.58 | 25.08 | 90.00 | 53.85 | 21.24 | **15.81** | 24.93 | 90.24 | 58.45 | 21.24 | 17.36 | 24.90 |
| $\eta_1 = 0.14$ | 89.85 | 57.12 | 20.91 | 16.40 | 25.09 | 90.06 | 53.83 | 21.33 | 15.62 | **24.99** | 90.69 | 58.25 | 21.17 | 16.74 | 24.91 |
| $\eta_1 = 0.16$ | 90.06 | 57.28 | 21.10 | 16.53 | 25.09 | 90.91 | 54.74 | **21.45** | 15.41 | 24.93 | 90.76 | 58.37 | 20.97 | 16.87 | 24.91 |
| $\eta_1 = 0.18$ | 90.16 | 57.29 | 20.88 | 15.10 | 25.08 | 90.26 | **53.79** | 21.06 | 15.07 | 24.97 | 90.78 | 58.33 | 21.05 | 17.21 | 24.90 |

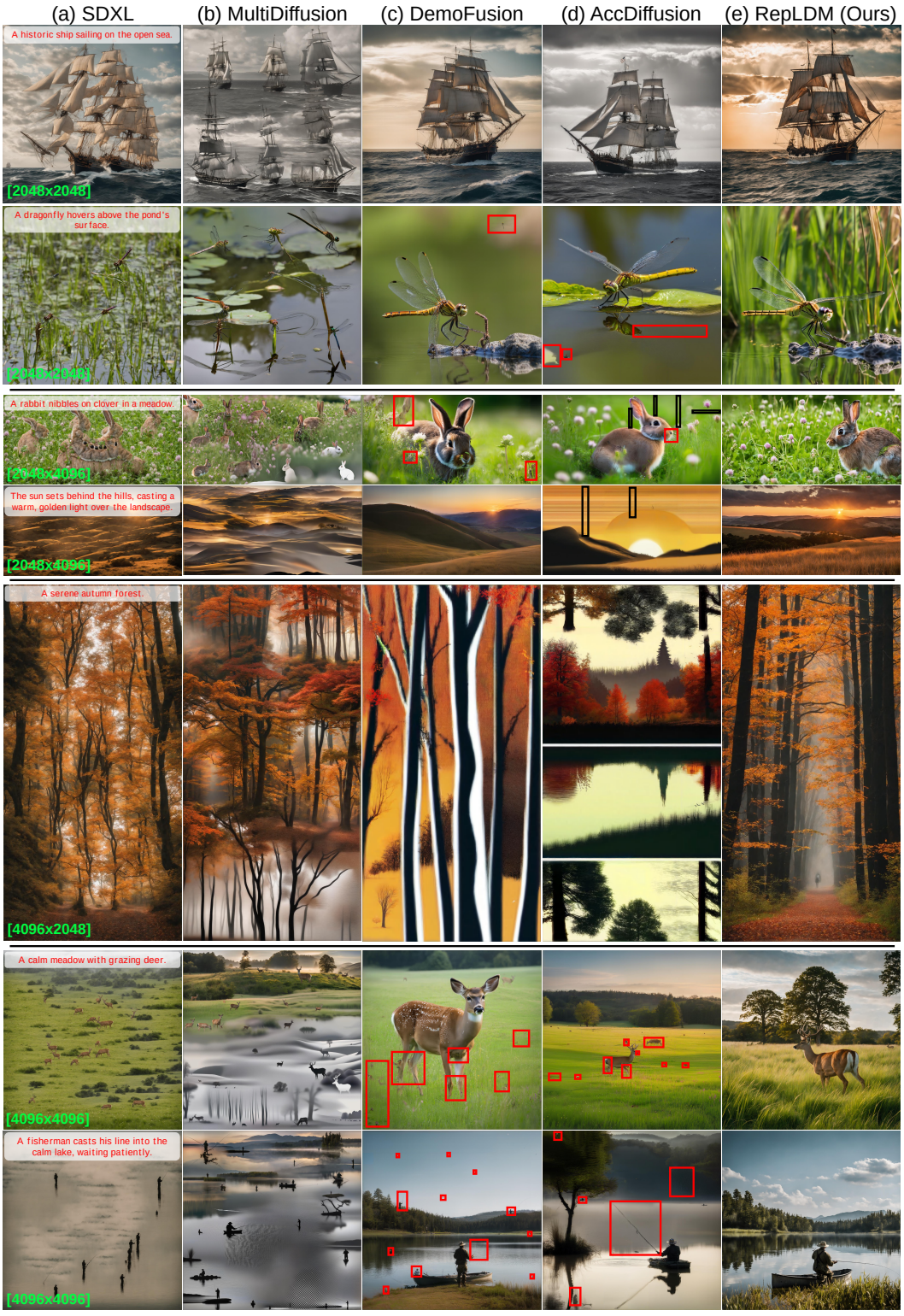



Figure 18: **Qualitative comparison with other baselines.** Zoom-in for a better view.

