
ScaleDiff: Higher-Resolution Image Synthesis via Efficient and Model-Agnostic Diffusion

Sungho Koh
Hanyang University
ksh000906@hanyang.ac.kr

SeungJu Cha
Hanyang University
sju9020@hanyang.ac.kr

Hyunwoo Oh
Hanyang University
komjii@hanyang.ac.kr

Kwanyoung Lee
Hanyang University
mobled37@hanyang.ac.kr

Dong-Jin Kim*
Hanyang University
djdkim@hanyang.ac.kr

Abstract

Text-to-image diffusion models often exhibit degraded performance when generating images beyond their training resolution. Recent training-free methods can mitigate this limitation, but they often require substantial computation or are incompatible with recent Diffusion Transformer models. In this paper, we propose **ScaleDiff**, a model-agnostic and highly efficient framework for extending the resolution of pretrained diffusion models without any additional training. A core component of our framework is Neighborhood Patch Attention (NPA), an efficient mechanism that reduces computational redundancy in the self-attention layer with *non-overlapping* patches. We integrate NPA into an SDEdit pipeline and introduce Latent Frequency Mixing (LFM) to better generate fine details. Furthermore, we apply Structure Guidance to enhance global structure during the denoising process. Experimental results demonstrate that ScaleDiff achieves state-of-the-art performance among training-free methods in terms of both image quality and inference speed on both U-Net and Diffusion Transformer architectures.

1 Introduction

Diffusion models have recently emerged as the leading approach in image generation [7], demonstrating the ability to synthesize high-fidelity images from simple text prompts [1, 3, 9, 21, 30]. While these models achieve impressive results at standard resolutions (e.g., under 1024^2), their performance significantly degrades when generating images at higher resolutions (e.g., beyond 2048^2), often producing artifacts such as repetitive patterns and structural distortions [12, 19]. However, training diffusion models directly on higher-resolution datasets is prohibitively expensive, requiring both large-scale, high-quality data and substantial computational resources.

As a result, recent research has focused on extending pre-trained diffusion models to generate higher-resolution images in a training-free manner [2, 8, 12, 16, 17, 19, 20, 22, 23, 49]. However, most of the existing methods are primarily designed for U-Net-based models [1, 30], and we observe that many existing methods are inapplicable [12, 16] or exhibit limited effectiveness [20, 28] when applied to recent Diffusion Transformer (DiT) models [9, 10, 21, 29]. Figure 1 highlights this issue, showing clear qualitative differences when existing methods are applied to DiT models. Although patch-based methods [2, 8, 22, 23] such as MultiDiffusion [2] are inherently architecture-agnostic and can generate detailed results with DiT models by processing the image in patches at its original trained resolution, they require significant computational redundancy to process overlapping patches. This

*Corresponding author.

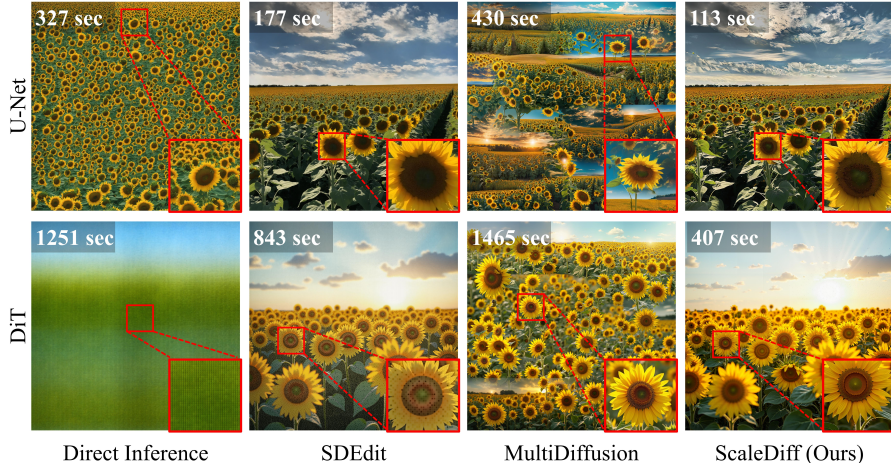


Figure 1: **Comparison between U-Net (SDXL) and DiT (FLUX).** Zoom in for a better view. Elapsed time to generate the image is shown in the top-left corner. Images are generated at 4096^2 .

inefficiency creates a major bottleneck for scalable higher-resolution image synthesis in real-world applications, highlighting the need for more efficient and architecture-agnostic solutions.

In this work, we propose **ScaleDiff**, a highly efficient and model-agnostic framework for extending the resolution capability of pre-trained diffusion models without any additional training. In particular, we introduce **Neighborhood Patch Attention (NPA)** to address the computational redundancy inherent in conventional patch-based methods. In self-attention layers, NPA divides the queries into *non-overlapping* patches and computes attention individually using key and value patches gathered from overlapping spatial neighborhoods. For non-self-attention layers (e.g., MLP), which are less sensitive to resolution, we process the full tensor directly. This design eliminates duplicate computations caused by overlapping image regions, ensuring seamless transitions across patch boundaries. We leverage an iterative upsample–diffuse–denoise pipeline [28] to generate higher-resolution images with global semantic coherence. While prior works [20, 44] perform upsampling in RGB-space, this often results in oversmoothed outputs and a loss of fine details [18, 20]. To address this, we introduce **Latent Frequency Mixing (LFM)**, which refines the RGB-space upsampled latent by replacing its low-frequency components with those from an alternative upsampling path in latent space. Finally, to further enforce global consistency, we incorporate Structure Guidance (SG) [17, 20, 39]. Unlike previous approaches that operate in RGB-space [20], our method applies SG in the latent space to avoid unnecessary computational overhead. SG reinforces structural coherence by aligning the low-frequency components of the model’s intermediate prediction with those from a reference latent.

Our main contributions are summarized as follows: (1) We present **ScaleDiff**, a model-agnostic framework demonstrating state-of-the-art results among training-free methods for higher-resolution image generation, achieving significant improvements in inference speed on both U-Net and DiT models. (2) We propose **NPA**, an efficient attention mechanism that reduces computational redundancy by using *non-overlapping* patches in self-attention layers. (3) We introduce **LFM**, a technique integrated with SG to enhance global structural coherence and fine detail synthesis during the denoising process.

2 Related Work

Text-to-Image Generation. Text-to-image generation has advanced rapidly with the development of diffusion models [15, 40, 41], which generate images by progressively denoising random noise. A major driver of this progress has been the integration of powerful text encoders—most notably CLIP [32]—which enable conditioning image generation on natural language prompts. Early methods such as DALL-E [3] and Imagen [36] demonstrated the potential of large-scale language-vision alignment. The introduction of the Latent Diffusion Model (LDM) [34], further improved efficiency by conducting the diffusion process in a lower-dimensional latent space, facilitating practical high-resolution synthesis. Recent works [5, 6, 9, 10, 21] continue to push the boundaries, exploring

alternative generative frameworks like Rectified Flow [26] and architectural innovations such as Diffusion Transformers (DiT) [29], which replace traditional U-Net [35] backbones with transformer-based architectures [45] to enhance scalability and performance.

Higher Resolution Image Generation. Scaling diffusion models to high resolutions often results in repetitive artifacts and structural distortions when naively extrapolated beyond their training resolution [12]. Several methods [6, 11, 25, 33, 43, 47] address this by fine-tuning or training on high-resolution datasets. Despite these efforts, their scalability remains limited due to the fundamental scarcity of high-resolution data and the sharply increasing training cost with image size.

To mitigate these challenges, recent work explores training-free strategies [2, 8, 12, 16, 17, 19, 20, 22, 23, 49] that extend the pretrained model’s resolution methods. [2, 8, 22, 23] subdivide the target high-resolution images into overlapping trained-resolution patches, which are processed individually and then stitched together. However, it significantly increases computation due to the necessary overlap and suffers from object repetition issues. Another line of research [12, 16, 19, 49] alters the internal behavior of the model during inference. For example, ScaleCrafter [12] introduces dilated convolutions into the U-Net to expand its receptive field and reduce repetition. However, these modifications are often architecture-specific and tend to degrade image quality at ultra-high resolutions. Editing-based pipelines [17, 20] generate an image at the model’s native resolution, upsample it, and then refine it using techniques such as SDEdit [28]. Nevertheless, these editing methods rely on the base model’s ability to generate strong local details at higher resolutions—a task that U-Net models can typically handle, but DiT models often struggle with. Compared to prior works, ScaleDiff generates high-resolution images with fine details regardless of the underlying model architecture, while significantly reducing computational overhead.

