Low-Resolution Editing is All You Need for High-Resolution Editing

Junsung Lee[1*] Hyunsoo Lee[1*] Yong Jae Lee[3] Bohyung Han[1,2]
[1]ECE & [2]IPAI, Seoul National University [3]University of Wisconsin-Madison
{leejs0525, philip21, bhhan}@snu.ac.kr, yongjaelee@cs.wisc.edu

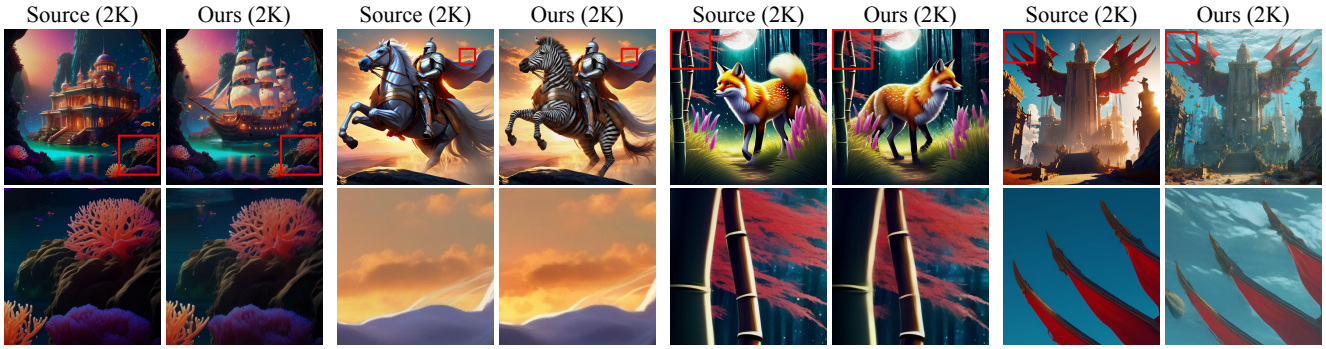


Figure 1. Image editing results of the proposed method, ScaleEdit. Our method successfully generates high-resolution edited images by leveraging low-resolution editing results as reference images.

Abstract

High-resolution content creation is rapidly emerging as a central challenge in both the vision and graphics communities. Images serve as the most fundamental modality for visual expression, and content generation that aligns with the user intent requires effective, controllable high-resolution image manipulation mechanisms. However, existing approaches remain limited to low-resolution settings, typically supporting only up to 1K resolution. In this work, we introduce the task of high-resolution image editing and propose a test-time optimization framework to address it. Our method performs patch-wise optimization on high-resolution source images, followed by a fine-grained detail transfer module and a novel synchronization strategy to maintain consistency across patches. Extensive experiments show that our method produces high-quality edits, facilitating high-resolution content creation.

1. Introduction

High-resolution visual content has become essential in modern digital workflows, including image synthesis [11, 16, 21, 38, 60], video generation [3, 6, 20], document processing [33], autonomous driving [57], and 3D shape generation [49, 50]. In such scenarios, users often seek precise and intentional modifications to an existing high-resolution image rather than generating a new one from scratch. Unlike naïve image generation, image manipulation preserves the identity and semantic layout of the source image, while allowing controlled adjustments that reflect the user's intention. Therefore, editing serves as a reliable strategy for producing user-controlled high-resolution content while preserving both structural integrity and visual fidelity. This motivates the need for editing methods that maintain fine-grained texture details and structural consistency at high resolutions (*e.g.*, $> 1\text{K} \approx 1024^2$).

We introduce the task of *high-resolution image editing*, where the objective is to modify a high-resolution source image according to a target visual concept while preserving its fine-grained details. However, existing image-to-image translation methods [8, 30, 63] operate at fixed low resolutions and cannot directly handle large-scale inputs. A naïve solution is to perform editing at low resolution and subsequently apply super-resolution methods [7, 10, 12, 48], but this approach fails to recover micro-scale textures since the fine-grained details of the source image are not conditioned during the editing process. Thus, a novel strategy is required to support editing in high-resolution.

*Equal contribution.

To address the task defined in our work, we propose *ScaleEdit*. Editing a high resolution image requires an optimization that preserves both semantics and fine-grained details. This can be facilitated by leveraging the strong performance of existing low-resolution image editing methods [8, 30, 63]. Under this view, the central challenge of the task becomes clear: how to faithfully transfer the fine-scale details of the high-resolution source image into the edited result. Our intuition is that this objective can be achieved by exploiting the generative priors. Specifically, we constrain optimization to the sampling process of the pretrained generative models [25, 37, 41], implicitly enforcing the generative prior as a structural constraint that ensures plausible results.

Concretely, our key idea is to transfer fine-grained details from the high-resolution source image to the target image by introducing a learnable transfer function defined in the intermediate feature space of a pretrained generative model. This transfer function operates as a learnable 1×1 convolution, enabling precise control over fine-grained detail transfer. However, generative models are trained at fixed input resolutions and cannot operate directly on high-resolution image setups. Thus, to fully inherit their strong generative priors while accommodating large-scale inputs, we adopt a patch-wise strategy, dividing the source image into model-native resolution regions. This enables preserving micro-scale details while still benefiting from the priors encoded in the pretrained generative model. A novel synchronization mechanism is then applied to ensure global consistency between patches. We note that the proposed synchronization method does not require overlapping inference, effectively decreasing the computational cost of the overall method. We summarize our contributions as follows:

- To the best of our knowledge, our work is the first to propose the high-resolution image editing task, supported by our novel patch-wise inference mechanism.
- We develop a feature transfer function and a synchronization strategy across non-overlapping patches to perform high-resolution editing in a test-time optimization manner.
- Experimental results demonstrate that the proposed method achieves state-of-the-art editing capacity that enable diverse instruction-based editing.

2. Related work

Text-driven image editing. Text-conditional image editing methods inherit the strong generative capabilities of pretrained text-to-image models [14, 25, 41, 43]. Building on these priors, approaches using U-Net-based [42] diffusion model achieve plausible performance across diverse scenarios via latent optimization, attention manipulation, and instruction-based control [2, 4, 5, 9, 13, 17, 23, 26–28, 34, 51].

More recently, generative models with transformer-based architectures [52] have further advanced editing capabilities. For example, Step1X-Edit [30] presents a general-purpose instruction-based image editing framework by coupling a multi-modal LLM with a DiT-based [36] decoder trained on a large-scale taxonomy-driven editing dataset. ICEdit [62] leverages the in-context generation capability of large DiT models, treating edits as diptych-style prompts and only requires lightweight LoRA-MoE fine-tuning [19]. KV-Edit [63] enables training-free editing by reusing cached key–value tokens under a user-provided mask, which is an idea inspired by LLM's KV-cache mechanisms [58], to preserve background pixels while effectively modifying the foreground regions. However, these methods can edit images only up to 1K resolution, whereas our approach leverages the generative priors of existing methods to enable editing even at 2K and higher resolutions.

Diffusion-based super-resolution. Real-World Image Super Resolution (Real-ISR) seeks to reconstruct detailed high-resolution images from low-resolution inputs that exhibit diverse and unknown degradations. With the rapid advancement of diffusion models, diffusion-based approaches have increasingly been adopted to tackle this task. Prior methods [1, 7, 10, 12, 22, 48, 53–56, 59] built on large text-to-image models have recently shown strong performance. For instance, TSD-SR [10] distills a multi-step text-to-image diffusion model into a one-step super-resolution network through target score distillation and distribution-aware sampling. PiSA-SR [48] introduces dual LoRA [19] modules separately dedicated to pixel-level regression and semantic-level enhancement. On the other hand, DiT-SR [7] introduces a diffusion transformer [36] trained from scratch with a U-shaped multi-scale design and frequency-aware timestep conditioning. DiT4SR [12] adapts the Stable Diffusion 3 [14] architecture by injecting low-resolution features directly into transformer [52] blocks via bidirectional attention and convolution. While existing SR methods rely on training to generate high-resolution outputs, our method restores fine details through test-time optimization without any model training.

3. Preliminary

Diffusion models. Diffusion models [18, 44–46] demonstrate a remarkable ability to synthesize versatile content in multiple modalities. Starting from an initial noisy latent $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, they generate a trajectory of latent variables $\{\mathbf{x}_t^{\text{rev}}\}_{t=T}^{0}$ through a reverse diffusion process. DDIM [45] formulates this reverse process as a deterministic procedure. Specifically, $\mathbf{x}_{t-1}^{\text{rev}}$ is sampled from $\mathbf{x}_t^{\text{rev}}$ as:

$$\begin{aligned}\mathbf{x}_{t-1}^{\text{rev}} &= f^{\text{rev}}(\mathbf{x}_t^{\text{rev}}, t) \\ &:= \sqrt{\alpha_{t-1}}\hat{\mathbf{x}}_0(\mathbf{x}_t^{\text{rev}}, t) + \sqrt{1-\alpha_{t-1}}\epsilon_\theta(\mathbf{x}_t^{\text{rev}}, t) \quad (1)\end{aligned}$$

where $\{\alpha_t\}_{t=0}^{T}$ is the predefined decreasing variance schedule, $\epsilon_\theta(\cdot,\cdot)$ denotes the pretrained noise prediction network,

and $\hat{\mathbf{x}}_0(\mathbf{x}_t, t)$ is the estimated Tweedie at timestep t, computed with the Tweedie's Formula [47] given by

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}}. \tag{2}$$

Iteratively applying Eq. (1) gives a clean data sample $\mathbf{x}_0$.