### C.3 Ablation on Progressive Scheduler Value

This section presents the results of quantitative ablation analysis on the progressive scheduler $\eta_2$ in the second stage of RepLDM. We fixed $\gamma = 0, \eta_1 = 0$, sampled 500 prompts, and generated 1k


Figure 19: **Further qualitative analysis of attention guidance (AG).** Using attention guidance significantly enhances image quality. The details were enriched, for example: the clouds in the sky, ripples on the water, reflections on the lake, and even the expressions in a person's eyes. Resolution: 2048×2048. Best viewed **ZOOMED-IN**.

images to investigate the optimal value of the progressive scheduler. Table 7 presents the quantitative results, indicating that using an excessively large progressive scheduler may lead to a decline in image quality.

Table 7: **Quantitative ablation study of the progressive scheduler Value**. The best results are marked in **bold**, and the second best results are marked by underline.

| Method | 1600 × 1600 | | | | | 2048 × 2048 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | FID ↓ | IS ↑ | FID$_c$ ↓ | IS$_c$ ↑ | CLIP ↑ | FID ↓ | IS ↑ | FID$_c$ ↓ | IS$_c$ ↑ | CLIP ↑ |
| SDXL | 101.56 | 25.78 | 73.67 | 21.23 | 26.87 | 112.64 | 18.44 | 79.03 | 20.61 | 26.55 |
| $\eta_2 = [0.9]$ | 94.59 | 27.04 | 67.60 | 23.01 | 26.97 | 97.14 | 24.48 | 64.34 | 22.14 | 26.59 |
| $\eta_2 = [0.8]$ | 93.13 | 28.80 | 65.67 | 24.83 | 26.99 | 93.93 | 26.75 | 60.84 | 23.27 | 26.77 |
| $\eta_2 = [0.7]$ | **92.05** | 29.44 | 65.35 | 24.97 | 27.07 | 92.50 | 28.17 | 57.34 | 24.05 | 26.93 |
| $\eta_2 = [0.6]$ | 92.94 | 30.79 | 64.57 | 24.29 | 27.11 | 91.86 | 30.45 | 55.38 | 24.96 | 26.98 |
| $\eta_2 = [0.5]$ | 92.73 | 30.65 | 63.43 | 24.26 | 27.13 | 91.80 | 31.18 | 54.32 | 24.48 | 27.02 |
| $\eta_2 = [0.4]$ | 93.04 | 30.96 | 63.33 | 24.77 | 27.14 | **91.71** | **32.47** | 53.72 | 25.16 | 27.03 |
| $\eta_2 = [0.3]$ | 92.93 | 30.91 | **63.09** | 24.84 | 27.15 | 92.39 | 30.72 | 53.32 | **26.63** | 27.07 |
| $\eta_2 = [0.2]$ | 93.09 | **31.17** | 63.23 | **25.71** | 27.17 | 92.71 | 30.45 | **53.19** | 26.19 | 27.12 |
| $\eta_2 = [0.1]$ | 93.44 | 30.69 | 63.75 | 25.18 | **27.22** | 92.94 | 30.69 | 53.77 | 24.71 | **27.18** |

# D Ablation on the Attention Guidance Components

## D.1 Ablation on the Guidance Scale Decay Strategy

To investigate the impact of different guidance scale decay strategies, we conduct ablation studies using two additional schemes—linear decay and exponential decay—and analyze their quantitative and qualitative performance. For quantitative ablation, we generate 2k samples at a resolution of $2048 \times 2048$ using each strategy and calculate the criterions on the SAM benchmark. Table 8 shows that different strategies yield similar results, indicating that RepLDM is not sensitive to a specific decay strategy. Fig. 20 illustrates the qualitative results. Qualitatively, these decay strategies also produce similar visual experience.