3 Methods

Given a diffusion model trained on fixed-resolution latents $z \in \mathbb{R}^{h \times w \times d}$, our goal is to generate higher-resolution image latents $Z \in \mathbb{R}^{s h \times s w \times d}$, where $s \geq 1$ denotes the scaling factor. To achieve this, we propose ScaleDiff, a training-free and model-agnostic framework that efficiently extends pre-trained diffusion models to higher resolutions. ScaleDiff consists of two main components. First, we introduce Neighborhood Patch Attention (NPA) (Section 3.2), an efficient attention mechanism applicable to both U-Net and DiT architectures that enables the processing of higher-resolution latents. Second, we present the ScaleDiff Upscaling Pipeline (Section 3.3), which builds upon the SDEdit framework [28]. The pipeline incorporates two key techniques: (i) Latent Frequency Mixing (LFM), which refines the reference latent to enhance details, and (ii) Structure Guidance (SG), which maintains global consistency by aligning the low-frequency components of intermediate latent predictions with those of the refined reference.

3.1 Backgrounds

Latent Diffusion Model. Latent diffusion models [29, 34] first compress an input image in RGB-space into a lower-dimensional latent $z_0 \in \mathbb{R}^{h \times w \times d}$ via an encoder \mathcal{E} . This enables subsequent diffusion and denoising to be performed more efficiently in latent space rather than directly on high-resolution pixels. During training, Gaussian noise is gradually added to the clean latent z_0 from $t = 0$ to T , following the forward process:

$$q(z_t|z_0) = \mathcal{N}(z_t; \sqrt{\bar{\alpha}_t}z_0, (1 - \bar{\alpha}_t)\mathbf{I}), \quad (1)$$

where $\{\bar{\alpha}_t\}_{t=0}^T$ is a set of prescribed noise schedules. A denoising network, often based on a U-Net [35] or Transformer [45] architecture, is trained to predict the noise added to z_t . During inference, sampling starts from random latent $z_T \sim \mathcal{N}(0, \mathbf{I})$. The trained network iteratively predicts the noise and denoises z_t to estimate z_{t-1} , progressively refining the latent until the final clean representation z_0 is obtained. This z_0 is then decoded to the pixel space by a decoder \mathcal{D} producing the final image.

Self-attention is key to capturing global context in diffusion models as it allows each token to weigh its interaction with all other tokens. The self-attention output O is formulated as

$$O = \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \right) V, \quad (2)$$

where Q, K, V are the Query, Key, and Value matrices derived by linearly transforming the features extracted from z_t through the network. Especially in transformer architecture, $Q, K, V \in \mathbb{R}^{h \times w \times d}$,

sharing the same spatial dimension as z_t . To encode positional information, transformer-based diffusion models [29] often incorporate positional encodings [42] into the self-attention mechanism. However, these encodings are tied to the training sequence length. Consequently, when self-attention is applied to sequences longer than those seen during training, unseen positional embeddings may disrupt spatial understanding and degrade image quality [27].

Patch-Wise Denoising. Conventional methods process $Z_t \in \mathbb{R}^{sh \times sw \times d}$ by directly computing self-attention with $Q, K, V \in \mathbb{R}^{sh \times sw \times d}$, which results in a computational cost of $s^4 h^2 w^2 d$ FLOPs. To reduce the significant computational cost and circumvent the resolution limitations introduced by positional encoding, MultiDiffusion [2] divides the input Z_t into N overlapping patches $\{z_t^i\}_{i=1}^N$, where $z_t^i \in \mathbb{R}^{h \times w \times d}$. Then, the denoising network is applied individually to each patch z_t^i to obtain the corresponding denoised patch z_{t-1}^i . These individually processed patches are then aggregated to reconstruct the full latent high-resolution Z_{t-1} by averaging the values in the overlapping regions.

Specifically, they apply a shifted crop sampling strategy with strides S_h and S_w , corresponding to the height and width dimensions, respectively. As a result, the total number of patches is given by $N = \left(\frac{sh-h}{S_h} + 1\right) \times \left(\frac{sw-w}{S_w} + 1\right)$. This leads to $Q, K, V \in \mathbb{R}^{N \times h \times w \times d}$, resulting in a total computational cost of $N h^2 w^2 d$ FLOPs in self-attention. In practice, the stride values are typically set to $(\frac{h}{2}, \frac{w}{2})$, yielding $N = (2s-1)^2$ patches and a corresponding FLOPs of $(2s-1)^2 h^2 w^2 d$ (Table 1).

3.2 Neighborhood Patch Attention

MultiDiffusion circumvents the inherent resolution limitations by decomposing the image into smaller, overlapping patches and processing them individually. They effectively reduce the computational cost in self-attention layers by limiting attention to local regions. However, it often requires substantial overlap between adjacent patches to ensure smooth transitions at the patch boundaries. This overlap causes non-self-attention layers to require nearly $4\times$ more FLOPs under a common stride setting, compared to a single forward pass that processes the full latent at once. A detailed breakdown of these computational costs is provided in Table 1.

To reduce the computational redundancy, we introduce Neighborhood Patch Attention (NPA) in Figure 2. Our key insight is that layers such as linear, convolution, and cross-attention perform operations on individual tokens or local regions. Unlike self-attention, these layers remain unaffected by increased input resolution, eliminating the need for patch-based processing. Building on this observation, NPA avoids patch-based processing for these non-self-attention layers, allowing them to operate on the full latent tensor Z_t in a single forward pass. This design eliminates redundant computations caused by overlapping patches, thereby keeping the computational cost of non-self-attention layers unchanged (Table 1).

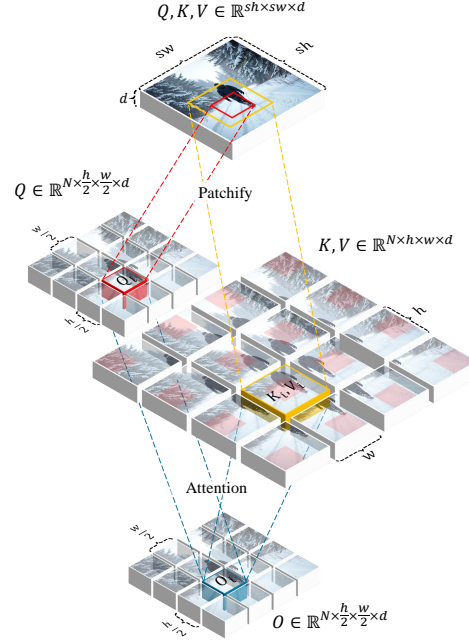


Figure 2: **Process of NPA.**

Within the self-attention mechanism, NPA is designed to further mitigate computational overhead by extracting queries from *non-overlapping* patches. Specifically, given a full query tensor $Q \in \mathbb{R}^{sh \times sw \times d}$, we apply a shifted crop sampling strategy to obtain a set of N query patches $\{Q_i\}_{i=1}^N$, where $Q_i \in \mathbb{R}^{\frac{h}{2} \times \frac{w}{2} \times d}$. The crop stride is set to match the patch size, i.e., $S_h = \frac{h}{2}$, $S_w = \frac{w}{2}$, resulting in a total of $N = \left(\frac{sh-h/2}{h/2} + 1\right) \times \left(\frac{sw-w/2}{w/2} + 1\right) = 4s^2$ patches. This ensures that the query patches *do not overlap*, keeping the total number of query tokens unchanged. For each non-overlapping query patch Q_i , we extract a corresponding key-value patch pair $(K_i, V_i) \in \mathbb{R}^{h \times w \times d}$ from its spatial neighborhood, using a larger window of size $h \times w$ centered on Q_i . Because each K_i and V_i patch is

Table 1: **Theoretical FLOPs comparison.** NPA reduces the computational complexity of self-attention without affecting the cost of non-self-attention operations. k denotes the convolution kernel size, and l is the length of text tokens. The Base method represents directly processing the input in a single forward pass. MultiDiffusion is calculated based on a common stride setting. Symbols highlighted in red indicate key elements for comparison.