Image-to-image translation. In diffusion-based image-to-image translation [5, 34, 51], deterministic forward and reverse processes [45] are commonly employed to reconstruct a given source image. Starting from a source image $\mathbf{x}_0$, the forward process introduces noise according to

$$\begin{aligned}\mathbf{x}_{t+1}^{\text{fwd}} &= f^{\text{fwd}}(\mathbf{x}_t^{\text{fwd}}, t) \\ &:= \sqrt{\alpha_{t+1}}\hat{\mathbf{x}}_0(\mathbf{x}_t^{\text{fwd}}, t) + \sqrt{1-\alpha_{t+1}}\epsilon_\theta(\mathbf{x}_t^{\text{fwd}}, t), \end{aligned} \tag{3}$$

and the iterative application of Eq. (3) for T times yield the noised source latent $\mathbf{x}_T^{\text{fwd}}$. The reconstruction of the source image is then obtained by applying the reverse DDIM process (Eq. (1)) from $\mathbf{x}_T^{\text{rev}} = \mathbf{x}_T^{\text{fwd}}$, producing $\mathbf{x}_0^{\text{recon}}$. Under the assumption of negligible discretization error [46], this forward–reconstruction procedure is theoretically lossless, guaranteeing $\mathbf{x}_0 = \mathbf{x}_0^{\text{recon}}$.

Null-text inversion. In practice, discretization errors can accumulate during the forward and reverse processes, leading to imperfect reconstructions. To account for such deviations, Null-text inversion [32] refines the reverse process (Eq. (1)) by introducing a timestep-dependent parameter ϕ_t, which serves as a learnable unconditional embedding. The parameter ϕ_t is optimized to align the reverse trajectory with the forward trajectory by minimizing the discrepancy between their corresponding latents:

$$\min_{\phi_t} \|\mathbf{x}_{t-1}^{\text{fwd}} - f^{\text{rev}}(\tilde{\mathbf{x}}_t^{\text{rev}}, t, \phi_t)\|_2^2, \tag{4}$$

where $\tilde{\mathbf{x}}_t^{\text{rev}}$ denotes the latent obtained from the optimized reverse process. This strategy enables an accurate reconstruction of the source image.

4. ScaleEdit

4.1. Problem formulation

We consider three conditioning images: (1) the high-resolution source $I_{\text{src}}^{\text{high}} \in \mathbb{R}^{H_h \times W_h \times 3}$, (2) its downsampled version $I_{\text{src}}^{\text{low}} \in \mathbb{R}^{H_l \times W_l \times 3}$, (3) and a low-resolution reference $I_{\text{ref}}^{\text{low}} \in \mathbb{R}^{H_l \times W_l \times 3}$ that represents the desired target image. The image $I_{\text{src}}^{\text{low}}$ is obtained by downsampling $I_{\text{src}}^{\text{high}}$, and the reference $I_{\text{ref}}^{\text{low}}$ is produced by applying a standard low-resolution image editing method (*e.g.* Nano Banana [8]) to $I_{\text{src}}^{\text{low}}$. The objective is to generate a high-resolution edited output $I_{\text{ref}}^{\text{high}}$ that reflects the overall semantics of $I_{\text{ref}}^{\text{low}}$ while maintaining the fine-grained details present in $I_{\text{src}}^{\text{high}}$.

4.2. Overview

The proposed method generates high-resolution output $I_{\text{ref}}^{\text{high}}$ in a patch-wise manner, for precise control over the fine-grained details present in $I_{\text{src}}^{\text{high}}$. We first divide all conditioning images into $N \times M$ patches (Sec. 4.3). For each patch, we transfer the fine-grained details residing in $I_{\text{src}}^{\text{high}}$ by manipulating the intermediate features of a pretrained generative model (Sec. 4.4), producing $N \times M$ enhanced target patches. Then, they are synchronized and fused into a coherent high-resolution target image (Sec. 4.5), ensuring smoothness across patch boundaries. Figure 2 provides a summary of our method.

4.3. Patch-wise generation trajectory extraction

Given three conditioning images, we first resize the low-resolution source $I_{\text{src}}^{\text{low}}$ and reference image $I_{\text{ref}}^{\text{low}}$ to match the resolution of the high-resolution source $I_{\text{src}}^{\text{high}}$. We then divide each image into non-overlapping $N \times M$ patches of size $\mathbb{R}^{H_d \times W_d \times 3}$, where $H_d \times W_d$ corresponds to the input resolution of the pretrained generative model [25, 41]. Thus, $N = H_h/H_d$ and $M = W_h/W_d$, resulting in a total of $N \times M$ patches. We denote the i^{th} patch of $I_{\text{src}}^{\text{high}}$ as $I_{\text{src}}^{\text{high}}[i]$, and we define $I_{\text{src}}^{\text{low}}[i]$ and $I_{\text{ref}}^{\text{low}}[i]$ likewise.

Each patch is then encoded using the pretrained VAE [24] encoder. Here, we define the resulting latent representations of the high-resolution source, low-resolution source, and low-resolution reference patches as $\mathbf{x}_0^{\text{high}}[i]$, $\mathbf{x}_0^{\text{low}}[i]$, and $\mathbf{y}_0^{\text{low}}[i]$, respectively, where $1 \le i \le NM$. We subsequently apply the forward process, optionally enhanced with Null-text inversion [32], to obtain their generation trajectories, which are given by

$$\{\mathbf{x}_t^{\text{high}}[i]\}_{t=0}^T, \{\mathbf{x}_t^{\text{low}}[i]\}_{t=0}^T, \{\mathbf{y}_t^{\text{low}}[i]\}_{t=0}^T. \tag{5}$$

4.4. Detail enhancement module

To inject fine-grained details from a low-resolution image to a target high-resolution image, we introduce a transfer function that aligns the generation trajectories of the low-resolution into high-resolution domains. Specifically, starting from $\tilde{\mathbf{x}}_T[i] = \mathbf{x}_T^{\text{low}}[i]$, we aim to guide its generation trajectory $\{\tilde{\mathbf{x}}_t[i]\}_{t=0}^T$ to follow that of the high-resolution source latent $\{\mathbf{x}_t^{\text{high}}[i]\}_{t=0}^T$.

To accomplish this, we define a timestep-dependent transfer function $\phi(i, t)$ that adjusts the intermediate feature representation $\mathbf{h}_t[i]$ of the pretrained generative model [25, 41]. Here, $\mathbf{h}_t[i]$ denotes the output of the ResNet block [15] along the upward path of the U-Net architecture [42], or the output of the final linear layer following the single-stream block in the transformer-based architecture [25]. This is illustrated as the yellow sub-block on the right side of Figure 2. We then introduce a feature modification term $\Delta\mathbf{h}_t[i] = \phi(i, t)$, which is applied to manipulate the intermediate feature of

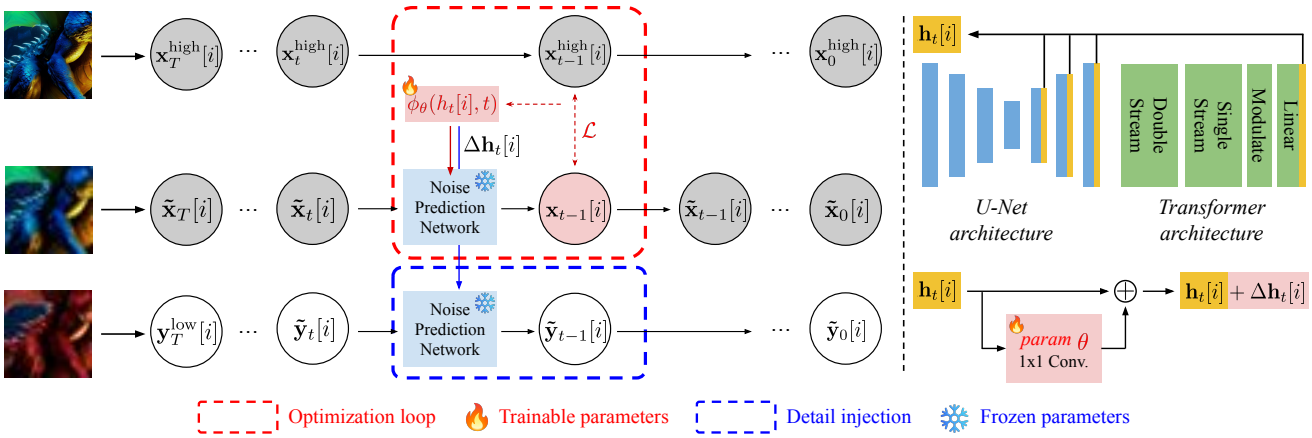


Figure 2. Overview of the proposed method. (Left) We first optimize the transfer function $\phi_\theta(\mathbf{h}_i[t], t)$ to capture fine-grained details encoded in the high-resolution source trajectory $\{\mathbf{x}_t^{\text{high}}[i]\}_{t=0}^T$. We then apply the optimized transfer function during the reverse process of $\{\tilde{\mathbf{y}}_t[i]\}_{t=0}^T$, yielding a detail-enhanced latent $\tilde{\mathbf{y}}_0[i] = \mathbf{y}_0^{\text{high}}[i]$. (Right) We illustrate how the transfer function modulates the intermediate feature within the pretrained model.

the pretrained model at each timestep t. The transfer function is optimized to align the low-resolution trajectory with the high-resolution trajectory by minimizing the following training objective:

$$\mathcal{L} := ||\mathbf{x}_{t-1}^{\text{high}}[i] - f^{\text{rev}}(\tilde{\mathbf{x}}_t[i], t\,; \Delta\mathbf{h}_t[i])||_2^2, \tag{6}$$

where $f^{\text{rev}}(\cdot, \cdot\,; \Delta\mathbf{h}_t[i])$ denotes the reverse process with the manipulated intermediate feature of $\mathbf{h}_t[i] + \Delta\mathbf{h}_t[i]$. For DDIM [45], it is defined as follows:

$$\begin{aligned} f^{\text{rev}}(\mathbf{x}_t^{\text{rev}}, t\,; \Delta\mathbf{h}_t[i]) := \sqrt{\alpha_{t-1}}\hat{\mathbf{x}}_0(\mathbf{x}_t^{\text{rev}}, t\,; \Delta\mathbf{h}_t[i]) \\ + \sqrt{1-\alpha_{t-1}}\epsilon_\theta(\mathbf{x}_t^{\text{rev}}, t\,; \Delta\mathbf{h}_t[i]). \end{aligned} \tag{7}$$

Note that $f^{\text{rev}}(\mathbf{x}_t^{\text{rev}}, t)$ in Eq. (1) is equal to $f^{\text{rev}}(\mathbf{x}_t^{\text{rev}}, t\,; \mathbf{0})$.