Table 8: **Ablation on the guidance scale decay strategies.** The best results are marked in **bold**, and the second best results are marked by underline.

| Strategies | FID ↓ | IS$_c$ ↑ | FID$_c$ ↓ | IS$_c$ ↑ | CLIP ↑ |
|---|---|---|---|---|---|
| Linear | 66.2 | 21.5 | 47.2 | **20.3** | **25.4** |
| Exponential | 66.8 | **21.8** | **47.0** | 16.3 | 25.3 |
| Cosine (default) | **66.0** | 21.0 | 47.4 | 17.5 | 25.1 |

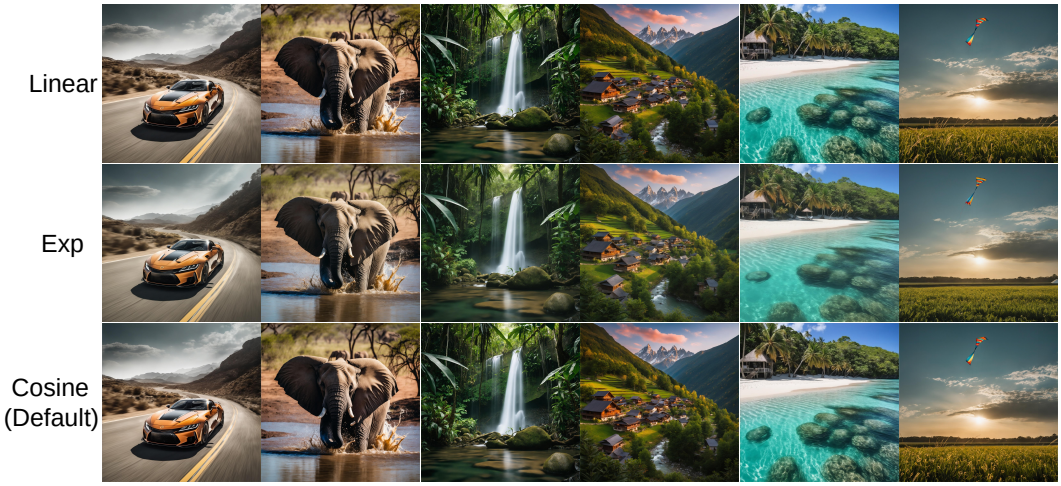



Figure 20: **Qualitative ablation on guidance scale decay ctrategies.**

### D.2 Ablation on the Attention Calculation Paradigm

For TFSA, our objective is to remove the learnable parameters from the Self-Attention mechanism, while maintaining its computational paradigm as unchanged as possible. In TFSA, $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are identical. Therefore, TFSA is a totally symmetric formula. As analyzed before, this paradigm encourages the clustering of semantically related tokens, and finally leads to finer details and richer colors. An interesting question arises: if we spatially downsample $\boldsymbol{Q}$, $\boldsymbol{K}$, or $\boldsymbol{V}$ before applying TFSA and reformulate it into an asymmetric paradigm (denoted as TFSA-A), would TFSA-A encourage the model to attend more explicitly from fine details to coarse structures?

To answer this question, we design an asymmetric variants, TFSA-A. Specifically, TFSA-A performs a 2×2 pooling operation to downsample the $\boldsymbol{K}$ and $\boldsymbol{V}$ matrices before the attention calculation operation, ensuring that the output of $\mathrm{Softmax}(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d})\boldsymbol{V}$ remains the of shape $(hw) \times c$. Table 9 shows that TFSA-A produces comparable quantitative results. In Fig. 21, we observe that although TFSA-A achieves quantitative results comparable to those of TFSA, its visual quality is significantly inferior. In fact, TFSA-A tends to reduce image details. This aligns with our hypothesis: the $2 \times 2$ pooling acts as a low-pass filter, causing the loss of fine-grained information in the latent representations and leading the model to focus more on low-frequency structures.