Method	Linear	Conv	Cross-Attn	Self-Attn
Base [29]	$s^2 h w d^2$	$s^2 h w k^2 d^2$	$s^2 h w l d$	$s^4 h^2 w^2 d$
MultiDiffusion [2]	$(2s-1)^2 h w d^2$	$(2s-1)^2 h w k^2 d^2$	$(2s-1)^2 h w l d$	$(2s-1)^2 h^2 w^2 d$
NPA(Ours)	$s^2 h w d^2$	$s^2 h w k^2 d^2$	$s^2 h w l d$	$s^2 h^2 w^2 d$

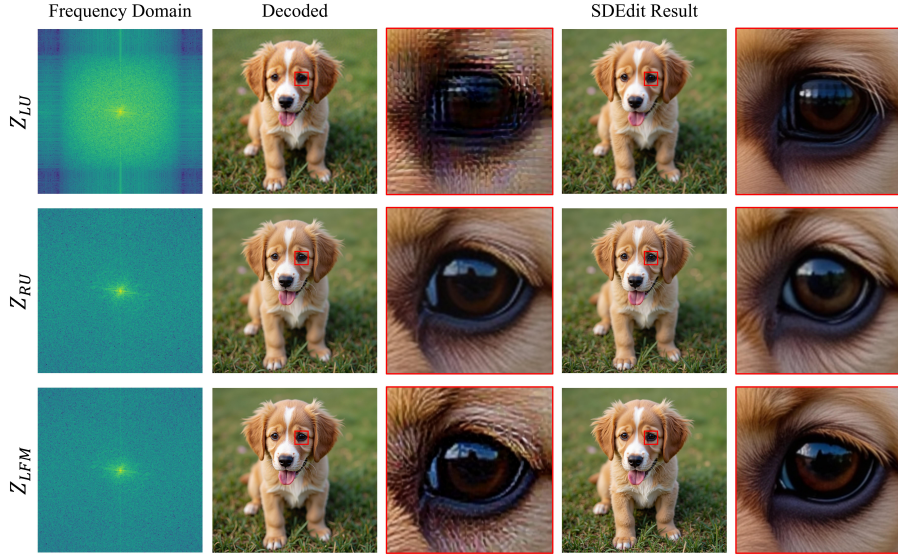


Figure 3: **Comparison between different reference latents.**

drawn from an expanded spatial window, the overlap between these patches allows every query patch to attend to a wider context and ensures smooth transitions across patch boundaries. We describe the overall process of the Query, Key, and Value patch extraction algorithm in the supplementary materials.

After that, self-attention is computed between non-overlapping query patch Q_i and its corresponding overlapping K/V neighborhood (K_i, V_i) , producing the output $O_i \in \mathbb{R}^{\frac{h}{2} \times \frac{w}{2} \times d}$ and resulting in $\frac{h^2 w^2}{4} d$ FLOPs. As this process is computed individually on $4s^2$ patches, the final cost is $s^2 h^2 w^2 d$ (Table 1). Finally, we reassemble the individually computed output patches $\{O_i\}_{i=1}^N$ into the full attention output tensor $O \in \mathbb{R}^{sh \times sw \times d}$ based on their original spatial positions, which is then passed to subsequent layers (e.g., an MLP block).

3.3 ScaleDiff Upscaling Pipeline

To maintain the global structure of a low-resolution image while enhancing high-frequency details during image generation, we employ an SDEdit [28]-based pipeline. Starting from a low-resolution image latent z , we first upscale it to obtain Z_{ref} , then inject noise up to the intermediate time step τ , and apply denoising using NPA. However, naively applying this process often results in outputs that lack fine texture details and appear overly smoothed [18]. This phenomenon arises because upscaling a low-resolution image closely resembles the resizing operation used during training. As a result, the model tends to denoise the input toward the distribution of resized training images, rather than synthesizing fine-grained details [20].

To understand this limitation, we compare two upsampling strategies in Figure 3. Upsampling directly in latent space yields Z_{LU} , which lacks high-frequency components, resulting in visible artifacts

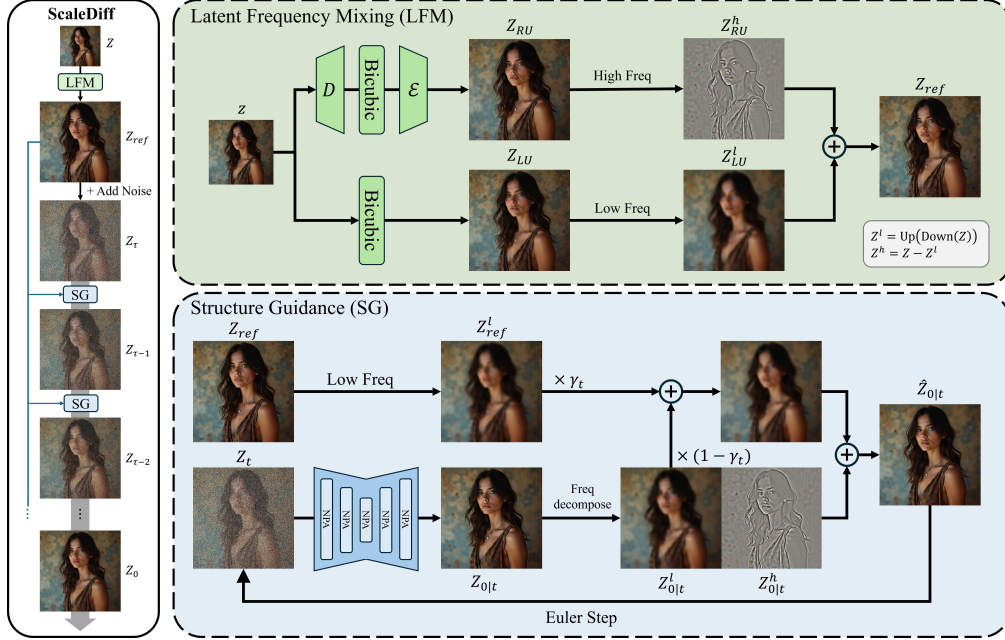


Figure 4: **Overview of our pipeline.** ScaleDiff starts from a generated low-resolution latent, upsamples it with LFM, and diffuses it to an intermediate timestep τ . At each denoising step, the network—integrated with NPA—applies structure guidance to preserve the global image structure.

in the decoded image that propagate to the final output. However, because Z_{LU} deviates from the training distribution of RGB-resized images, the subsequent denoising process is not biased toward oversmoothing. In contrast, upsampling in RGB space followed by VAE encoding yields Z_{RU} , which contains rich frequency information and ensures stable, artifact-free decoding. However, since this process closely mimics training-time resizing operations, it strongly biases the model toward reproducing oversmoothed textures instead of generating fine details.

To leverage the complementary strengths of both approaches, we propose **Latent Frequency Mixing (LFM)**. By combining the low-frequency content from Z_{LU} —which steers denoising away from the oversmoothing regime—with the high-frequency content from Z_{RU} —which ensures stable decoding—we can achieve both sharpness and natural texture. The refined reference latent is:

$$Z_{ref} = Z_{RU}^h + Z_{LU}^l, \quad (3)$$

where l and h denote low- and high-frequency components obtained by downsampling and upsampling operations and their residual. This construction guides subsequent denoising toward generating detailed outputs without oversmoothing.

Since NPA processes images through patches, it can introduce repetitive patterns. To mitigate this and enforce global structural consistency, we apply **Structure Guidance (SG)** following prior work [17, 20, 39]. At each timestep t , we obtain a clean estimate $Z_{0|t}$ from the noisy latent Z_t and guide it toward Z_{ref} by blending their low-frequency components:

$$\hat{Z}_{0|t} = Z_{0|t}^h + (1 - \gamma_t)Z_{0|t}^l + \gamma_t Z_{ref}^l, \quad (4)$$

where γ_t controls the guidance strength. This guided prediction $\hat{Z}_{0|t}$ is then utilized to compute the subsequent noisy latent Z_{t-1} . This process steers the generation towards the global structure defined by Z_{ref} while allowing the model to synthesize high-frequency details. Figure 4 illustrates our pipeline.