A straightforward parameterization of the timestep-dependent transfer function is to use a learnable constant vector $\phi(i, t) = \mathbf{c}_t[i]$, resulting in the adjusted intermediate feature $\mathbf{h}_t[i] + \mathbf{c}_t[i]$. However, we observe that this parameterization fails to produce reliable transformations, especially when the source and reference differ in semantics, *e.g.*, cat-to-dog. We hypothesize that this limitation arises because a single global constant vector is insufficient to model varying degrees of change throughout the edited image, and thus cannot adapt to diverse spatial structures.

To overcome this issue, we instead formulate the transfer function as an operation adaptive to $\mathbf{h}_t[i]$, rather than as a global constant vector. That is, we design the feature modification term as

$$\Delta\mathbf{h}_t[i] := \phi_\theta(\mathbf{h}_t[i], t), \tag{8}$$

where θ is the trainable parameter that defines the operation for each timestep t. Empirically, we define the transfer function using a 1×1 convolution applied to $\mathbf{h}_t[i]$, *i.e.* $\text{conv}_{1\times1}(\mathbf{h}_t[i])$. This lightweight operation allows feature mixing across channels while preserving spatial layout, which turns out to be effective for transferring fine-grained details in a spatially adaptive manner. A detailed discussion of the design of transfer function is explained in Appendix.

After optimizing the transfer function, we incorporate it during the reverse process to generate the target image, starting from $\tilde{\mathbf{y}}_T[i] = \mathbf{y}_T^{\text{low}}[i]$ and resulting in a trajectory of $\{\tilde{\mathbf{y}}_t[i]\}_{t=0}^T$, where $\tilde{\mathbf{y}}_{t-1}[i]$ is sampled via

$$\tilde{\mathbf{y}}_{t-1}[i] = f^{\text{rev}}(\tilde{\mathbf{y}}_t[i], t\,; \Delta\mathbf{h}_t[i]). \tag{9}$$

Then, we finally take $\mathbf{y}_0^{\text{high}}[i] = \tilde{\mathbf{y}}_0[i]$. This sampling strategy is effective since the objective in Eq. (6) encourages the transfer function to act as a feature-level detail injection operator, guiding the low-resolution source trajectory toward the high-resolution source trajectory while preserving the fine-grained details.

Optimizing and injecting $\Delta\mathbf{h}_t[i]$ over all timesteps facilitates fine-detail transfer, but may gradually distort the content of the source image and yield higher computation. In contrast, limiting optimization to early timesteps preserves the source image's content but may under-transfer fine details. To handle this trade-off, we introduce a hyperparameter τ that denotes the initial timestep at which our transfer function optimization is applied; that is, $\Delta\mathbf{h}_t[i] = \mathbf{0}$, $^\forall t > \tau$. Thus, it serves as a flexible controller that balances the level of detail transfer relative to content preservation, further allowing one to choose the trade-off depending on the editing strength and desired fidelity.

4.5. Synchronization between patches

While the transfer function $\Delta\mathbf{h}_t[i]$ injects fine-grained details into each patch, applying it independently to each patch may lead to inconsistent textures and artifacts at patch boundaries. To ensure global coherence across patches, we introduce a synchronization mechanism that combines blended Tweedie updates (Sec. 4.5.1) with a resampling strategy (Sec. 4.5.2).

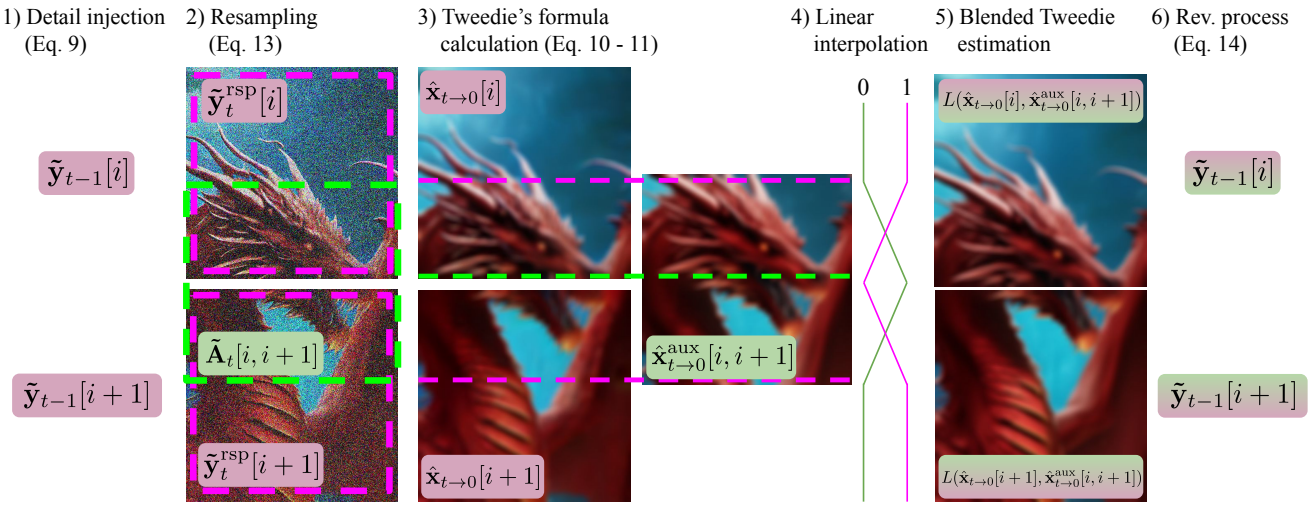


Figure 3. Overview of the synchronization strategy. Starting from the detail enhancement process (Eq. (9)), we perform resampling (Eq. (13)) and compute the blended Tweedie estimate. This estimate is then used to synchronize adjacent patches during the reverse process (Eq. (14)).

For clarity, we describe the case where adjacent patches are vertically aligned within the diffusion process, namely $\tilde{\mathbf{y}}_t[i]$ and $\tilde{\mathbf{y}}_t[i+1]$. The same procedure applies equivalently when they are placed along the horizontal directions.

4.5.1. Blended-Tweedie-based updates

To synchronize adjacent patches, we introduce an auxiliary latent $\tilde{\mathbf{A}}_t[i, i+1]$, constructed by spatially blending the bottom half of $\tilde{\mathbf{y}}_t[i]$ and the upper half of $\tilde{\mathbf{y}}_t[i+1]$ at timestep t. Since each patch is denoised independently, their latent trajectories may diverge, often producing visible discontinuities along their boundaries. However, $\tilde{\mathbf{A}}_t[i, i+1]$ contains the boundary region, and therefore its Tweedie estimate may naturally captures a smoother transition between the two patches. By blending the Tweedie estimate of an auxiliary latent with those of the original patches, the boundary between neighboring patches becomes more coherent, reducing artifacts and enhancing spatial continuity.

For spatial blending, we first apply Eq. (2) to obtain the original patches' Tweedie estimates:

$$\hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1] = \hat{\mathbf{x}}_0(\tilde{\mathbf{A}}_t[i, i+1], t) \tag{10}$$

$$\hat{\mathbf{x}}_{t\to0}[i] = \hat{\mathbf{x}}_0(\tilde{\mathbf{y}}_t[i], t). \tag{11}$$

Then we define $L(\hat{\mathbf{x}}_{t\to0}[i], \hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1])$, a blended Tweedie estimate that is used to swap the $\hat{\mathbf{x}}_0$ term in Eq. (1) during the reverse process of $\tilde{\mathbf{y}}_t[i]$. Let W_p and H_p denote the width and height of a single patch, respectively. We linearly interpolate the bottom half of $\tilde{\mathbf{y}}_t[i]$ and the corresponding region of the auxiliary latent $\tilde{\mathbf{A}}_t[i, i+1]$, *i.e.* the top half of $\tilde{\mathbf{A}}_t[i, i+1]$, whose overlap has spatial size $H_p/2 \times W_p$. To achieve this, we define a vertical interpolation weight as:

$$\mathbf{M}(v, t) = \frac{2v}{H_p} \cdot \left(1 - \frac{t}{\tau}\right) \quad 0 \le v \le H_p/2, \tag{12}$$

which increases linearly from the boundary toward the center of the overlap, so that the contribution of the auxiliary latent grows smoothly across the overlapping region. We then substitute the bottom half of the Tweedie estimate for $\tilde{\mathbf{y}}_t[i]$ with the linearly interpolated one, while keeping its top half unchanged. This yields the blended Tweedie estimate with respect to $\hat{\mathbf{x}}_{t\to0}[i]$ and $\hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1]$.

For the adjacent patch $\tilde{\mathbf{y}}_t[i+1]$, we apply the vertically mirrored blend operation. Specifically, we compute $L(\hat{\mathbf{x}}_{t\to0}[i+1], \hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1])$ with the weight of $1 - \mathbf{M}$ (vertically reversed ramp). This blending ensures spatial smoothness between adjacent patches. Detailed formulas indicating these procedures are described in the Appendix.

