

LUA: Latent Upscaling Adapter for Diffusion-Based Image Synthesis

Aleksandr Razin^{1*}, Danil Kazantsev^{1*}, and Ilya Makarov²

¹ Independent Researcher, Yerevan, Armenia

² iMak AI Lab, Ngerulmud, Palau

`razin.x.aleks@gmail.com, danilkazansevml@gmail.com`

Abstract. Generating high-resolution images with latent diffusion models is limited by the cost of high-resolution denoising, while post-hoc super-resolution introduces artifacts and additional latency by operating after decoding. We present the Latent Upscaling Adapter (LUA), a lightweight module that performs super-resolution directly on the generator’s latent code before the final VAE decoding step. LUA integrates as a drop-in component requiring no modifications to the base model or additional diffusion stages, enabling high-resolution synthesis through a single feed-forward pass in latent space. A shared Swin-style backbone with scale-specific pixel-shuffle heads supports $\times 2$ and $\times 4$ factors with nearly $3\times$ lower decoding and upscaling time. Because LUA operates directly on latents, it transfers across diffusion systems—including SDXL, SD3, and FLUX—through architecture-level reuse, adapting only the input convolution with brief fine-tuning rather than retraining from scratch. Extensive experiments demonstrate that LUA closely matches the fidelity of native high-resolution generation at 2K and 4K while offering a practical and efficient path to scalable image synthesis.

Keywords: Latent super-resolution · Diffusion models · High-resolution image synthesis · Latent upscaling adapter

1 Introduction

Diffusion models have transformed image synthesis, progressing from pixel-space formulations [17, 36] to *Latent Diffusion Models* (LDMs), which shift computation into compact latent representations [33]. This latent formulation underpins contemporary systems for image generation, editing, and translation [2, 12]. Despite these advances, current models are effectively constrained by the spatial resolutions seen during training (typically 512^2 or 1024^2). Naïvely sampling beyond these scales often yields repetition, geometric distortions, and texture breakdown [15, 44]. While retraining or high-resolution fine-tuning can reduce such artifacts [23], these remedies demand substantial compute and data.

A pragmatic alternative is to generate at the native resolution and subsequently upsample. Two main paradigms implement this strategy. *Pixel-space*

* These authors contributed equally.

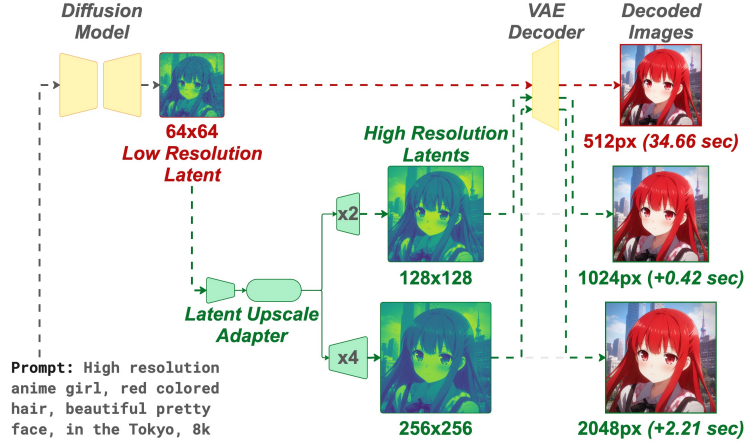


Fig. 1: LUA integrates into diffusion pipelines without retraining the generator or adding diffusion stages. A FLUX [2] generator produces a 64×64 latent (red path decodes directly); LUA (green path) upsamples to 128×128 ($\times 2$) or 256×256 ($\times 4$) and decodes once, adding only $+0.42$ s/ $+2.21$ s on an NVIDIA L40S.

super-resolution (SR) applies an external SR model to the decoded image. This approach is simple but reconstructs fine structure purely from pixels, which encourages oversmoothing, semantic drift, and a computational cost that grows quadratically with output size [6, 41]. *Latent-space upsampling* instead enlarges the latent representation prior to decoding, reducing inference cost and better preserving semantics. Recent reference-based methods, such as DemoFusion-style pipelines [11] and LSRNA [18], first generate a low-resolution reference, upsample it, and then run a second diffusion stage guided by the upscaled latent. Although effective, these pipelines require multi-stage inference, auxiliary noise or guidance branches, and tight coupling to specific VAEs, which increases latency and limits generality across model families.