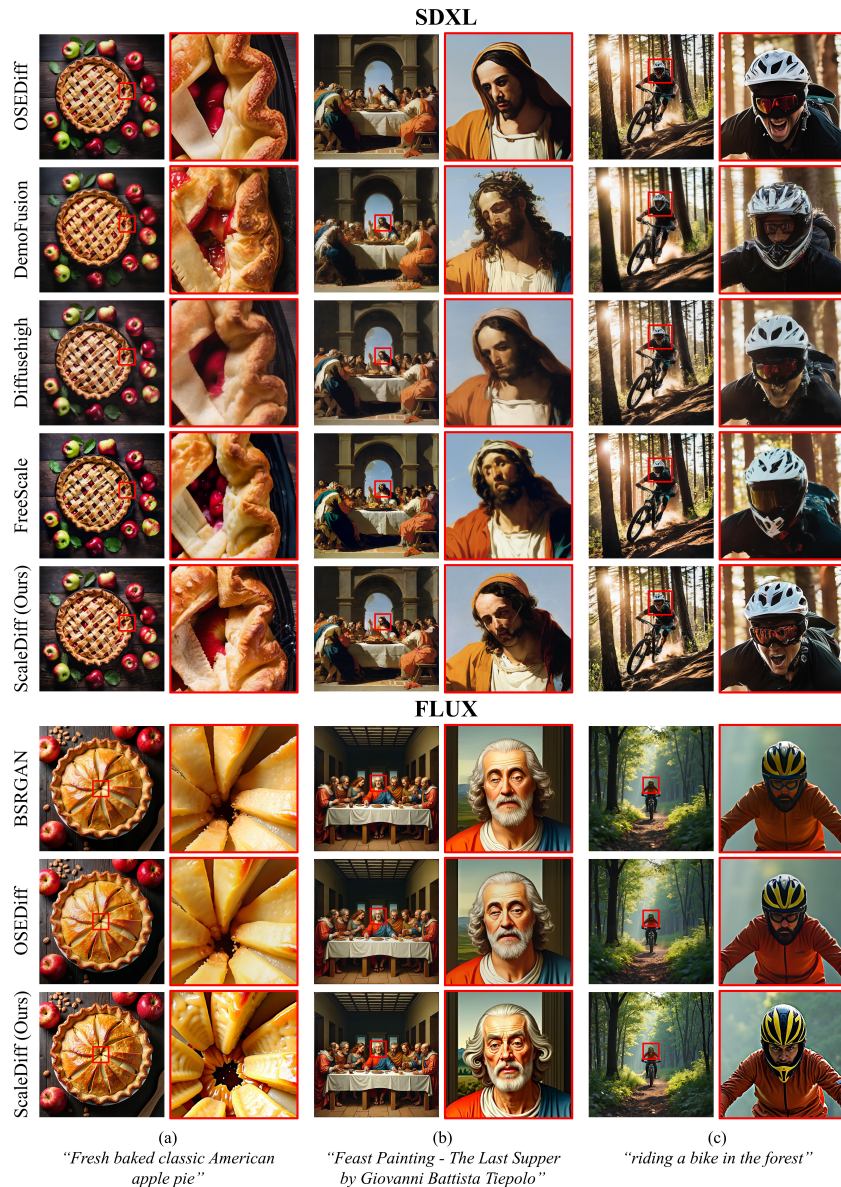


Figure 5: **Qualitative comparison with other methods.** All images are generated at 4096^2 from the same low-resolution input. Zoom in for a better view.

4 Experiments

4.1 Experimental Settings

Implementation Details. We evaluate our proposed method, **ScaleDiff**, on both FLUX [21] and SDXL [30] within an iterative $1024^2 \rightarrow 2048^2 \rightarrow 4096^2$ generation pipeline. For **FLUX**, we use a noise timestep $\tau = 600$, and a structure guidance strength of $\gamma_t = t$. This setup uses 30 denoising steps with a guidance scale of 3.5. For **SDXL**, we set $\tau = 400$, and the structure guidance strength to $\gamma_t = 1 - \bar{\alpha}_t$. This configuration uses 50 denoising steps with a classifier-free guidance (CFG) [14] scale of 7.5. All experiments are conducted on a single NVIDIA A6000 GPU.

Baselines. We compare our method with recent training-free methods (ScaleCrafter [12], HiDiffusion [49], DiffuseHigh [20], FreeScale [31], DemoFusion [8], AccDiffusion v2 [24]), super-resolution models (BSRGAN [48], OSEDiff [46]), and a training-based model UltraPixel [33]. Training-free

Table 2: **Quantitative comparison results.** The best results are shown in **bold**, and the second best results are underlined. All time measurements are expressed in seconds.

Model	Resolution	Method	FID ↓	KID ↓	IS ↑	FID _p ↓	KID _p ↓	IS _p ↑	CLIP ↑	Time ↓		
SDXL	2048 ²	SDXL Direct [30]	88.56	0.0124	13.25	58.73	0.0137	20.79	31.57	47		
		SDXL + BSRGAN [48]	64.60	0.0041	18.40	41.40	0.0092	23.19	33.03	13		
		SDXL + OSEDiff [46]	64.79	0.0046	18.89	41.76	0.0094	23.58	<u>32.79</u>	<u>29</u>		
		UltraPixel [33]	64.61	0.0056	18.58	42.44	0.0093	25.15	32.61	71		
		ScaleCrafter [12]	68.68	0.0033	16.56	43.46	0.0064	23.52	32.07	64		
		HiDiffusion [49]	69.52	0.0040	18.22	42.92	0.0067	24.01	31.50	33		
		DiffuseHigh [20]	<u>63.27</u>	0.0033	19.10	38.15	0.0062	24.95	32.77	45		
		FreeScale [31]	63.50	0.0031	19.06	38.27	<u>0.0062</u>	24.25	32.62	69		
		AccDiffusion v2 [24]	64.86	0.0039	18.37	38.24	0.0068	25.66	32.62	199		
		Demofusion [8]	63.36	0.0032	<u>19.15</u>	35.98	0.0050	26.42	32.72	125		
		ScaleDiff (Ours)	62.98	<u>0.0032</u>	19.54	<u>38.03</u>	0.0067	<u>25.70</u>	33.11	31		
		SDXL	4096 ²	SDXL Direct [30]	182.05	0.0717	7.99	80.80	0.0250	17.68	27.82	328
				SDXL + BSRGAN [48]	64.88	0.0044	18.16	48.97	0.0160	17.04	<u>33.02</u>	14
				SDXL + OSEDiff [46]	65.35	0.0045	18.69	45.67	0.0118	17.61	32.88	122
UltraPixel [33]	65.39			0.0055	19.08	47.09	0.0112	20.64	32.33	386		
ScaleCrafter [12]	86.66			0.0110	15.14	79.39	0.0217	14.47	30.25	932		
HiDiffusion [49]	105.37			0.0216	13.87	112.30	0.0494	12.22	27.21	124		
DiffuseHigh [20]	<u>63.91</u>			0.0034	18.99	42.30	0.0079	19.54	32.68	325		
FreeScale [31]	64.33			0.0036	<u>19.18</u>	<u>39.56</u>	0.0079	18.91	32.56	517		
AccDiffusion v2 [24]	64.64			0.0037	18.56	40.92	0.0083	18.42	32.34	1599		
Demofusion [8]	65.06			0.0041	19.13	41.29	<u>0.0079</u>	19.59	32.61	1005		
ScaleDiff (Ours)	61.87			0.0025	19.56	38.89	0.0080	<u>20.41</u>	33.04	<u>113</u>		
FLUX	2048 ²			FLUX Direct [21]	68.78	0.0069	<u>18.57</u>	42.84	0.0086	22.46	30.79	150
				FLUX + BSRGAN [48]	<u>64.65</u>	<u>0.0052</u>	19.07	42.01	0.0081	22.98	<u>31.21</u>	33
				FLUX + OSEDiff [46]	65.10	0.0056	18.46	<u>41.88</u>	<u>0.0078</u>	<u>23.25</u>	31.03	46
		ScaleDiff (Ours)	64.31	0.0047	18.51	40.03	0.0073	23.38	31.22	103		
		FLUX	4096 ²	FLUX Direct [21]	459.07	0.2775	1.61	367.47	0.2642	1.22	18.03	1251
				FLUX + BSRGAN [48]	64.76	0.0051	<u>18.84</u>	49.30	0.0125	16.92	31.19	34
FLUX	4096 ²	FLUX + OSEDiff [46]	<u>64.22</u>	<u>0.0052</u>	19.16	<u>48.37</u>	<u>0.0112</u>	<u>16.99</u>	31.13	<u>136</u>		
		ScaleDiff (Ours)	64.06	0.0044	18.36	44.29	0.0098	17.41	<u>31.14</u>	407		

baselines are evaluated on SDXL, as they are optimized for U-Net architectures. For FLUX, we compare against the base model (natively supports resolutions up to 2048²) and SR methods.