4.5.2. Resampling strategy

A challenge in the above formulation is that $\Delta\mathbf{h}_t$ remains undefined for $\tilde{\mathbf{A}}_t[i, i+1]$, since adjacent patches $\tilde{\mathbf{y}}_t[i]$ and $\tilde{\mathbf{y}}_t[i+1]$ contain independently injected detail terms, $\Delta\mathbf{h}_t[i]$ and $\Delta\mathbf{h}_t[i+1]$. In principle, $\Delta\mathbf{h}_t$ for $\tilde{\mathbf{A}}_t[i, i+1]$ can be computed explicitly; however, this entails a costly additional optimization. Instead, we propose an alternative approach which we refer to as the *resampling strategy*, where synchronization is decoupled from detail injection. The key idea is to reconstruct latents that preserve the fine-grained details already injected by the transfer function, but without dependency on any explicit $\Delta\mathbf{h}_t$ term. This allows synchronization to proceed without defining a separate $\Delta\mathbf{h}_t$ for $\tilde{\mathbf{A}}_t[i, i+1]$.

Given the detail-injected latents $\tilde{\mathbf{y}}_{t-1}[i]$ and $\tilde{\mathbf{y}}_{t-1}[i+1]$ obtained via Eq. (9), we apply a forward process *without* re-injecting the transfer function to get the resampled latents $\tilde{\mathbf{y}}_t^{\text{rsp}}[i]$ and $\tilde{\mathbf{y}}_t^{\text{rsp}}[i+1]$:

$$\begin{aligned} \tilde{\mathbf{y}}_t^{\text{rsp}}[i] &= f^{\text{fwd}}(\tilde{\mathbf{y}}_{t-1}[i], t-1), \\ \tilde{\mathbf{y}}_t^{\text{rsp}}[i+1] &= f^{\text{fwd}}(\tilde{\mathbf{y}}_{t-1}[i+1], t-1). \end{aligned} \tag{13}$$

Table 1. Quantitative evaluation under 1K- and 2K-editing scenarios using the pretrained Stable Diffusion [41]. We compare the proposed method with diffusion-based super-resolution methods [7, 10, 12, 48]. Our method shows superior performance compared to the baselines.

Method	1K-editing					2K-editing				
	HaarPSI ↑	M-MSE ↓	M-SSIM ↑	M-PSNR ↑	LPIPS ↓	HaarPSI ↑	M-MSE ↓	M-SSIM ↑	M-PSNR ↑	LPIPS ↓
DiT-SR [7]	0.335	0.058	0.695	21.528	0.477	0.316	0.057	0.754	21.380	0.507
DiT4SR [12]	0.324	0.060	0.625	20.740	0.509	0.305	0.058	0.684	20.701	0.534
PiSA-SR [48]	0.328	0.058	0.668	21.273	0.465	0.312	0.056	0.755	21.320	**0.472**
TSD-SR [10]	0.329	0.061	0.649	20.766	0.489	0.312	0.059	0.715	20.796	0.514
ScaleEdit (Ours)	**0.342**	**0.054**	**0.739**	**22.132**	**0.460**	**0.331**	**0.053**	**0.806**	**21.955**	0.496

This resampling process produces detail-preserving latents $\tilde{\mathbf{y}}_t^{\text{rsp}}$ that eliminates the need to define $\Delta\mathbf{h}_t$ for the auxiliary latent $\tilde{\mathbf{A}}_t[i, i+1]$. Using these resampled latents, we compute the blended Tweedie estimate using the synchronization procedure described in Sec. 4.5.1. Finally, the actual reverse process at timestep t is performed via Eq. (14):

$$\tilde{\mathbf{y}}_{t-1}[i] = \sqrt{\alpha_{t-1}} L(\hat{\mathbf{x}}_{t\to 0}[i], \hat{\mathbf{x}}^{\text{aux}}_{t\to 0}[i, i+1]) + \sqrt{1-\alpha_{t-1}}\epsilon_\theta(\tilde{\mathbf{y}}_t^{\text{rsp}}[i], t). \tag{14}$$

The proposed mechanism harmonizes patch-wise diffusion trajectories, yielding globally consistent synthesis with reduced boundary artifacts and preserved fine-grained details.

5. Experiments

5.1. Implementation details

We implement our method based on the official codebase of Null-text inversion [32] using PyTorch [35]. We primarily evaluate our method on Stable Diffusion [41] v2.1-base and further demonstrate its applicability to FLUX.1-dev [25], which adopts a transformer-based architecture [52]. For both cases, we use a total timestep of $T = 50$ and set hyperparameter $\tau = 15$. During the forward and reverse process, we use an empty prompt (“”), since a global text prompt describing $I_{\text{src}}^{\text{high}}$ and $I_{\text{ref}}^{\text{low}}$ does not fully cover the semantics of each patch. We adapt Null-text inversion for Stable Diffusion to achieve accurate reconstruction. After obtaining $N \times M$ latents $\{\mathbf{y}_0^{\text{high}}[i]\}_{i=1}^{NM}$, we spatially merge them and decode the resulting latent using the pretrained VAE [24] decoder to get $I_{\text{ref}}^{\text{high}}$. Additional implementation details are provided in Appendix.

5.2. Data acquisition

Editing scenarios. To comprehensively evaluate high-resolution image editing, we consider two settings that reflect practical usage scenarios. We first examine the *1K-editing* scenario, where the reference and target resolutions are set to 512^2 and 1K, respectively. While existing editing methods generally support editing up to this scale, we include this setting not as a challenging benchmark, but to highlight that our method transfers the fine-grained details faithfully. Next, we evaluate *2K-editing*, which corresponds to a more common real-world setup for editing beyond the operating resolution of pretrained generative models.

Asset generation. We construct high-quality assets tailored for our task. We first generate a total of 100 source images at 4K resolution using FreeScale [38], then downsample them to 2K, 1K, and 512^2 resolution to obtain multi-resolution source images. Source prompts and editing instructions are produced with ChatGPT, where the instructions span four editing configurations: (1) two types of object replacement, (2) style transfer, and (3) background modification. We then apply Nano Banana [8] to the 512^2 and 1K versions of each source image to obtain the corresponding edited images. Here, the 512^2-sized images serve as low-resolution references for 1K-editing, whereas the 1K resolution images serve as references for 2K-editing.

5.3. Quantitative evaluation

Baselines. A straightforward strategy for our task is to perform editing at a lower resolution and then apply super-resolution (SR) methods. However, as discussed in the introduction, this approach fails to recover micro-scale textures because the fine-grained details of the source image are not conditioned during SR process. To verify that our claim is valid, we construct baselines by combining a state-of-the-art low-resolution editing method, Nano Banana [8], with various super-resolution approaches. Specifically, we compare our method against (1) DiT-SR [7], (2) DiT4SR [12], (3) PiSA-SR [48], and (4) TSD-SR [10]. We use a total of 400 (conditioning images, text instruction) pairs for evaluation of each editing scenario.

Metrics. We evaluate the results using five widely used metrics: MSE, SSIM, PSNR, LPIPS [61] and HaarPSI [40]. To measure MSE, SSIM, and PSNR, we compute their masked variants by applying region-specific masks to the source and target images, which are denoted as M-MSE, M-SSIM, and M-PSNR, respectively. In particular, we use background masks for object change tasks and foreground masks for background modification tasks. These masks are generated using LANG-SAM[1], which produces instruction-aware foreground and background segmentations. We emphasize