In practice, the *upsampling step* is the principal bottleneck. Bicubic resizing or naïve latent interpolation departs from the manifold of valid latents, producing unnatural textures after decoding (Fig. 2, column 2). Conversely, pixel-space SR applied after decoding can improve fidelity but incurs higher latency and may introduce noise (Fig. 2, column 3). The core challenge is to increase latent resolution while preserving manifold geometry and high-frequency latent details that decode into realistic images—*without* invoking an additional diffusion process.

We address this challenge with the **Latent Upscaling Adapter (LUA)**, a lightweight module inserted between the generator and VAE decoder (Fig. 1). Given a latent $z \in \mathbb{R}^{h \times w \times C}$, LUA predicts an upscaled latent $\hat{z} \in \mathbb{R}^{\alpha h \times \alpha w \times C}$ for $\alpha \in \{2, 4\}$, which is decoded once to produce the final image. Because VAE decoders typically expand spatial dimensions by stride $s=8$, a $\times 2$ latent upscaling yields a $16 \times$ increase in pixel count with no extra denoising. LUA employs a shared SwinIR-style backbone [25] with lightweight, scale-specific pixel-shuffle

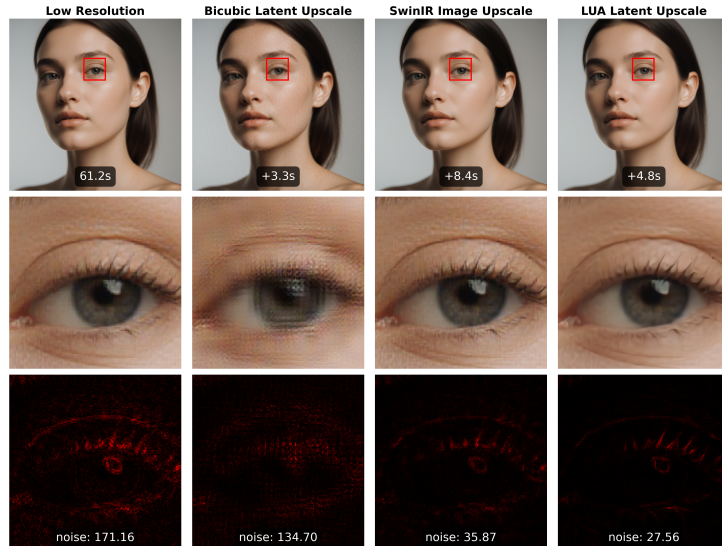


Fig. 2: FLUX $1024^2 \rightarrow 2048^2$. Columns: (1) base decode, (2) bicubic latent, (3) SwinIR pixel-space, (4) LUA latent-space. Middle: $8\times$ crops; bottom: Laplacian-variance (darker = less noise). LUA preserves detail with the lowest noise and overhead.

heads and is trained directly in the latent space with complementary latent- and pixel-domain objectives to preserve high-frequency consistency and enable photorealistic decoding (Fig. 2, column 4).

Our study focuses on three questions:

1. Can a simple latent adapter deliver higher-quality images at lower cost than native high-resolution synthesis or pixel-space SR?
2. Can a single adapter handle multiple upscaling factors within one unified framework?
3. Can an adapter trained for one model’s VAE transfer to other generators with minimal fine-tuning?

We evaluate LUA across backbones and resolutions using FID/KID/CLIP and local-detail metrics, with ablations on architecture, objectives, and scale handling. A single adapter generalizes beyond its OpenImages training data, both to a second benchmark (COCO-100) and across diverse content domains. Preserving latent microstructure via a single upscaling stage narrows the gap to native high-res synthesis while reducing complexity and latency relative to multi-stage diffusion and pixel-space SR.