Evaluation. For quantitative evaluation, we randomly sample 1,000 image-text pairs from the LAION-5B [38] dataset and generate one image per prompt using each method. We compute the Fréchet Inception Distance (FID) [13], Kernel Inception Distance (KID) [4], and Inception Score (IS) [37] between generated images and real images. However, these metrics typically require resizing images to 299² pixels, thereby limiting the evaluation of fine-grained details. To better assess detail fidelity, we extract multiple patches from each image and calculate patch-level FID_p, KID_p, and IS_p following [8]. We also measure the CLIP Score [32] to evaluate text-image alignment.

4.2 Quantitative Comparison

Table 2 compares ScaleDiff with baseline methods for generating images at 2048² and 4096² resolutions. On SDXL, ScaleDiff consistently outperforms existing training-free, training-based, and super-resolution methods across key quality metrics, demonstrating its ability to generate high-fidelity images. Similar results on FLUX further confirm ScaleDiff’s robustness and model-agnostic design.

ScaleDiff also achieves remarkable inference efficiency. For 4096² resolution on SDXL, it requires only 113 seconds—the fastest among training-free methods. Compared to the patch-based method Demofusion, ScaleDiff achieves an 8.9× speedup while surpassing it in most evaluation metrics, demonstrating NPA’s effectiveness. On FLUX, applying NPA yields a 3.1× speedup over direct inference at 4096² resolution. While super-resolution models like BSRGAN offer faster inference, they struggle to produce fine details, as reflected in lower patch-level scores. In contrast, ScaleDiff successfully balances generative quality with computational efficiency.

4.3 Qualitative Comparison

Figure 5 presents a qualitative comparison of ScaleDiff with baseline methods for 4096² image generation. While all methods produce high-quality outputs, prior approaches exhibit notable limitations. Super-resolution models like BSRGAN and OSEDiff fail to reproduce fine details, resulting in visibly corrupted facial features (Figure 5b,c). DemoFusion effectively generates fine

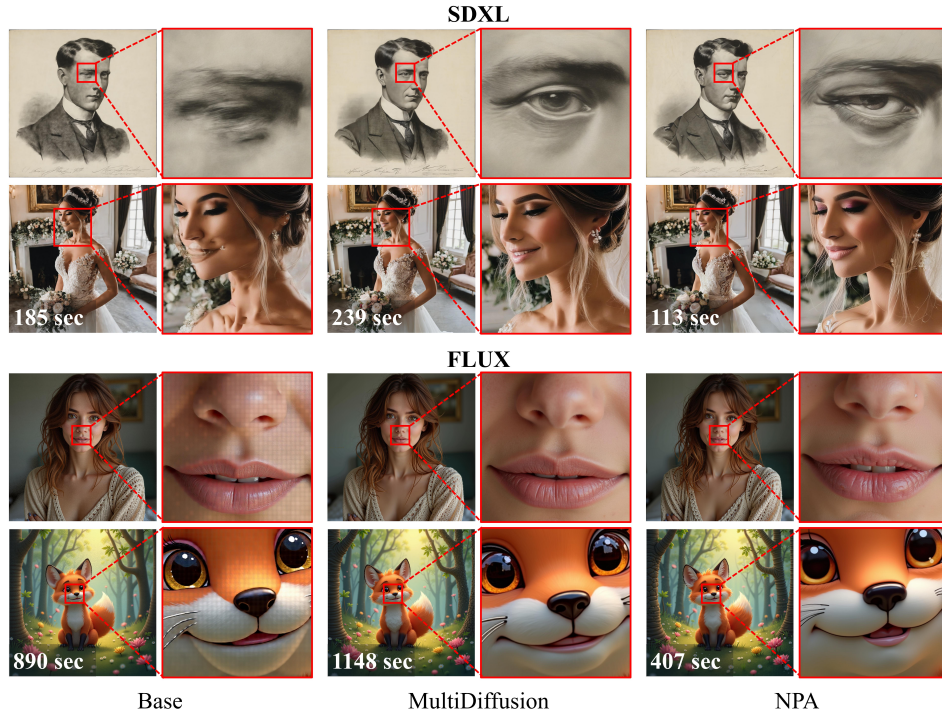


Figure 6: **Qualitative comparison of replacing NPA in the ScaleDiff pipeline.** Inference time for each method is shown in the bottom left. All images are generated at 4096^2 resolution.

Table 3: **Quantitative results of ablation study.**

Attention	<i>LFM</i>	<i>SG</i>	FID ↓	KID ↓	IS ↑	FID _p ↓	KID _p ↓	IS _p ↑	CLIP ↑	Time ↓
Base	✓	✓	61.91	0.0028	19.47	39.94	0.0082	20.09	33.01	185
MultiDiffusion	✓	✓	61.71	0.0021	19.71	38.08	0.0069	20.85	33.04	239
NPA	✓	✓	<u>61.87</u>	<u>0.0025</u>	<u>19.56</u>	<u>38.89</u>	<u>0.0080</u>	<u>20.41</u>	33.04	113
NPA			64.17	0.0036	<u>19.49</u>	41.55	0.0092	19.41	33.02	113
NPA	✓		<u>62.34</u>	0.0028	19.19	<u>39.49</u>	<u>0.0085</u>	<u>20.16</u>	33.01	113
NPA		✓	64.12	0.0035	18.86	41.50	0.0091	19.71	<u>33.04</u>	113
NPA	✓	✓	61.87	0.0025	19.56	38.89	0.0080	20.41	33.04	113

details but often suffers from repetitive object patterns due to its patch-based processing (Figure 5c). DiffuseHigh lacks detailed textures due to inherent constraints of RGB-space upsampling (Figure 5a). In contrast, ScaleDiff produces results with improved global structure and finer details, demonstrating superior qualitative performance across different models.

4.4 Ablation Study

Effectiveness of NPA. We validate the effectiveness of our proposed Neighborhood Patch Attention (NPA) by comparing it against two alternatives integrated into the ScaleDiff pipeline: (1) direct high-resolution inference (Base) and (2) a standard patch-based method (MultiDiffusion). As shown in Figure 6 and Table 3, all methods maintain a stable global structure, likely due to the shared ScaleDiff pipeline. However, the Base method produces local artifacts on SDXL and lacks fine details on FLUX, while requiring significantly longer inference times. MultiDiffusion generates high-quality images and achieves the best scores, but suffers from substantial computational overhead (1148s on FLUX). In contrast, our NPA achieves scores comparable to MultiDiffusion while being more efficient (407s on FLUX, $2.8\times$ speedup), demonstrating an effective balance between generation quality and computational efficiency.

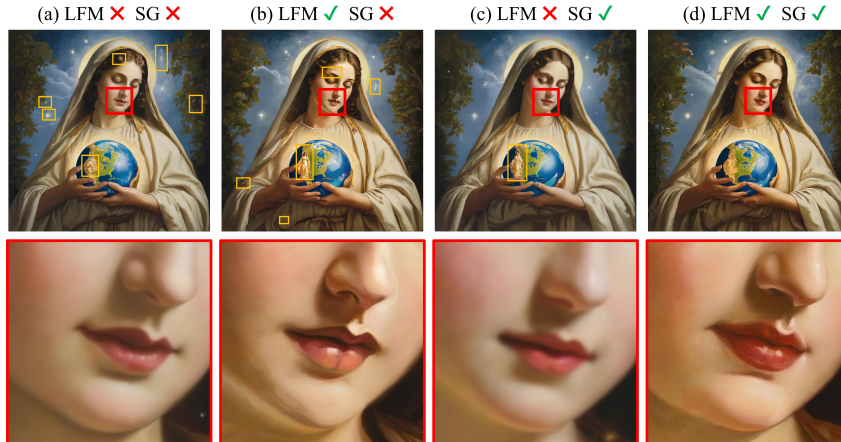


Figure 7: Ablating each component of ScaleDiff. The yellow box highlights the repetition artifacts. All images are generated at 4096^2 using SDXL [30]. Zoom in for a better view.

Table 4: Ablating noise timestep τ .