[1] https://github.com/luca-medeiros/lang-segment-anything

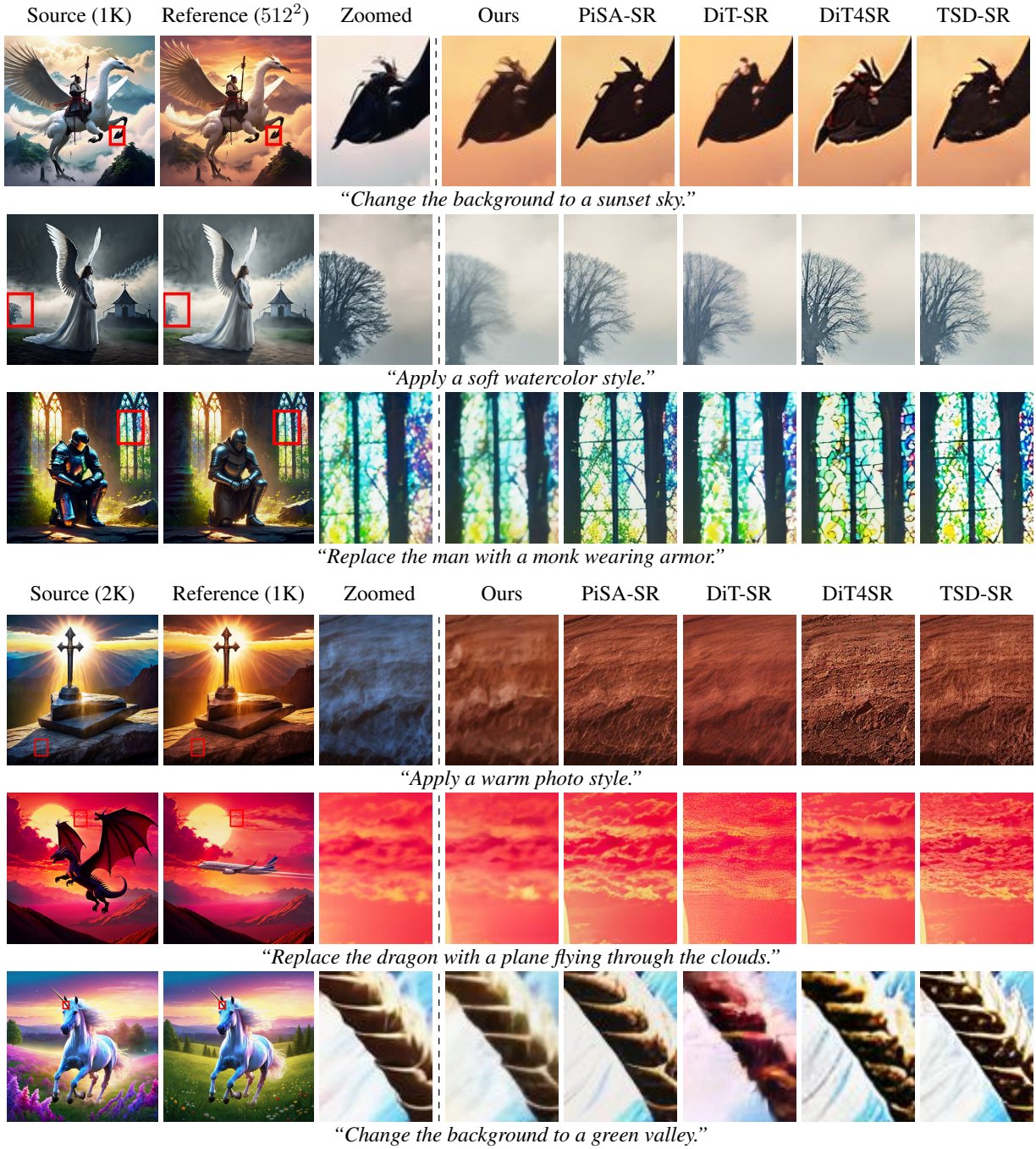
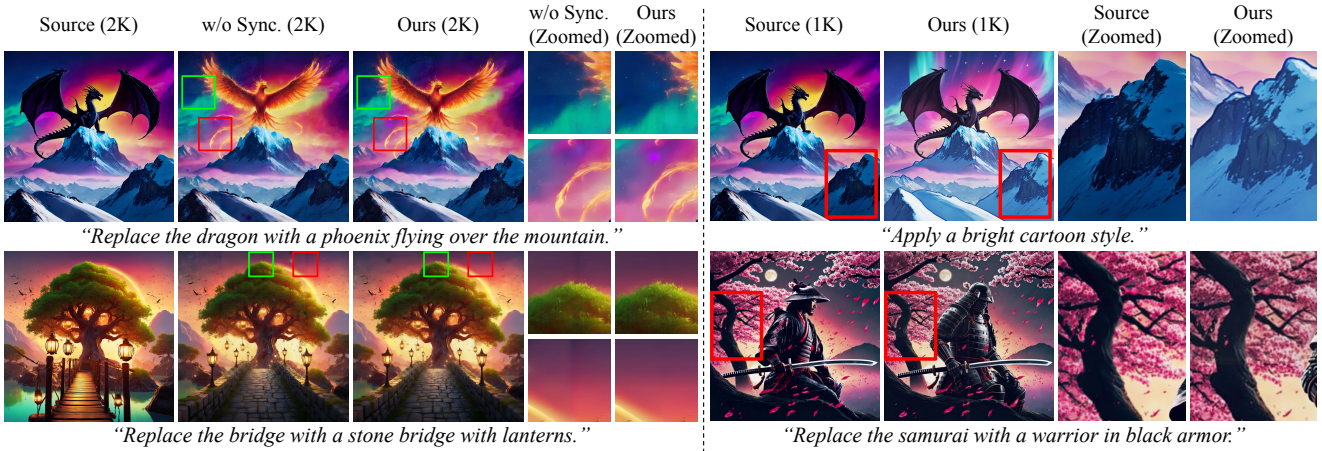


Figure 4. **[Best visualized when magnified.]** Qualitative comparison of the ScaleEdit with diffusion-based super-resolution baselines [7, 10, 12, 48]. First three rows show the results of 1K-editing, while the last three rows visualize 2K-editing. Here, we use the pretrained Stable Diffusion [41] for ScaleEdit.

that the masked metrics allow us to access how well each method preserves the regions of the source image that are intended to remain unchanged during editing. LPIPS quantifies the perceptual similarity between source and target images, which is widely used for image-to-image translation evaluation. HaarPSI measures similarity in a wavelet-aware manner, enabling evaluation of how effectively fine-grained details are transferred from the source to the target image.

Comparisons. We quantitatively compare our method with baselines under 1K-editing and 2K-editing scenarios. Table 1 summarizes the evaluation results. For each column of the table, we **bold** and underline the best and second-best result. As shown, our method consistently outperforms all baselines, highlighting the effectiveness of the detail en-



(a) Ablation study results.

(b) 1K-editing results using the pretrained FLUX.1-dev model

Figure 5. **[Best visualized when magnified.]** We visualize the effect of synchronization strategy in part (a), while we show the results of our method combined with the pretrained FLUX.1-dev [25] model in part (b).

hancement and synchronization mechanism. In contrast, SR-based pipelines struggle to recover fine-grained details from the source image. Moreover, the superior performance of ScaleEdit on masked-variant metrics indicates that it synthesizes the target image in a more source-aware manner, mitigating the inherent limitations of SR-based methods that exhibits source-unconditional generation.

5.4. Qualitative results

We present qualitative comparisons between our method and baselines using the pretrained Stable Diffusion [41] in Figure 4. Here, the target image is expected to satisfy two criteria simultaneously: (1) accurately inherit the fine-grained details from the zoomed source image and (2) correctly reflect the semantics described in the text instruction. Our method produces superior results, faithfully transferring fine-grained details of the source image to the target image while leveraging the desired semantics from the text instructions. For example, in the 2nd row, the baselines struggle to adopt the "watercolor" style, whereas our method effectively renders the watercolor appearance while preserving the detail of the source image. Similarly, in the 4th and 6th rows, our approach reliably carries over the texture-like details from the source, while the baselines exhibit noticeable color shifts or distorted textures.

We visualize additional qualitative results in Figure 1 and 5 (b). Importantly, our method is not limited to Stable Diffusion architecture; it also generalizes to other backbones such as FLUX [25]. We also demonstrate the *scalability* of the proposed method by visualizing 8K-editing results in Figure 6. While existing image editing methods are either unable to handle 8K-editing scenario or require additional training, our method effectively performs editing regardless of the resolution without any additional tuning, enabled by the patch-wise detail transfer mechanism.

Figure 6. Qualitative results of ScaleEdit in 8K-editing scenario.

5.5. Ablation study

We show the effectiveness of the proposed patch-wise synchronization strategy through an ablation study. As illustrated in Figure 5 (a), the absence of synchronization leads to noticeable boundary artifacts, producing visible seams and inconsistencies along patch borders. In contrast, synchronization enforces coherent denoising trajectories across neighboring patches, resulting in artifact-free transitions while preserving the local fidelity of each patch.

6. Conclusion

In this work, we introduce the task of high-resolution image editing for the first time and propose a general framework to address it. To fully inherit the strong prior of the pretrained image generation model, we divide the high-resolution source image into patches that match the model's native input resolution. We then transfer fine-grained details from the source to the target image through a learnable transfer function that operates on intermediate features of the pretrained network. To alleviate boundary artifacts between patches, we further propose a synchronization approach that eliminates the need for overlapping-view sampling used in prior synchronization methods, meaningfully reducing computational overhead. Experiments across multiple resolution editing scenarios demonstrate that our method is capable of high-resolution image editing.

Acknowledgements

This work was supported in part by NSF IIS2404180, and Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration) and (No. RS-2022-00187238, Development of Large Korean Language Model Technology for Efficient Pre-training). This work was also supported by Samsung Electronics Co., Ltd (IO250418-12669-01).

References

[1] Yuang Ai, Xiaoqiang Zhou, Huaibo Huang, Xiaotian Han, Zhengyu Chen, Quanzeng You, and Hongxia Yang. Dreamclear: High-capacity real-world image restoration with privacy-safe dataset curation. *NeurIPS*, 2024. 2

[2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022. 2

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 1

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2

[5] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *ICCV*, 2023. 2, 3

[6] Junyu Chen, Wenkun He, Yuchao Gu, Yuyang Zhao, Jincheng Yu, Junsong Chen, Dongyun Zou, Yujun Lin, Zhekai Zhang, Muyang Li, et al. Dc-videogen: Efficient video generation with deep compression video autoencoder. *arXiv:2509.25182*, 2025. 1

[7] Kun Cheng, Lei Yu, Zhijun Tu, Xiao He, Liyu Chen, Yong Guo, Mingrui Zhu, Nannan Wang, Xinbo Gao, and Jie Hu. Effective diffusion transformer architecture for image super-resolution. In *AAAI*, 2025. 1, 2, 6, 7, 14

[8] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv:2507.06261*, 2025. 1, 2, 3, 6

[9] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *ICLR*, 2023. 2

[10] Linwei Dong, Qingnan Fan, Yihong Guo, Zhonghao Wang, Qi Zhang, Jinwei Chen, Yawei Luo, and Changqing Zou. Tsd-sr: One-step diffusion with target score distillation for real-world image super-resolution. In *CVPR*, 2025. 1, 2, 6, 7, 14

[11] Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion: Democratising high-resolution image generation with no $$$. In *CVPR*, 2024. 1

[12] Zheng-Peng Duan, Jiawei Zhang, Xin Jin, Ziheng Zhang, Zheng Xiong, Dongqing Zou, Jimmy Ren, Chun-Le Guo, and Chongyi Li. Dit4sr: Taming diffusion transformer for real-world image super-resolution. In *ICCV*, 2025. 1, 2, 6, 7, 14

[13] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *NeurIPS*, 2023. 2

[14] 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. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[16] 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*, 2024. 1

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *ICLR*, 2023. 2

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022. 2

[20] Teng Hu, Jiangning Zhang, Zihan Su, and Ran Yi. Ultragen: High-resolution video generation with hierarchical attention. *AAAI*, 2026. 1

[21] 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*, 2024. 1

[22] Junoh Kang, Donghun Ryou, and Bohyung Han. Icm-sr: Image-conditioned manifold regularization for image super-resoultion. *arXiv:2511.22048*, 2025. 2

[23] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, 2022. 2

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013. 3, 6, 12

[25] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 2, 3, 6, 8, 13, 14, 15

[26] Hyunsoo Lee, Minsoo Kang, and Bohyung Han. Conditional score guidance for text-driven image-to-image translation. *NeurIPS*, 2023. 2

[27] Hyunsoo Lee, Minsoo Kang, and Bohyung Han. Diffusion-based conditional image editing through optimized inference with guidance. In *WACV*, 2025.