Our contributions are threefold:

- We train *only* a latent upscaling adapter with a backbone-agnostic three-stage curriculum and show that latent-space super-resolution attains quality comparable to modern high-resolution diffusion pipelines at 2K and 4K while being more efficient and less noisy than pixel-space SR.

- We design a single model that supports multiple scale factors ($\times 2$, $\times 4$) via a shared backbone with jointly trained, scale-specific heads—avoiding retraining from scratch.
- We demonstrate cross-VAE generalization: the same backbone operates across SD3 [12], SDXL [29], and FLUX [2] by changing only the first layer to match input channels, with brief fine-tuning rather than retraining from scratch.

2 Related Work

This section reviews three lines of work relevant to high-resolution image synthesis: (i) efficient diffusion-based generation at large scales, (ii) super-resolution in pixel and latent spaces, and (iii) multi-scale (discrete vs. continuous) super-resolution. For each, we outline representative methods and the limitations that motivate our approach.

Efficient high-resolution generation with diffusion models. Diffusion models in compressed latent spaces [33] enable controllable synthesis, yet sampling beyond the training scale (512^2 – 1024^2) often yields repetition, distortions, or texture loss [15, 44]. Direct high-resolution training has been demonstrated in large systems such as SDXL [29], SD3 [12], and related frameworks [23], but it demands massive datasets and compute. Inference-time strategies avoid full retraining: tiling/blending in MultiDiffusion [1] preserves locality but risks seams; receptive-field expansions via adaptive/dilated or sparse convolutions [13] and step-reduction by distillation/consistency methods [28, 37] improve speed but can reduce fidelity at extreme resolutions; progressive pipelines such as HiDiffusion and ScaleCrafter [15, 44] iteratively upsample and refine; reference-based approaches like DemoFusion [11] synthesize a low-resolution latent, upsample it, and re-diffuse; FreeScale [30] fuses multi-scale features at inference time without training but remains tied to the diffusion loop; and FMBoost [35] enhances latent diffusion using flow matching to map low-resolution latents to higher-resolution latent spaces before decoding, improving efficiency but still introducing an additional generative transformation stage. These families rely on additional diffusion or generative passes with method-specific schedules and step counts, increasing latency and coupling execution to particular resolution settings.

Super-resolution in image and latent spaces. Pixel-space SR advanced from CNN regressors (SRCNN, EDSR) [10, 26] to adversarial/perceptual (SRGAN, ESRGAN) [21, 40] and transformer models (SwinIR, HAT) [7, 25]; diffusion-based SR (SR3, SRDiff, SeeSR, StableSR, DiffBIR, SUPIR) [24, 27, 34, 39, 41, 43] further boosts perceptual fidelity. However, all denoise at the target resolution, incurring quadratic compute/memory with image size and risking semantic drift in fine textures. Latent-space SR reduces cost by upsampling before decoding, but naïve latent interpolation (e.g., bicubic/linear) departs from the generative manifold, and learned mappings such as LSRNA or the latent guidance used in DemoFusion [11, 18] still depend on a subsequent diffusion stage, yielding multi-stage, latency-heavy pipelines, whereas LUA performs a single feed-forward pass

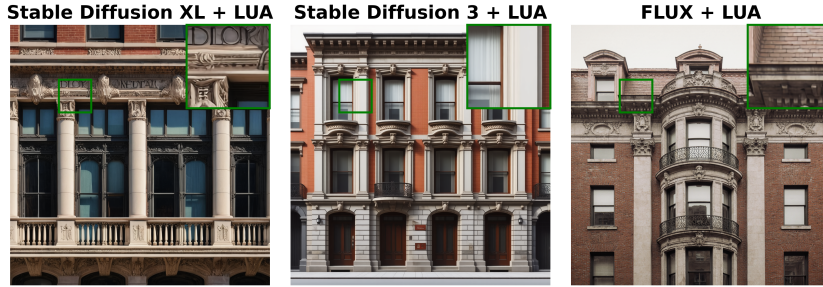


Fig. 3: Cross-model $2\times$ latent upscaling with a single adapter. The same LUA upscales $128\times128\rightarrow256\times256$ latents from SDXL [29], SD3 [12], and FLUX [2], decoded once to 2048^2 . SDXL ($C=4$) requires only a first-convolution change. Green boxes: $\times8$ zooms.

without re-diffusion, roughly $6\times$ faster at 2048^2 (Sec. 4). Latent SR has also been studied for video, where SimpleGVR [42] performs latent-cascaded super-resolution with a multi-step diffusion stage that LUA avoids.