Table 9: **Ablation on the attention calculation paradigm.** The best results are marked in **bold**, and the second best results are marked by underline.

| **Paradigm** | FID ↓ | $\mathrm{IS}_c$ ↑ | $\mathrm{FID}_c$ ↓ | $\mathrm{IS}_c$ ↑ | CLIP ↑ |
|---|---|---|---|---|---|
| w/o guidance | 66.8 | 21.6 | 47.5 | 17.4 | **25.3** |
| w/ TFSA-A | 67.4 | **22.6** | 47.9 | **20.4** | **25.3** |
| w/ TFSA | **66.0** | 21.0 | **47.4** | 17.5 | 25.1 |

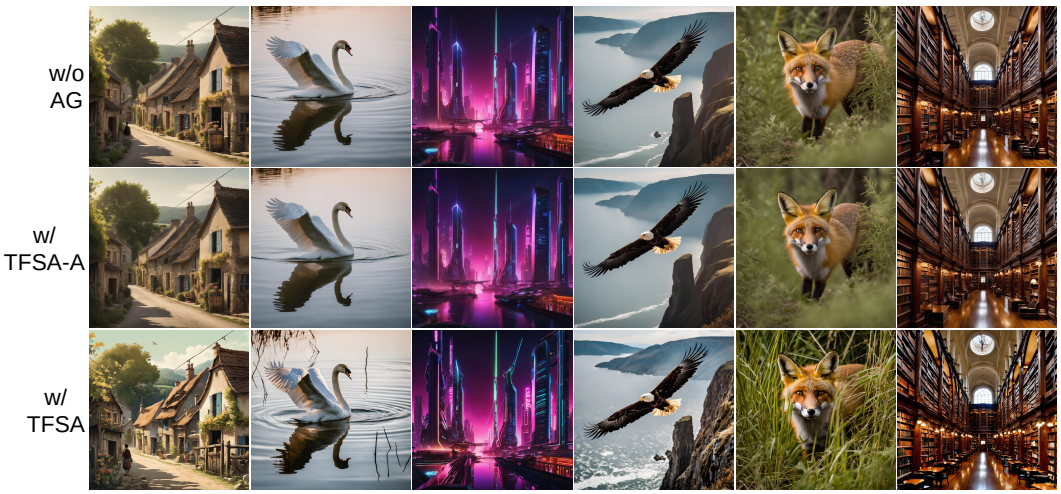



Figure 21: **Ablation on the attention calculation paradigm.** Resolution: $2048 \times 2048$.

## E Further Model Efficiency Analysis

**Computational complexity analysis of TFSA.** Note that attention guidance is only applied during the first stage of generation. Assume we have a HR image $\boldsymbol{x}_0$ with a resolution of $H \times W \times C$. we encode the image $\boldsymbol{x}_0$ into latent space and obtain latent representation $\boldsymbol{z}_0 \in \mathbb{R}^{h \times w \times c}$. Before feeding $\boldsymbol{z}_0$ into TFSA, we reshape it to a $(hw) \times c$ matrice. The computation of TFSA follows a formulation similar to that of self-attention: $\mathrm{Softmax}(\boldsymbol{z}_0 \boldsymbol{z}_0^T / \sqrt{c})\boldsymbol{z}$. Thus, the computational complexity of TFSA is $O((hw)^2 c)$. Taking SDXL as an example, the training resolution is $H = 1024$, $W = 1024$. After VAE encoding, $c = 4$, $h = H/8 = 128$, $w = W/8 = 128$. For each denoising step, the FLOPs of TFSA is approximately $2 \times (h \times w)^2 \times c$, which is around 2.15 GFLOPs—negligible compared to the FLOPs of the denoising network (several TFLOPs per step).

**How does pixel space upsampling accelerate generation?** To answer this question, we analyze the time consumption of each component in DemoFusion and RepLDM when generating images at the resolution of $4096 \times 4096$.

Table 10: **The time consumption of DemoFusion when generating $4096 \times 4096$ resolution images.**

| Metric | Denoise 1024 | Denoise 2048 | Denoise 3072 | Denoise 4096 | Decode 4096 | Total |
|---|---|---|---|---|---|---|
| number of steps | 50 | 50 | 50 | 50 | - | 200 |
| Time (s) | 12 | 185 | 480 | 901 | 106 | 1684 |

Table 11: **The time consumption of RepLDM when generating $4096 \times 4096$ resolution images.** The intermediate encoding/decoding operations are highlighted in underline.

| Metric | Denoise 1024 | Decode 1024 | Encode 3304 | Denoise 3304 | Decode 3304 | Encode 4096 | Denoise 4096 | Decode 4096 | Total |
|---|---|---|---|---|---|---|---|---|---|
| number of steps | 50 | - | - | 5 | - | - | 10 | - | 65 |
| Time (s) | 12 | 0 | 12 | 20 | 64 | 11 | 118 | 106 | 343 |