Model	τ	FID \downarrow	KID \downarrow	IS \uparrow	FID _p \downarrow	KID _p \downarrow	IS _p \uparrow	CLIP \uparrow
SDXL	700	63.61	0.0036	18.93	37.65	0.0064	25.02	33.01
	600	63.60	0.0034	19.20	<u>37.94</u>	<u>0.0067</u>	25.15	33.06
	500	63.30	0.0033	19.43	38.77	0.0073	24.42	33.09
	400	63.12	0.0032	19.36	38.33	0.0070	24.56	<u>33.07</u>
	300	<u>63.22</u>	<u>0.0032</u>	<u>19.56</u>	39.38	0.0077	23.85	33.07
FLUX	700	64.44	0.0049	18.22	40.37	0.0073	23.62	31.27
	600	64.45	0.0049	18.52	<u>40.76</u>	<u>0.0076</u>	23.16	<u>31.29</u>
	500	64.13	0.0049	18.36	41.72	0.0081	22.78	31.29
	400	63.59	0.0047	<u>18.45</u>	41.99	0.0083	22.90	31.25
	300	<u>63.67</u>	<u>0.0048</u>	18.30	42.07	0.0084	<u>23.24</u>	31.33

Effectiveness of LFM and SG. In Figure 7 and Table 3, we validate the contributions of Latent Frequency Mixing (LFM) and Structure Guidance (SG). When both components are removed (Fig. 7a), the model fails to generate coherent results, exhibiting severe object repetition and heavily oversmoothed textures. Adding LFM alone (Fig. 7b) reduces oversmoothing and enables the synthesis of finer details, which is confirmed by improvements in patch-level metrics in Table 3. Applying SG alone (Fig. 7c) effectively mitigates object repetition, demonstrating its role in enforcing global structural coherence. The full ScaleDiff pipeline (Fig. 7d), which combines both LFM and SG, concurrently addresses both issues and achieves the best overall performance among all ablated configurations.

Ablation of Noise Timestep τ . The noise timestep τ is a critical hyperparameter governing the trade-off between preserving global structure from the upsampled reference image (lower τ) and enabling sufficient synthesis of fine-grained details (higher τ). We conduct an ablation study to determine the optimal τ for both architectures. As detailed in Table 4, $\tau = 400$ for SDXL and $\tau = 600$ for FLUX provide the best balance between structural fidelity and detail generation.

5 Conclusion

In this paper, we propose ScaleDiff, an efficient and model-agnostic framework that enhances the resolution capabilities of pretrained diffusion models without requiring additional training. We introduce Neighborhood Patch Attention (NPA), a mechanism that significantly reduces the computational redundancy typical of traditional patch-based diffusion approaches. In addition, we propose Latent Frequency Mixing (LFM) and incorporate Structure Guidance (SG) within an upsample–diffuse–denoise pipeline to improve fine detail synthesis and structural consistency. Our experiments, conducted on both U-Net and Diffusion Transformer architectures, show that ScaleDiff achieves state-of-the-art performance among training-free methods, delivering superior image quality and faster inference across diverse models. These results highlight ScaleDiff as a powerful and versatile solution for higher-resolution image generation.

Acknowledgment. This was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government(MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program(Hanyang University)) and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government(MSIT) (No.RS-2025-02215122, Development and Demonstration of Lightweight AI Model for Smart Homes).

References

- [1] S. AI. Stable diffusion: A latent text-to-image diffusion model. <https://stability.ai/blog/stable-diffusion-public-release>, 2022.
- [2] O. Bar-Tal, L. Yariv, Y. Lipman, and T. Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023.
- [3] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023.
- [4] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.
- [5] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, et al. Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.
- [6] J. Chen, C. Ge, E. Xie, Y. Wu, L. Yao, X. Ren, Z. Wang, P. Luo, H. Lu, and Z. Li. Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pages 74–91. Springer, 2024.
- [7] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [8] R. Du, D. Chang, T. Hospedales, Y.-Z. Song, and Z. Ma. Demofusion: Democratising high-resolution image generation with no \$\$\$\$. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6159–6168, 2024.
- [9] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.
- [10] P. Gao, L. Zhuo, D. Liu, R. Du, X. Luo, L. Qiu, Y. Zhang, C. Lin, R. Huang, S. Geng, et al. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. *arXiv preprint arXiv:2405.05945*, 2024.
- [11] L. Guo, Y. He, H. Chen, M. Xia, X. Cun, Y. Wang, S. Huang, Y. Zhang, X. Wang, Q. Chen, et al. Make a cheap scaling: A self-cascade diffusion model for higher-resolution adaptation. In *European Conference on Computer Vision*, pages 39–55. Springer, 2024.
- [12] Y. He, S. Yang, H. Chen, X. Cun, M. Xia, Y. Zhang, X. Wang, R. He, Q. Chen, and Y. Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.
- [13] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [14] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [15] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

- [16] L. Huang, R. Fang, A. Zhang, G. Song, S. Liu, Y. Liu, and H. Li. Fouriscale: A frequency perspective on training-free high-resolution image synthesis. In *European Conference on Computer Vision*, pages 196–212. Springer, 2024.
- [17] J. Hwang, Y.-H. Park, and J. Jo. Upsample guidance: Scale up diffusion models without training. *arXiv preprint arXiv:2404.01709*, 2024.
- [18] J. Jeong, S. Han, J. Kim, and S. J. Kim. Latent space super-resolution for higher-resolution image generation with diffusion models. *arXiv preprint arXiv:2503.18446*, 2025.
- [19] Z. Jin, X. Shen, B. Li, and X. Xue. Training-free diffusion model adaptation for variable-sized text-to-image synthesis. *Advances in Neural Information Processing Systems*, 36:70847–70860, 2023.
- [20] Y. Kim, G. Hwang, J. Zhang, and E. Park. Diffusehigh: Training-free progressive high-resolution image synthesis through structure guidance. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 4338–4346, 2025.
- [21] B. F. Labs. Flux.1: An advanced generative ai model. <https://flux1.io/>, 2024.
- [22] Y. Lee, K. Kim, H. Kim, and M. Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. *Advances in Neural Information Processing Systems*, 36:50648–50660, 2023.
- [23] Z. Lin, M. Lin, M. Zhao, and R. Ji. Accdiffusion: An accurate method for higher-resolution image generation. In *European Conference on Computer Vision*, pages 38–53. Springer, 2024.
- [24] Z. Lin, M. Lin, W. Zhan, and R. Ji. Accdiffusion v2: Towards more accurate higher-resolution diffusion extrapolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [25] S. Liu, W. Yu, Z. Tan, and X. Wang. Linfusion: 1 gpu, 1 minute, 16k image. *arXiv preprint arXiv:2409.02097*, 2024.
- [26] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- [27] Z. Lu, Z. Wang, D. Huang, C. Wu, X. Liu, W. Ouyang, and L. Bai. Fit: Flexible vision transformer for diffusion model. *arXiv preprint arXiv:2402.12376*, 2024.
- [28] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon. Sedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.
- [29] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- [30] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- [31] H. Qiu, S. Zhang, Y. Wei, R. Chu, H. Yuan, X. Wang, Y. Zhang, and Z. Liu. Freescale: Unleashing the resolution of diffusion models via tuning-free scale fusion. *arXiv preprint arXiv:2412.09626*, 2024.
- [32] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [33] J. Ren, W. Li, H. Chen, R. Pei, B. Shao, Y. Guo, L. Peng, F. Song, and L. Zhu. Ultrapixel: Advancing ultra-high-resolution image synthesis to new peaks. *arXiv preprint arXiv:2407.02158*, 2024.
- [34] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

- [35] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
- [36] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494, 2022.
- [37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.
- [38] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. 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.
- [39] S. Shi, W. Li, Y. Zhang, J. He, B. Gong, and Y. Zheng. Resmaster: Mastering high-resolution image generation via structural and fine-grained guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6887–6895, 2025.
- [40] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.
- [41] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [42] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [43] J. Teng, W. Zheng, M. Ding, W. Hong, J. Wangni, Z. Yang, and J. Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. *arXiv preprint arXiv:2309.03350*, 2023.
- [44] A. Tragakis, M. Aversa, C. Kaul, R. Murray-Smith, and D. Faccio. Is one gpu enough? pushing image generation at higher-resolutions with foundation models, 2024. URL <https://arxiv.org/abs/2406.07251>.
- [45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [46] R. Wu, L. Sun, Z. Ma, and L. Zhang. One-step effective diffusion network for real-world image super-resolution. *arXiv preprint arXiv:2406.08177*, 2024.
- [47] E. Xie, L. Yao, H. Shi, Z. Liu, D. Zhou, Z. Liu, J. Li, and Z. Li. Diffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4230–4239, 2023.
- [48] K. Zhang, J. Liang, L. Van Gool, and R. Timofte. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4791–4800, 2021.
- [49] S. Zhang, Z. Chen, Z. Zhao, Y. Chen, Y. Tang, and J. Liang. Hidiffusion: Unlocking higher-resolution creativity and efficiency in pretrained diffusion models. In *European Conference on Computer Vision*, pages 145–161. Springer, 2024.