Discrete vs. continuous multi-scale SR. Discrete-factor SR (e.g., $\times2$, $\times4$) commonly trains separate networks or employs a shared backbone with scale-specific heads (MDSR, SwinIR) [25, 26], which is effective but resource intensive to train and store for multiple scales. Arbitrary-scale methods (LIIF, LTE, CiaoSR) [4, 8, 22] predict continuous coordinates from learned features, yet often underperform direct upscaling on fine textures where high-frequency structures dominate.

In contrast, we target a single-pass *latent* upscaler that avoids extra diffusion stages, side-steps the quadratic cost of pixel-space SR, and replaces naïve latent resizing with a dedicated training curriculum. The design supports multiple scales via a shared backbone with lightweight, scale-specific heads, delivering high-resolution fidelity with substantially lower latency.

3 Proposed Method

We target high-resolution synthesis without an extra diffusion stage or retraining the generator/VAE by inserting a single-pass *Latent Upscaling Adapter* (LUA) between the generator and the frozen decoder: LUA enlarges the latent, then one decode produces the final image with near-base-decode overhead. In Sec. 3.1 we formalize the upscaling operator U_α , establish its computational efficiency relative to image-space SR and multi-stage diffusion, and show cross-model generalization (FLUX, SD3, SDXL) via changing only the first convolution layer and minimal fine-tuning. We then describe the multi-scale *architecture*—a shared backbone with scale-specific heads for $\times2$ and $\times4$ (Sec. 3.2)—and the *multi-stage training* strategy that preserves latent microstructure and stabilizes decoded appearance (Sec. 3.3).

3.1 Latent Upscaling

Formulation. Given text condition c and noise ϵ , a pretrained generator G produces a latent $z \in \mathbb{R}^{h \times w \times C}$:

$$z = G(c, \epsilon). \quad (1)$$

A frozen VAE decoder D with spatial stride s (typically $s=8$) maps z to an RGB image $x \in \mathbb{R}^{(sh) \times (sw) \times 3}$:

$$x = D(z). \quad (2)$$

We introduce a deterministic latent upscaler U_α with scale factor $\alpha \in \{2, 4\}$ that maps $z \in \mathbb{R}^{h \times w \times C}$ to $\hat{z} \in \mathbb{R}^{\alpha h \times \alpha w \times C}$:

$$\hat{z} = U_\alpha(z). \quad (3)$$

A single decode yields the high-resolution image:

$$\hat{x} = D(\hat{z}). \quad (4)$$

Here, h, w are the latent spatial dimensions, C is the latent channel width, s is the decoder stride, c is the conditioning (e.g., text embedding), and ϵ is the generator noise. All generative stochasticity resides in G ; U_α is a feed-forward operator trained to remain on the latent manifold and to preserve the fine-scale statistics required for photorealistic decoding.

Computational efficiency. Pixel-space SR operates on $(sh) \times (sw)$ positions; LUA operates on $h \times w$ positions and still decodes *once*. The cost ratio scales as

$$\frac{O((sh)(sw))}{O(hw)} \approx s^2, \quad (5)$$

so for typical $s=8$ our upscaler touches about $1/64$ as many spatial elements as image-space SR. In addition, unlike progressive or reference-based pipelines, LUA does not add a second diffusion pass or any full-resolution refinement stage. Empirically, this yields lower wall-clock overhead and memory traffic while achieving comparable high-resolution fidelity (Sec. 4).

Cross-model generalization. LUA acts on latents, not backbone internals, and therefore does not require training from scratch for each VAE. The same model generalizes across FLUX, SD3, and SDXL by changing only the *first convolution layer* to match the input channel count and fine-tuning the adapter on a small set of latents from the target model. The backbone and scale heads remain unchanged. Architectural details are given in Sec. 3.2; the fine-tuning protocol and data sizes are in Sec. 4.