Table 10 shows that denoising at high resolutions is a time-consuming process. DemoFusion requires substantial generation time because it performs the full denoising process at high resolutions. Note that, compared with the cost of the denoising process at high resolutions, the costs of encoding and decoding are negligible. Table 11 shows that RepLDM significantly accelerates generation by substantially reducing the number of denoising steps at high resolutions. This is because RepLDM performs pixel space upsampling through multiple rounds of encoding and decoding, producing high-quality low-resolution images that serve as better initialization. As a result, RepLDM can significantly reduce the number of sampling steps required for HR generation, thereby accelerating the process. Moreover, Table 11 shows that the additional overhead from multiple intermediate encoding and decoding operations is also relatively minor compared to the total generation cost.

**Further efficiency comparison across different models.** To provide a more comprehensive assessment of model efficiency, we further report the NFE and FLOPs of different models when generating a single image at resolutions of $2048 \times 2048$ and $4096 \times 4096$. Tables 12 and 13 show that RepLDM significantly reduces the NFE and FLOPs required for inference by decreasing the number of denoising steps at high resolutions, thereby substantially reducing the time needed to generate HR images.

Table 12: **Inference cost of generating a $2048 \times 2048$ Image for different models.**

| Model | SDXL [32] | MultiDiff. [1] | ScaleCrafter [11] | HiDiff. [51] | UG [18] | DemoFusion [5] | AccDiff. [25] | RepLDM |
|---|---|---|---|---|---|---|---|---|
| NFE | 50 | 50 | 50 | 50 | 80 | 100 | 100 | 60 |
| TFLOPs | 3010 | 5420 | 2437 | 1857 | 3608 | 9015 | 8597 | 1140 |
| Time (min) | 1.0 | 3.0 | 1.0 | 0.8 | 1.8 | 3.0 | 3.0 | 0.6 |

Table 13: **Inference cost of generating a $4096 \times 4096$ Image for different models.**

| Model | SDXL [32] | MultiDiff. [1] | ScaleCrafter [11] | HiDiff. [51] | UG [18] | DemoFusion [5] | AccDiff. [25] | RepLDM |
|---|---|---|---|---|---|---|---|---|
| NFE | 50 | 50 | 50 | 50 | 80 | 200 | 200 | 65 |
| TFLOPs | 12026 | 29566 | 9759 | 5211 | 12624 | 72167 | 74225 | 7140 |
| Time (min) | 8.0 | 15.0 | 19.0 | 3.4 | 11.1 | 25.0 | 26.0 | 5.7 |

## F RepLDM Algorithm

The implementation details of RepLDM can be found in Algorithm 1, and further information is available in our code repository.

**Algorithm 1** RepLDM Inference Pipeline

**Require:** The number of inference time steps of the first stage $T_0$; progressive scheduler $\eta_2$; attention guidance scale $\gamma$; attention guidance delay rate $\eta_1$; the decay factor $\beta$; target image size tuple $(H', W')$; the denoising model $\boldsymbol{\mathcal{F}}$; denoising model's training resolution tuple $(H, W)$; VAE encoder $\boldsymbol{\mathcal{E}}$; VAE decoder $\boldsymbol{\mathcal{D}}$; noise scheduler's hyper-parameter list $\bar{\alpha}_{1:T_0}$.