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: All claims made in the paper are thoroughly examined, with corresponding explanations provided for each.

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: Limitations are provided in supplementary materials

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: [Yes]

Justification: We detail how FLOPs is calculated in section 3.1 and 3.2

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: Detailed explanation of experimental setup is provided, see section 4.1

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: [No]

Justification: We will release our code after submission.

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We do not train any models. Details about evaluation is provided in section 3.1. We cannot share subset for evaluation because we do not hold the rights.

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: [No]

Justification: We build upon prior research in the field, utilizing established evaluation metrics and conducting qualitative analysis for comparison.

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: GPU used for evaluation is provided. Inference time is also provided

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 utilize a pre-trained diffusion model, which—like any generative model—must be used with caution. In our work, we do not fine-tune or modify the model in any way.

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: [No]

Justification: Our approach preserves the original societal implications of the diffusion model, given that we do not modify its generation behavior.

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: [No]

Justification: We utilize a pre-trained model without making any modifications to its weights.

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: All sources are properly cited, and the pre-trained model is publicly available for research use.

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: Our paper does not release 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: [No]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

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: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

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: We used LLM only for editing (e.g., grammar, spelling, word choice).

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.

A Additional Implementation Details

When generating images with various aspect ratios, we ensure that the longer side of the initial image matches the model’s trained resolution. For frequency decomposition, we use a spatial downsampling ratio of 4 for FLUX and 8 for SDXL. When evaluating the inference speed of MultiDiffusion [2], the overlap ratio is set to 50%, and we use a batch size of 16 for SDXL and 1 for FLUX.

In U-Net-based models such as SDXL [30], the spatial resolution is progressively downsampled across layers. Accordingly, we also downsample the native resolution (h, w) of NPA to match the downsampling ratio at each corresponding layer. In FLUX [21], the MM-DiT architecture [9] is used, where the text tokens are concatenated with latent tokens and jointly processed through a self-attention layer. Accordingly, in NPA, the text tokens are duplicated for each patch and concatenated with the corresponding latent tokens within each patch. After the NPA processing, the text tokens are averaged across all patches. In the original setting, text tokens are assigned the position $(0, 0)$ in RoPE [42]. When duplicating the text tokens, we assign them the position of the top-left corner of the corresponding Key/Value patch to ensure proper spatial processing.

B Additional Details and Experiments on Neighborhood Patch Attention

B.1 Query Window Random Shifting

While Neighborhood Patch Attention utilizes overlapping key/value patches to ensure a smooth transition at patch boundaries, minor boundary artifacts can sometimes appear in the generated output. **Query Window Random Shifting** is an optional technique designed to further alleviate such artifacts by introducing random variations to the query patch grid at each layer (Figure 8). Specifically, we randomly sample the top and left offsets for padding uniformly from the respective ranges $[0, \frac{h}{2}]$ and $[0, \frac{w}{2}]$. The query tensor is then zero-padded by a total of $\frac{h}{2}$ in height and $\frac{w}{2}$ in width using these randomly sampled top-left offsets. Query patches are subsequently extracted from this enlarged canvas. After attention computation, regions corresponding to the added padding are discarded. Since offsets are independently resampled at each layer, explicit patch boundaries are avoided, which reduces border artifacts with minimal computational overhead. Note that this technique was not used during the evaluation in this paper.

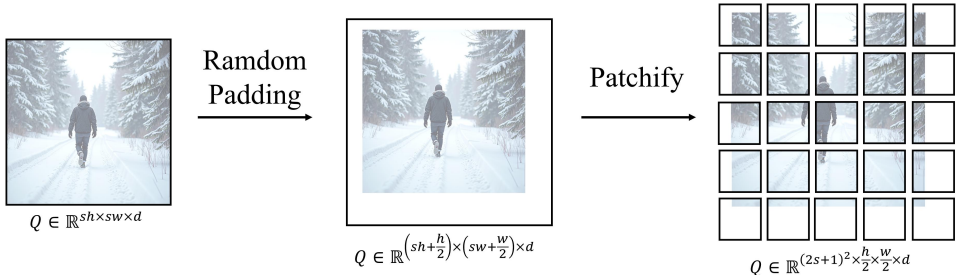


Figure 8: Illustration of Query Window Random Shifting.

B.2 Comparison of Generation time

Figure 9 presents the model processing time at various resolutions for SDXL and FLUX, comparing three methods: Direct Inference (Base), MultiDiffusion, and NPA. Direct Inference shows a quadratic growth in processing time as resolution increases, primarily due to the cost of global self-attention. MultiDiffusion achieves linear scaling with resolution but suffers from higher baseline overhead, caused by redundant computation on overlapping patches. In contrast, NPA eliminates such redundancy and maintains linear scaling, resulting in the lowest processing time.

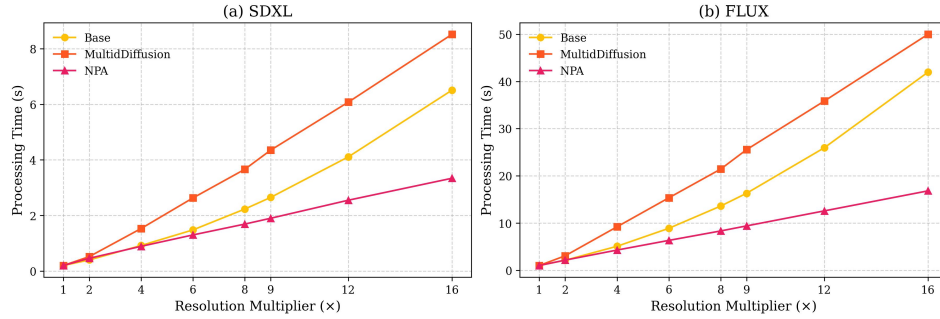


Figure 9: **Comparison of model processing time.** A resolution multiplier of $1\times$ corresponds to generation at 1024^2 resolution.

B.3 Panorama Generation with NPA

Our NPA adopts the behavior of MultiDiffusion, making it suitable for a wide range of applications. Notably, Figure 10 presents the results of using NPA for panorama generation on FLUX. No other ScaleDiff components were used.

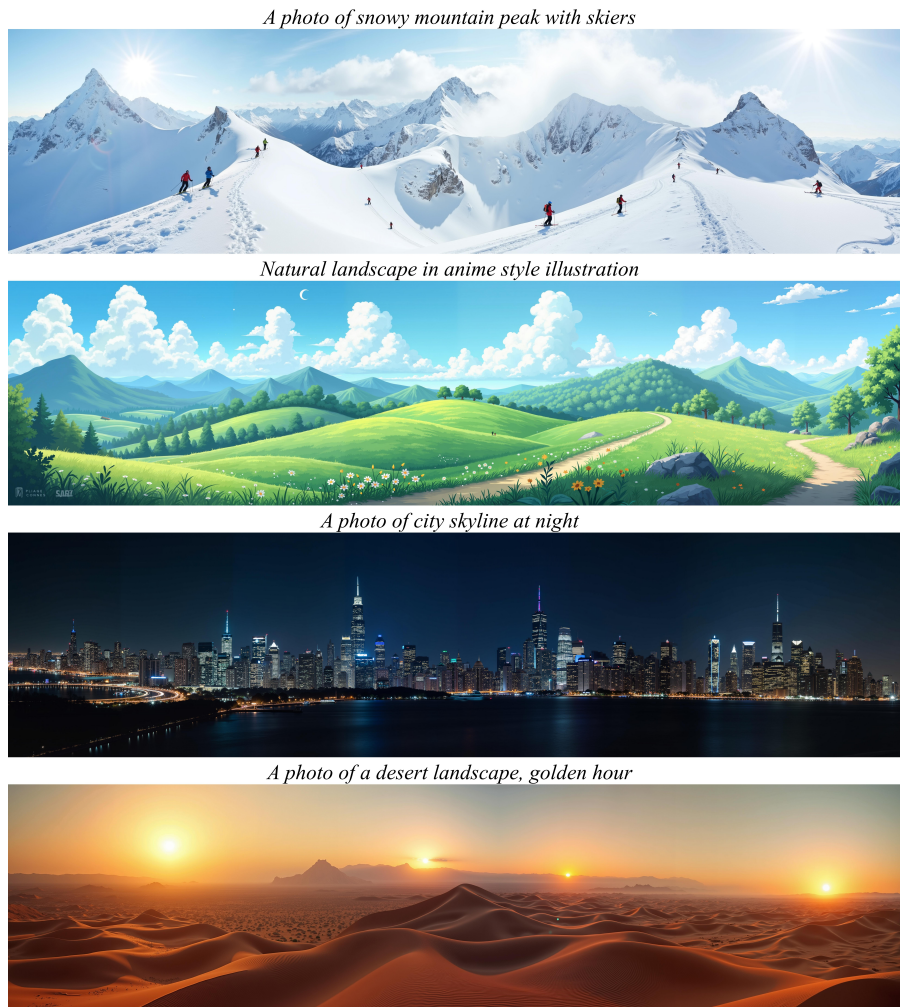


Figure 10: **Panorama Generation with NPA.** All images are generated at 1024×4096 resolution.