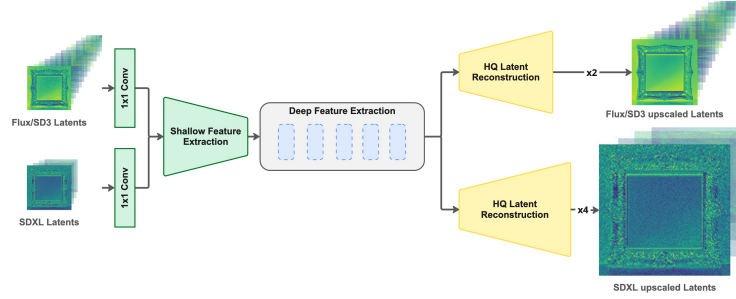


Fig. 4: LUA architecture. A shared SwinIR-style backbone [25] with a 1×1 input conv adapting the VAE channel width ($C=16$ for FLUX/SD3; $C=4$ for SDXL) and scale-specific pixel-shuffle heads for $\times 2/\times 4$ upscaling.

3.2 Architecture

We adopt a Swin Transformer restoration backbone in the spirit of SwinIR [25], which has proven effective for super-resolution in RGB space, and consistent with latent-domain adaptations such as LSRNA [18]. Windowed self-attention provides long-range context while preserving locality, matching the spatial-statistical structure of VAE latents. Given an input latent $z \in \mathbb{R}^{h \times w \times C}$, the backbone $\phi(\cdot)$ extracts features through an encoder-decoder with residual connections. Upscaling is realized with explicit SR heads—shallow convolutions followed by pixel-shuffle—that directly predict an enlarged latent; compared with implicit coordinate decoders (e.g., LIIF [8]), this better preserves high-frequency latent microstructure that decodes into sharper textures.

To support multiple scale factors without duplicating capacity, a single backbone ϕ is shared across scale-specific heads for $\times 2$ and $\times 4$ (denoted $U_{\times 2}$ and $U_{\times 4}$). Joint training with balanced sampling encourages the backbone to learn scale-agnostic representations while each head specializes to its factor’s aliasing and artifact profile. At inference, the backbone runs once and the head corresponding to the requested scale $\alpha \in \{2, 4\}$ is applied:

$$\hat{x} = D(U_{\times \alpha}(\phi(z))), \quad \alpha \in \{2, 4\}, \quad (6)$$

where D is the frozen VAE decoder and z is the input latent with channel width C . The overall module layout is depicted in Fig. 4.

To operate across VAEs with different latent channel widths (e.g., SD3/FLUX with $C=16$ and SDXL with $C=4$), we change only the first convolution layer to match the input channels; the shared backbone and the scale heads are reused unchanged. A brief fine-tuning on a small set of latents from the target model aligns statistics and enables cross-VAE transfer without training from scratch. The fine-tuning protocol and data sizes are detailed in Sec. 4, and cross-model results are shown in Fig. 3.

3.3 Multi-stage Training Strategy

Single-domain objectives are insufficient for latent SR: optimizing *only* in latent space preserves coarse structure but yields decoded images with residual grid/blur and spurious high-frequency noise, while optimizing *only* in pixel space is unstable—gradients backpropagated through the frozen VAE decoder interact with unnormalized latents and fail to converge. To address both issues, we adopt a progressive three-stage curriculum that first secures latent structural and spectral alignment, then couples latent fidelity to decoded appearance with high-frequency emphasis, and finally refines edges in pixel space without re-diffusion. Throughout this section, z and $\hat{z}$ denote input and upscaled latents, z_{HR} the reference HR latent (from the frozen VAE encoder), and $x=D(z)$, $\hat{x}=D(\hat{z})$, $x_{\text{HR}}=D(z_{\text{HR}})$ their decoded images via the frozen decoder D . Superscripts z and x indicate losses computed in latent and pixel domains, respectively.