[28] Junsung Lee, Minsoo Kang, and Bohyung Han. Diffusion-based image-to-image translation by noise correction via prompt interpolation. In *ECCV*, 2024. 2

[29] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ICLR*, 2023. 13

[30] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *arXiv:2504.17761*, 2025. 1, 2

[31] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ICLR*, 2023. 13

[32] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023. 3, 6, 14

[33] Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, et al. Mineru2. 5: A decoupled vision-language model for efficient high-resolution document parsing. *arXiv:2509.22186*, 2025. 1

[34] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *SIGGRAPH*, 2023. 2, 3

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 6

[36] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 2

[37] 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. *ICLR*, 2024. 2

[38] 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. In *ICCV*, 2025. 1, 6, 13

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 13

[40] Rafael Reisenhofer, Sebastian Bosse, Gitta Kutyniok, and Thomas Wiegand. A haar wavelet-based perceptual similarity index for image quality assessment. *Signal Processing: Image Communication*, 2018. 6

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 6, 7, 8, 13, 14, 15

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 2, 3, 13

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022. 2

[44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021. 2, 3, 4

[46] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021. 2, 3

[47] Charles M Stein. Estimation of the Mean of a Multivariate Normal Distribution. *The annals of Statistics*, 1981. 3

[48] Lingchen Sun, Rongyuan Wu, Zhiyuan Ma, Shuaizheng Liu, Qiaosi Yi, and Lei Zhang. Pixel-level and semantic-level adjustable super-resolution: A dual-lora approach. In *CVPR*, 2025. 1, 2, 6, 7, 14

[49] Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025. 1

[50] Tencent Hunyuan3D Team. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details, 2025. 1

[51] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023. 2, 3

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 2017. 2, 6, 14

[53] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *IJCV*, 2024. 2

[54] Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C Kot, and Bihan Wen. Sinsr: diffusion-based image super-resolution in a single step. In *CVPR*, 2024.

[55] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *NeurIPS*, 2024.

[56] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *CVPR*, 2024. 2

[57] Wei Wu, Xi Guo, Weixuan Tang, Tingxuan Huang, Chiyu Wang, and Chenjing Ding. Drivescape: High-resolution driving video generation by multi-view feature fusion. In *CVPR*, 2025. 1

[58] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ICLR*, 2024. 2

[59] Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild. In *CVPR*, 2024. 2

[60] Jinjin Zhang, Qiuyu Huang, Junjie Liu, Xiefan Guo, and Di Huang. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. In *CVPR*, 2025. 1

[61] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6, 13

[62] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *NeurIPS*, 2025. 2

[63] Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. Kv-edit: Training-free image editing for precise background preservation. *ICCV*, 2025. 1, 2

Low-Resolution Editing is All You Need for High-Resolution Editing

Supplementary Material

A. Pseudo-code of ScaleEdit

We provide an abstract overview of the proposed method by outlining its core procedure in Algorithm 1.

Algorithm 1 ScaleEdit

Inputs: Conditioning images $I_{\text{src}}^{\text{high}}$, $I_{\text{src}}^{\text{low}}$, $I_{\text{ref}}^{\text{low}}$, hyperparameter τ
// 1. Forward process
for $i \leftarrow 1, \cdots, NM$ **do**
  Get $\{\mathbf{x}_t^{\text{high}}[i]\}_{t=0}^{T}, \{\mathbf{x}_t^{\text{low}}[i]\}_{t=0}^{T}, \{\mathbf{y}_t^{\text{low}}[i]\}_{t=0}^{T}$
  $\tilde{\mathbf{x}}_T[i] \leftarrow \mathbf{x}_T^{\text{low}}[i]$, $\tilde{\mathbf{y}}_T[i] \leftarrow \mathbf{y}_T^{\text{low}}[i]$
end for
// 2. Transfer function optimization
for $t \leftarrow T, \cdots, 1, \quad i \leftarrow 1, \cdots, NM$ **do**
  if $t \leq \tau$ **then**
    Optimize ϕ_θ using Eq. (6)
    $\Delta\mathbf{h}_t[i] \leftarrow \phi_\theta(\mathbf{h}_t[i], t)$
  else
    $\Delta\mathbf{h}_t[i] \leftarrow \mathbf{0}$
  end if
  Get $\tilde{\mathbf{x}}_{t-1}[i]$ using Eq. (7)
end for
// 3. Detail injection with synchronization
for $t \leftarrow T, \cdots, 1$ **do**
  Get $\tilde{\mathbf{y}}_{t-1}[i]$ using Eq. (9) for $1 \leq i \leq NM$
  if $t \leq \tau$ **then**
    Get $\tilde{\mathbf{y}}_t^{\text{rsp}}[i]$ using Eq. (13) for $1 \leq i \leq NM$
    Get $\tilde{\mathbf{y}}_{t-1}[i]$ using Eq. (14) for $1 \leq i \leq NM$
  end if
end for
$I_{\text{ref}}^{\text{high}} \leftarrow \text{Decode}(\text{Composite}(\{\tilde{\mathbf{y}}_0[i]\}_{i=1}^{NM}))$
Output: Target image $I_{\text{ref}}^{\text{high}}$

B. Detailed explanation on synchronization

To synchronize adjacent patches, we introduce an auxiliary latent $\tilde{\mathbf{A}}_t[i, i+1]$, constructed by spatially blending the bottom half of $\tilde{\mathbf{y}}_t^{\text{rsp}}[i]$ and the upper half of $\tilde{\mathbf{y}}_t^{\text{rsp}}[i+1]$ at timestep t. Since each patch is denoised independently, their latent trajectories may diverge, often producing visible discontinuities along their boundary. However, $\tilde{\mathbf{A}}_t[i, i+1]$ contains the boundary region, and therefore its Tweedie estimate may naturally captures a smoother transition between the two patches. By blending the Tweedie estimate of an auxiliary patch with those of the original patches, the boundary between neighboring patches becomes more coherent, reducing artifacts and enhancing spatial continuity.

For the spatial blending, we first apply Eq. (2) to obtain the original patches' Tweedie estimates:

$$\hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1] = \hat{\mathbf{x}}_0(\tilde{\mathbf{A}}_t[i, i+1], t) \tag{15}$$

$$\hat{\mathbf{x}}_{t\to0}[i] = \hat{\mathbf{x}}_0(\tilde{\mathbf{y}}_t^{\text{rsp}}[i], t). \tag{16}$$

Then we define $L(\hat{\mathbf{x}}_{t\to0}[i], \hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1])$, a blended Tweedie estimate that is used to swap the $\hat{\mathbf{x}}_0$ term in Eq. (1) during the reverse process of $\tilde{\mathbf{y}}_t^{\text{rsp}}[i]$. Let $\mathcal{T}_1$ be the vertical translation operator defined in $[0, H_p] \times [0, W_p]$ as follows:

$$\mathcal{T}_1 f(u, v) = \begin{cases} 0, & 0 \leq v < H_p/2, \\ f(u, v - H_p/2), & H_p/2 \leq v \leq H_p, \end{cases} \tag{17}$$

where W_p and H_p denote the patch width and height, respectively. Thus, $\mathcal{T}_1\hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1]$ repositions $\hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1]$ so that its top edge aligns with the midpoint of $\hat{\mathbf{x}}_{t\to0}[i]$. Then we define a spatial weight mask $\mathbf{M}_1(u, v, t)$ for smooth transition across the overlap:

$$\mathbf{M}_1(u, v, t) = \begin{cases} 0, & 0 \leq v < H_p/2, \\ \frac{v - H_p/2}{H_p/2} \cdot \left(1 - \frac{t}{\tau}\right), & H_p/2 \leq v \leq H_p. \end{cases} \tag{18}$$

The blended Tweedie estimate is then computed as

$$\begin{aligned} &L(\hat{\mathbf{x}}_{t\to0}[i], \hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1]) \\ &= (1 - \mathbf{M}_1) \odot \hat{\mathbf{x}}_{t\to0}[i] + \mathbf{M}_1 \odot \mathcal{T}_1\hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1]. \end{aligned} \tag{19}$$

For the rest of the paragraph, we detail the case of $\tilde{\mathbf{y}}_t^{\text{rsp}}[i+1]$. Let $\mathcal{T}_2$ and $\mathbf{M}_2(u, v, t)$ defined in the vertically-flipped manner compared to $\mathcal{T}_1$ and $\mathbf{M}_1(u, v, t)$:

$$\mathcal{T}_2 f(u, v) = \begin{cases} f(u, v + H_p/2), & 0 \leq v \leq H_p/2 \\ 0, & H_p/2 < v < H_p, \end{cases} \tag{20}$$

$$\mathbf{M}_2(u, v, t) = \begin{cases} \frac{H_p/2 - v}{H_p/2} \cdot \left(1 - \frac{t}{\tau}\right), & 0 \leq v \leq H_p/2 \\ 0, & H_p/2 < v < H_p. \end{cases} \tag{21}$$

Subsequently, the blended Tweedie estimate is calculated as:

$$\begin{aligned} &L(\hat{\mathbf{x}}_{t\to0}[i+1], \hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1]) \\ &= (1 - \mathbf{M}_2) \odot \hat{\mathbf{x}}_{t\to0}[i+1] + \mathbf{M}_2 \odot \mathcal{T}_2\hat{\mathbf{x}}_{t\to0}^{\text{aux}}[i, i+1]. \end{aligned} \tag{22}$$

C. Implementation details

Latent decoding. After denoising $N \times M$ latent patches, we first merge them into a single latent tensor. The merged latent is then passed through the pretrained VAE decoder [24] to obtain the final high-resolution image $I_{\text{ref}}^{\text{high}}$.