B.4 Patch Extraction Algorithm of NPA

Algorithm 1 presents the detailed patch extraction procedure for NPA. Note that query window random shifting (Section B.1) is not included in this algorithm.

Algorithm 1 NPA: Query/Key/Value Patch Extraction

```

1: Input:  $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{mh \times nw \times d}$  ▷ Full query, key, value tensor
2: Parameters:  $h, w$  ▷ Native height and width
3: Output:  $\{\mathbf{Q}_i\}_{i=1}^N$  ▷ Set of non-overlapping query patches
4:  $\{\mathbf{K}_i\}_{i=1}^N, \{\mathbf{V}_i\}_{i=1}^N$  ▷ Set of overlapping key, value patches

5:  $N_r \leftarrow \frac{mh-h/2}{h/2} + 1$  ▷ Number of patch rows
6:  $N_c \leftarrow \frac{nw-w/2}{w/2} + 1$  ▷ Number of patch columns
7:  $N \leftarrow N_r \times N_c$  ▷ Total number of patches
8: for  $i \leftarrow 1$  to  $N$  do
9:  $h_{start}^q \leftarrow \lfloor i/N_r \rfloor \times \frac{h}{2}$  ▷ Top-left coordinate of the query patch
10:  $w_{start}^q \leftarrow (i \bmod N_r) \times \frac{w}{2}$ 
11:  $h_{end}^q \leftarrow h_{start}^q + \frac{h}{2}$  ▷ Bottom-right coordinate of the query patch
12:  $w_{end}^q \leftarrow w_{start}^q + \frac{w}{2}$ 
13:  $h_{start}^{kv} \leftarrow \text{clamp}(h_{start}^q - \frac{h}{4}, 0, sh - h)$  ▷ Center K/V patch around query patch
14:  $w_{start}^{kv} \leftarrow \text{clamp}(w_{start}^q - \frac{w}{4}, 0, sw - w)$  ▷ Clamp for window shifting at the edge
15:  $h_{end}^{kv} \leftarrow h_{start}^{kv} + h$ 
16:  $w_{end}^{kv} \leftarrow w_{start}^{kv} + w$ 
17:  $\mathbf{Q}_i \leftarrow \mathbf{Q}[h_{start}^q : h_{end}^q, w_{start}^q : w_{end}^q, :]$  ▷ Non-overlapping query patch extraction
18:  $\mathbf{K}_i \leftarrow \mathbf{K}[h_{start}^{kv} : h_{end}^{kv}, w_{start}^{kv} : w_{end}^{kv}, :]$  ▷ Overlapping key, value patch extraction
19:  $\mathbf{V}_i \leftarrow \mathbf{V}[h_{start}^{kv} : h_{end}^{kv}, w_{start}^{kv} : w_{end}^{kv}, :]$ 
20: end for
21: return  $\{\mathbf{Q}_i\}_{i=1}^N, \{\mathbf{K}_i\}_{i=1}^N, \{\mathbf{V}_i\}_{i=1}^N$ 

```

C Qualitative Results on Various Models

We present qualitative results of ScaleDiff using SDXL and FLUX across various aspect ratios and resolutions in Figure 11 and Figure 12. To highlight the model-agnostic nature of our method, we also include results using Lumina-T2X [10] in Figure 13.

D Limitation

ScaleDiff has some limitations. First, as a tuning-free framework, its performance is inherently constrained by the capabilities of the underlying diffusion model. Second, being a patch-based approach, it relies heavily on the diffusion model’s prior knowledge of cropped image regions. This can sometimes lead to inconsistent local content when generating sharp close-up images. Finally, repetitive artifacts may still occur in background regions, a common drawback of patch-based generation methods.



Figure 11: Qualitative results of ScaleDiff on SDXL



Figure 12: Qualitative results of ScaleDiff on FLUX

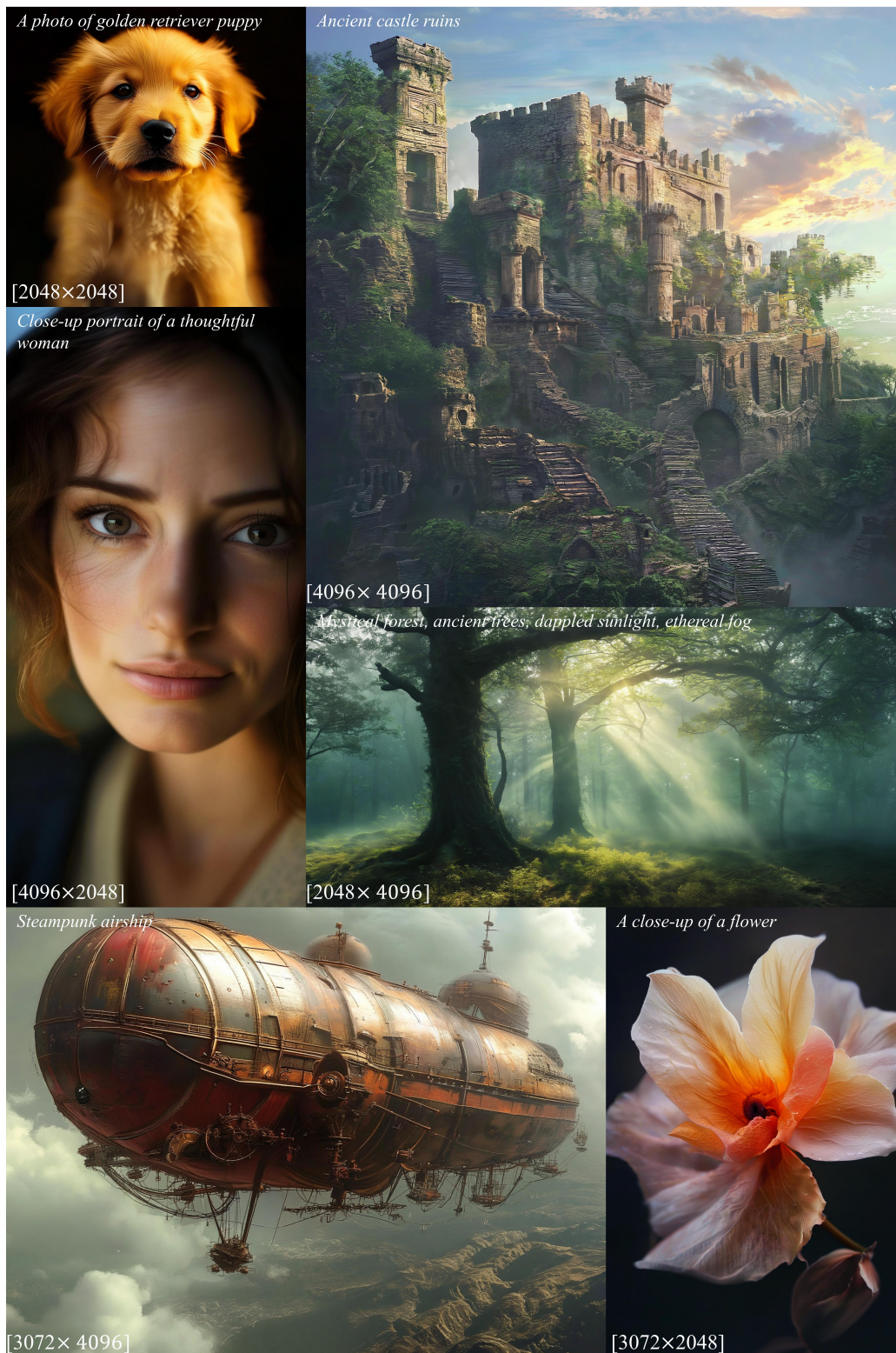


Figure 13: Qualitative results of ScaleDiff on Lumina-T2X