**Initialization:**
$\boldsymbol{z}_{T_0}^{(0)} = \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ {Sampling from Standard Gaussian Distribution}
$n_{\text{stages}} = \text{length}(\eta_2) + 1$ {Get the total number of denoising stages}
$r' = \frac{H'}{W'}$ {Keep the aspect ratio and number of pixels unchanged}
$H^{(0)} = \text{ceil}(\sqrt{H \times W \times r'})$
$W^{(0)} = \text{ceil}(\sqrt{\frac{H \times W}{r'}})$
$H^{(n)} = H'$
$W^{(n)} = W'$
$area_{\text{list}} = \text{linspace}(H^{(0)} \times W^{(0)}, H^{(n)} \times W^{(n)}, n_{\text{stages}})$ {Upsampling according to the number of pixels}
$H_{\text{list}} = [\text{ceil}(\sqrt{i \times r'}) \quad \text{for} \quad i \quad \text{in} \quad area_{\text{list}}]$ {Get the height and width of each stage}
$W_{\text{list}} = [\text{ceil}(\sqrt{i / r'}) \quad \text{for} \quad i \quad \text{in} \quad area_{\text{list}}]$
$k_{\text{denoising}} = [T_0]$ {Get the number of denoising steps for each stage}
$k_{\text{denoising}}.\text{extend}([i \times T_0 \quad \text{for} \quad i \quad \text{in} \quad \eta_2])$
$k = T_0 \times \eta_1$ {Obtain the number of delay steps}
$\gamma_{\text{list}} = [\gamma(\frac{cos(\frac{T-k-i}{T-k}\pi)+1}{2})^{\beta} \quad \text{for} \quad i = 1, ..., T-k]$ {Obtain the guidance scale for each step}
**Denoising:**
**for** $s = 0, \ldots, n_{\text{stages}} - 1$ **do**
  $n_{\text{steps}} \leftarrow k_{\text{denoising}}[s]$
  **if** $s \geq 1$ **then**
    $\boldsymbol{x}^{(s)} \leftarrow \text{upsample}(\boldsymbol{x}^{(s-1)}, H_{\text{list}}[s], W_{\text{list}}[s])$ {Upsampling in pixel space}
    $\boldsymbol{z}_0^{(s)} \leftarrow \boldsymbol{\mathcal{E}}(\boldsymbol{x}^{(s)})$
    $\boldsymbol{z}_{n_{\text{steps}}}^{(s)} \sim \mathcal{N}(\sqrt{\bar{\alpha}[n_{\text{steps}}]}\boldsymbol{z}_0^{(s)}, (1 - \bar{\alpha}[n_{\text{steps}}])\boldsymbol{I})$
  **end if**
  **for** $t = n_{\text{steps}} - 1, \ldots, 0$ **do**
    $\boldsymbol{z}_t^{(s)} \leftarrow \boldsymbol{\mathcal{F}}(\boldsymbol{z}_{t+1}^{(s)}, t+1)$ {Denoising}
    **if** $s == 0$ and $t \leq T - 1 - k$ **then**
      $\boldsymbol{z}_t^{(s)} \leftarrow \gamma_{\text{list}}[t]\text{PFSA}(\boldsymbol{z}_t^{(s)}) + (1 - \gamma_{\text{list}}[t])\boldsymbol{z}_t^{(s)}$ {Attention Guidance}
    **end if**
  **end for**
  $\boldsymbol{x}^{(s)} \leftarrow \boldsymbol{\mathcal{D}}(\boldsymbol{z}_0^{(s)})$ {Obtain the pixel space image}
**end for**

## G Robustness Analysis

In this section, we conduct a robustness analysis to complement the experiments in §4.2, providing a more comprehensive evaluation of the models' performance. Our robustness analysis is conducted from two perspectives: (**i**) we vary the random seeds and repeat each experiment three times to compute the mean and standard deviation of all results; (**ii**) we randomly sample 20k HR images from the HR subset of LAION-5B dataset [36] to construct a new benchmark for evaluating the models' generalization performance. Since HR generation requires substantial computational resources, we analyze the four best-performing models from Table 1, *i.e.*, HiDiffusion, DemoFusion, AccDiffusion, and RepLDM.

**Analysis on the SAM benchmark.** We maintain the exact experimental settings as in §4.2 and conduct the analysis at resolutions of $2048 \times 2048$ and $4096 \times 4096$. Table 14 shows that RepLDM continues to exhibit superior performance across the repeated experiments.

Table 14: **Robustness analysis on the SAM benchmark**. The best results are marked in **bold**.

| Method | $2048 \times 2048$ | | | | | $4096 \times 4096$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | FID ↓ | IS ↑ | $\text{FID}_c$ ↓ | $\text{IS}_c$ ↑ | CLIP ↑ | FID ↓ | IS ↑ | $\text{FID}_c$ ↓ | $\text{IS}_c$ ↑ | CLIP ↑ |
| HiDiff. [51] | 80.29±0.57 | 17.18±0.40 | 63.55±0.63 | 15.26±0.76 | 24.95±0.04 | 144.24±0.84 | 12.71±0.14 | 146.62±0.32 | 7.48±0.28 | 21.18±0.05 |
| DemoF. [5] | 71.89±0.60 | 22.10±0.37 | 53.58±0.22 | 19.21±0.27 | 25.21±0.01 | 101.83±0.49 | 20.81±0.11 | 63.60±0.46 | 14.92±1.24 | **24.75**±0.03 |
| AccDiff. [25] | 71.37±0.48 | 21.21±0.32 | 53.04±0.33 | 19.24±1.72 | 25.13±0.01 | 102.41±1.40 | 19.88±0.24 | 65.86±0.17 | 12.73±0.71 | 24.65±0.02 |
| RepLDM | **66.08**±0.02 | **22.13**±0.74 | **47.31**±0.11 | **20.38**±2.03 | **25.30**±0.12 | **91.46**±0.61 | **21.63**±0.46 | **58.93**±0.20 | **15.02**±0.16 | 24.62±0.02 |