Adaptation to Stable Diffusion. For Stable Diffusion [41] v2.1-base, the upsampling path of the U-Net [42] consists of four upsampling blocks, each composed of three sub-blocks. We insert a 1×1 convolution layer into the last sub-block of each upsampling block, resulting in four additional 1×1 convolution layers in total.

Adaptation to FLUX. For FLUX.1-dev [25], we attach the detail enhancement module to the linear layer located immediately after the last single-stream block, which is followed by a Layer Normalization operation. Since FLUX is based on flow matching [29, 31], we apply ScaleEdit with several additional modifications. The forward and reverse processes follow deterministic ODE dynamics, defined as:

$$\mathbf{x}_{t+1}^{\text{fwd}} = \mathbf{x}_t^{\text{fwd}} + (\sigma_{t+1} - \sigma_t) \cdot \mathbf{v}_\theta(\mathbf{x}_t^{\text{fwd}}, t), \tag{23}$$

$$\mathbf{x}_{t-1}^{\text{rev}} = \mathbf{x}_t^{\text{rev}} + (\sigma_{t-1} - \sigma_t) \cdot \mathbf{v}_\theta(\mathbf{x}_t^{\text{rev}}, t), \tag{24}$$

where $\mathbf{v}_\theta(\cdot, \cdot)$ denotes the pretrained vector field prediction network. The clean data sample is estimated as:

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t) = \mathbf{x}_t - \sigma_t \mathbf{v}_\theta(\mathbf{x}_t, t), \tag{25}$$

which corresponds to the Tweedie estimate in diffusion models. In addition, the reverse process using the blended latent differs from that of diffusion models. We calculate the blended latent $L(\hat{\mathbf{x}}_{t\to0}[i], \hat{\mathbf{x}}^{\text{aux}}_{t\to0}[i, i+1])$ using the same blending operation described in Sec. 4.5.1 of the main paper, which corresponds to the blended Tweedie estimate in diffusion models. We then compute the modified vector field using the blended latent as follows:

$$\mathbf{v}'(t)[i] = \frac{\tilde{\mathbf{y}}_t^{\text{rsp}}[i] - L(\hat{\mathbf{x}}_{t\to0}[i], \hat{\mathbf{x}}^{\text{aux}}_{t\to0}[i, i+1])}{\sigma_t}, \tag{26}$$

Finally, we perform the reverse process using the modified vector field:

$$\tilde{\mathbf{y}}_{t-1}[i] = \tilde{\mathbf{y}}_t^{\text{rsp}}[i] + (\sigma_{t-1} - \sigma_t) \cdot \mathbf{v}'(t)[i]. \tag{27}$$

In practice, to achieve accurate reconstruction, we cache the output of each attention layer in the forward process and reuse it in the reverse process.

D. Additional ablation studies

Impact of hyperparameter τ. As described in Algorithm 1, ScaleEdit utilizes a hyperparameter τ. To investigate its effect, we conduct an ablation study by varying τ in $15, 25, 35$ and evaluate the results using the same metrics as the main experiment on a set of 60 (image, instruction) pairs. As shown in Table 2, the default setting $\tau = 15$ achieves the best performance across all configurations.

Table 2. Ablation study on τ values.

τ	HaarPSI ↑	M-MSE ↓	M-SSIM ↑	M-PSNR ↑	LPIPS ↓
15	0.335	0.042	0.573	17.430	0.472
25	0.326	0.043	0.561	17.097	0.471
35	0.308	0.044	0.537	16.308	0.496

Ablation on the design choice of transfer function. We first analyze the effect of layers by varying the index of the sub-block used within each block of the U-Net [42] that are used for the detail transfer module. Specifically, for each stage (down/middle/up), we select N-th sub-block from all blocks in that stage. Variant of ϕ_θ is also evaluated by replacing the convolution with constant mapping. Evaluation is conducted using a total of 60 (image, instruction) pairs. Tab. 3 shows our design (Up #3) performs best; while maintaining the overall perceptual similarity (measured by LPIPS [61]), it also preserves the intended editing effects (quantified by CLIP-Sim [39] between target image and target prompt). Alternatives still outperform most baselines but fall short. Since the last (3rd) sub-blocks in up stage operate at high-resolution, they are most effective at transferring fine-grained details needed for high-resolution generation.

Table 3. Ablation study on design of detail transfer module.

Method	Down #1	Down #2	Middle	Up #1	Up #2	**Up #3**	Constant
LPIPS ↓	0.1718	0.1746	0.1913	**0.1648**	0.1663	**0.1648**	0.1781
CLIP-Sim. ↑	0.2676	0.2677	0.2674	0.2681	0.2681	**0.2684**	0.2683

Ablation on synchronization strategy. In Figure 9, we illustrate the effect of the proposed synchronization method. As shown, naive patch sampling without synchronization brings noticeable edge artifacts along patch boundaries, whereas our method effectively mitigates these artifacts.

E. Receptive field of generative models

In this work, we leverage a low-resolution image generation model to perform high-resolution image editing in a patch-wise manner. Although FreeScale [38] modifies internal components of a low-resolution diffusion model (*e.g.* via dilated convolutions) to generate high-resolution outputs, such architectural changes do not guarantee a strong high-resolution image prior, as the model is not trained on high-resolution image datasets.

In contrast, our patch-wise approach is well-suited for extracting detailed information. Each patch matches the resolution of the low-resolution model's receptive field (*e.g.* 512×512 for the pretrained Stable Diffusion [41]), enabling faithful detail reconstruction without modifying the underlying generator. The main challenge in the modified framework lies in generating details that the model was not originally trained to produce. For these reasons, we adopt a patch-wise strategy grounded in low-resolution image priors instead of modifying diffusion architectures to synthesize high-resolution content directly.

F. Additional results

To demonstrate the performance of ScaleEdit in real scenarios, we conduct additional 1K-editing experiments using a total of 285 (image, text instruction) pairs. We sample high-resolution real images from the publicly accessible Internet source[2]. Then we follow the procedure described in Sec. 5.2 of the main paper for experiments. Figure 7 and Table 4 shows the result of real image editing. As shown, our method demonstrates strong performance.

We also show the additional results of ScaleEdit on synthetic images in Figure 8 and 10. Figure 8 shows that our method is even applicable to transformer-based [52] FLUX model [25], demonstrating the robustness and generalizability of our method across backbone architectures. Then, we show the additional 1K- and 2K-editing results obtained with the pretrained Stable Diffusion [41] in Figure 10. We emphasize that the proposed method effectively transfers the fine-grained details of the source image into the target image.

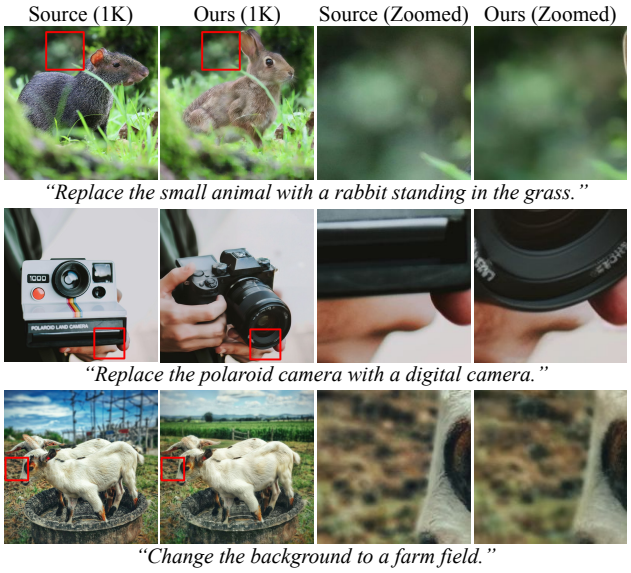


Figure 7. **[Best visualized when magnified.]** Qualitative results of real image editing in 1K resolution.

Table 4. Quantitative evaluation under 1K-editing scenario with real images using the pretrained Stable Diffusion [41].

Method	HaarPSI ↑	M-MSE ↓	M-SSIM ↑	M-PSNR ↑	LPIPS ↓
DiT-SR [7]	0.347	0.068	0.692	22.407	0.148
DiT4SR [12]	0.345	0.068	0.706	22.377	0.141
PiSA-SR [48]	0.346	0.067	0.719	22.658	0.137
TSD-SR [10]	0.347	0.071	0.693	22.127	0.139
ScaleEdit (Ours)	**0.375**	**0.065**	**0.781**	**23.270**	**0.134**

[2]https://www.pexels.com/

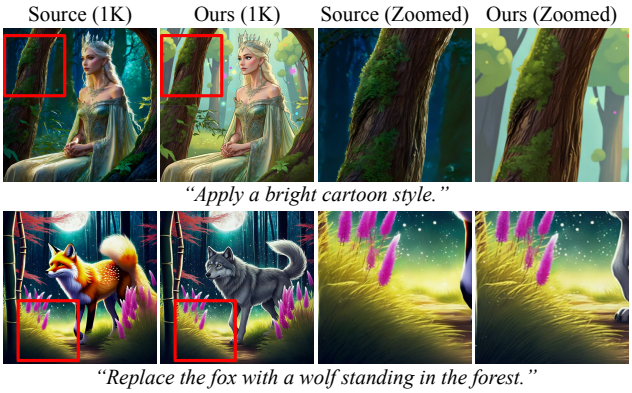


Figure 8. **[Best visualized when magnified.]** Qualitative results of our method combined with the pretrained FLUX.1-dev [25] model.

G. Discussion on computational cost

We evaluate the computational efficiency of our method against baselines in terms of runtime and GPU memory usage using a single NVIDIA A6000 GPU on 1K-editing scenario. While baselines in half-precision typically require 0.74–88.06 seconds and 8.40–36.00 GB of VRAM, our method currently operates in full-precision to ensure numerical stability during optimization, requiring 20.29 GB. Although our method currently has a higher overhead, we anticipate that further refinement of the implementation will enable comparable efficiency in half-precision without compromising performance.

We also note that our framework is compatible with Null-text Inversion (NTI) [32] for accurate reconstruction. The runtime without and with NTI are 234.77 and 635.51 seconds, respectively; we clarify that latter case's overhead is mainly due to NTI's iterative optimization rather than our core components. Ultimately, these results represent a justifiable trade-off for the state-of-the-art performance our method achieves in Table 1 and Figure 4, consistently outperforming baselines in high-resolution image editing. Note that for the evaluation results reported in the main paper, all methods were executed in full-precision whenever possible, unless this led to out-of-memory errors.

H. Limitations

Our method may produce suboptimal results due to the limited performance of the pretrained generative models [25, 41]. Furthermore, since the proposed method relies on the low-resolution reference image $I_{\text{ref}}^{\text{low}}$ generated by existing low-resolution image-to-image translation methods, some artifacts introduced by these editing methods can propagate to the final high-resolution output $I_{\text{ref}}^{\text{high}}$, potentially leading to visible artifacts.

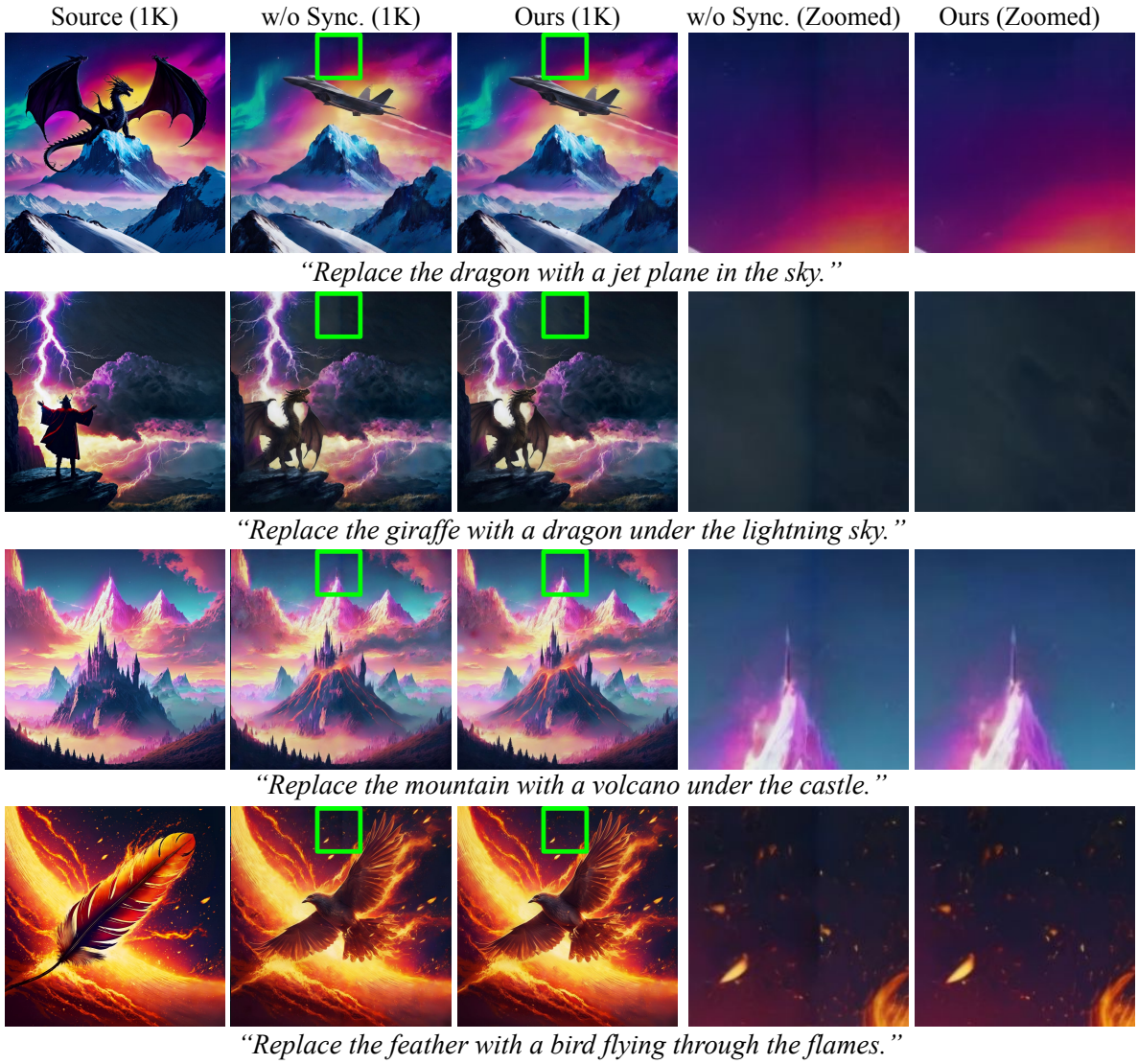


Figure 9. **[Best visualized when magnified.]** Effectiveness of the proposed synchronization strategy. While images sampled without synchronization incorporates a significant artifact on the patch boundaries, our method effectively alleviates the edge artifacts.

I. Societal impacts

The proposed method may generate some harmful images due to the imperfections of the underlying pretrained generative models [25, 41]. Rare cases of undesired or inappropriate results may arise, especially when the generative prior itself produces such outputs under challenging conditions.

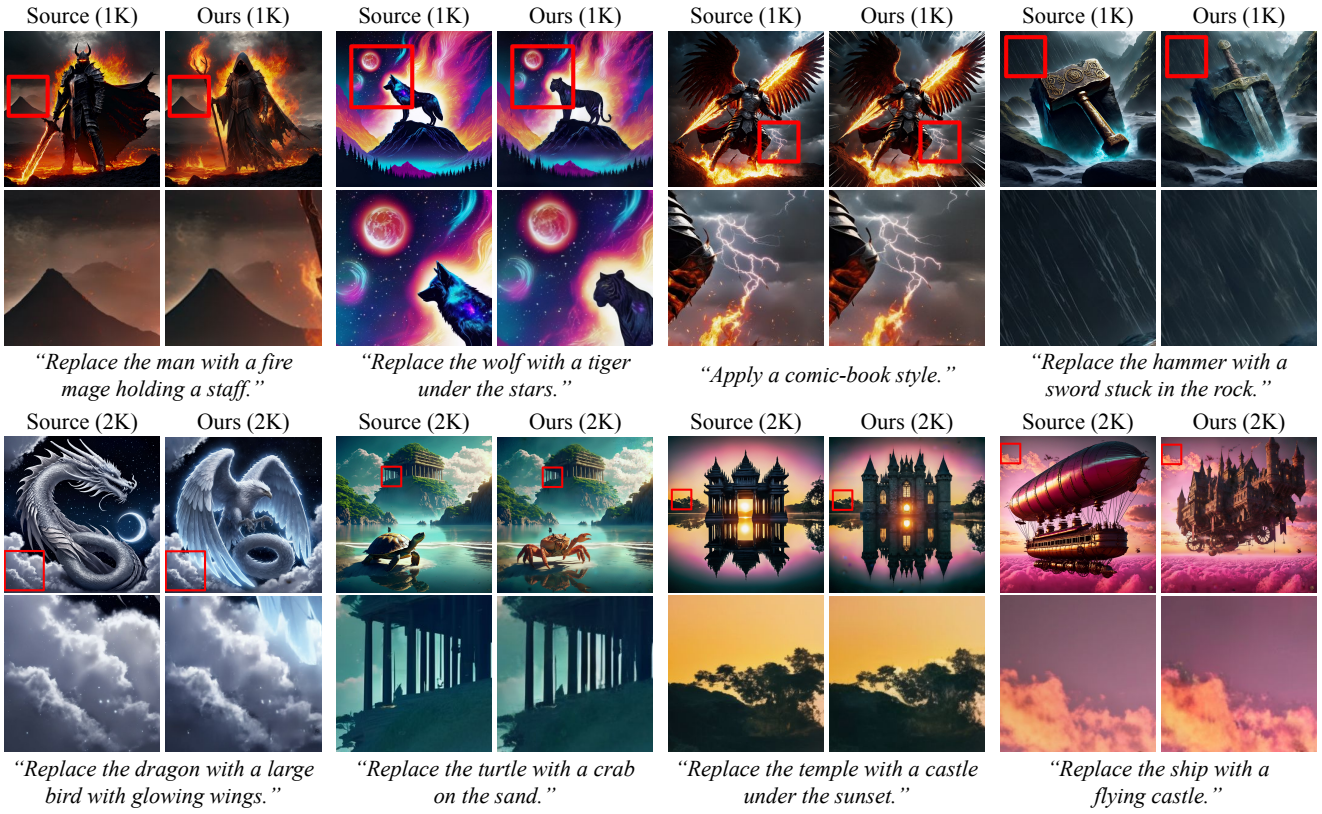


Figure 10. **[Best visualized when magnified.]** We visualize the additional results of 1K-editing and 2K-editing. By conditioning on low-resolution reference images, our method is successfully synthesizes high-resolution edited images.