**Analysis on the LAION-5B benchmark.** Considering that only 1K samples were used for the $4096 \times 4096$ resolution in §4.2, which may lead to unstable metric evaluations, we double the number of samples to 2k for this resolution in the current experiment. Regarding evaluation metrics, since IS may lead to high variances beyond ImageNet, we follow some recent studies and adopt Kernel Inception distance (KID) for more accurate evaluation [17, 33]. Table 15 shows that on the LAION benchmark, RepLDM still demonstrates superior performance, surpassing previous SOTA models across all metrics.

Table 15: **Robustness analysis on the LAION-5B benchmark**. The best results are marked in **bold**. Since the magnitude of KID is relatively small, we multiply its mean and standard deviation by $10^3$.

| Method | $2048 \times 2048$ | | | | | $4096 \times 4096$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | FID ↓ | KID↓ | $\text{FID}_c$ ↓ | $\text{KID}_c$ ↓ | CLIP ↑ | FID ↓ | KID↓ | $\text{FID}_c$ ↓ | $\text{KID}_c$ ↓ | CLIP ↑ |
| HiDiff. [51] | 48.17±0.41 | 8.06±0.20 | 36.26±0.37 | 10.93±0.11 | 23.16±0.03 | 92.81±0.78 | 35.36±0.60 | 120.26±0.91 | 103.45±0.27 | 18.55±0.06 |
| DemoF. [5] | 34.15±0.31 | 4.50±0.05 | 21.38±0.17 | 6.80±0.06 | 25.44±0.02 | 37.03±0.27 | 5.71±0.14 | 30.77±0.36 | 16.12±0.22 | 25.12±0.04 |
| AccDiff. [25] | 34.49±0.31 | 4.92±0.08 | 22.71±0.17 | 8.57±0.11 | 24.90±0.02 | 38.56±0.23 | 7.21±0.20 | 38.85±0.29 | 20.87±0.20 | 24.46±0.01 |
| RepLDM | **34.08**±0.25 | **4.18**±0.04 | **20.30**±0.30 | **4.87**±0.13 | **25.78**±0.03 | **34.01**±0.26 | **4.13**±0.05 | **23.08**±0.26 | **12.08**±0.13 | **25.88**±0.04 |

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in §6, analyze their underlying causes, and discuss potential directions for future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a detailed description of the methodology and its underlying ideas in §3. In addition, we present the full algorithmic pipeline using pseudocode in Appendix F. We also commit to open-sourcing the code of our method.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: In Appendix F, we detail the method using pseudocode. Additionally, upon acceptance, we will clean and release our code base and share it on GitHub. All data used in this paper belongs to existing open source datasets and have been correctly cited to ensure reproduction.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The proposed method requires no training. We provide a detailed explanation of the inference and evaluation settings in §4.1. We determine the hyperparameters through ablation studies, with details provided in Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We repeated the experiments in Table 1 in Appendix G, computing the mean and standard deviation, and also conducted additional replication experiments on the LAION dataset [36].

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the resources needed to reproduce the experiments in §4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We conform to the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: the proposed method builds upon a pretrained generative model to produce higher-resolution images in a training-free manner, and thus does not introduce any additional or specific societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our method does not rely on any specific publicly released model and requires no specialized fine-tuning, and therefore does not necessitate additional safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all used resources such as implementations of baselines and data. We release our work with CC-By 4.0 license.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This study does not involve the release of any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: In the user study conducted in this work, we recruited volunteers to evaluate the quality of the generated images. Detailed instructions were provided to the participants. As the participants were volunteers, no compensation was involved.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: This study only required volunteers to evaluate the quality of generated images, and therefore poses no particular potential risks. Furthermore, to protect the privacy of participants' preferences, all responses were anonymized and randomized, and no personal information was collected.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.