Stage I – Latent-domain structural alignment. This stage learns a stable mapping $\hat{z} = U_\alpha(z)$ that matches high-resolution latent structure and spectra while avoiding over-smoothing. The objective is

$$\mathcal{L}_{\text{SI}} = \alpha_1 \mathcal{L}_{\text{L1}}^z + \beta_1 \mathcal{L}_{\text{FFT}}^z, \quad (7)$$

where $\alpha_1=1.0$ and $\beta_1=0.1$ balance reconstruction and spectral alignment in latent space, with

$$\mathcal{L}_{\text{L1}}^z = \|\hat{z} - z_{\text{HR}}\|_1, \quad (8)$$

$$\mathcal{L}_{\text{FFT}}^z = \|\mathcal{F}(\hat{z}) - \mathcal{F}(z_{\text{HR}})\|_1. \quad (9)$$

Here $\hat{z} \in \mathbb{R}^{\alpha h \times \alpha w \times C}$ and $\mathcal{F}(\cdot)$ is the channel-wise 2D FFT magnitude [14]. $\mathcal{L}_{\text{L1}}^z$ enforces element-wise correspondence; $\mathcal{L}_{\text{FFT}}^z$ aligns high-frequency latent statistics to preserve microstructure [19].

Stage II – Joint latent–pixel consistency. This stage links latent fidelity to decoded appearance by adding image-domain constraints while retaining the Stage I latent terms:

$$\mathcal{L}_{\text{SII}} = \alpha_2 \mathcal{L}_{\text{L1}}^z + \beta_2 \mathcal{L}_{\text{FFT}}^z + \gamma_2 \mathcal{L}_{\text{DS}}^x + \delta_2 \mathcal{L}_{\text{HF}}^x, \quad (10)$$

with $\alpha_2=1.0$, $\beta_2=0.1$, $\gamma_2=0.1$, $\delta_2=0.05$, and

$$\mathcal{L}_{\text{DS}}^x = \|\downarrow_d(\hat{x}) - \downarrow_d(x_{\text{HR}})\|_1, \quad (11)$$

$$\mathcal{L}_{\text{HF}}^x = \left\| (\hat{x} - \mathcal{G}_\sigma(\hat{x})) - (x_{\text{HR}} - \mathcal{G}_\sigma(x_{\text{HR}})) \right\|_1. \quad (12)$$

Here $\downarrow_d(\cdot)$ denotes bicubic downsampling [31] with $d=2$ for $\times 2$ and $d=4$ for $\times 4$, and $\mathcal{G}_\sigma(\cdot)$ is a Gaussian blur with $\sigma=1.0$ (applied channel-wise). $\mathcal{L}_{\text{DS}}^x$ enforces coarse appearance consistency at a common reduced scale; $\mathcal{L}_{\text{HF}}^x$ emphasizes edges and textures by matching high-frequency residuals.

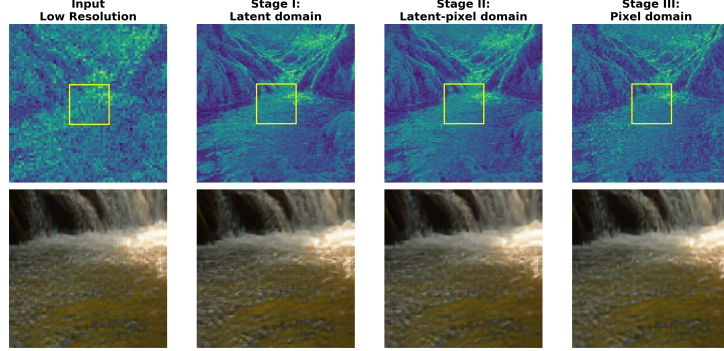


Fig. 5: Three-stage curriculum progression (FLUX). Top: latent feature maps (ch. 10); bottom: $8\times$ decoded crops. Columns: (1) LR input, (2–4) after Stages I–III. Quality improves, with Stage III sharpening edges while preserving high-frequency detail.

Stage III – Edge-aware image refinement. The final stage sharpens edges and suppresses residual ringing/grids in pixel space, without any additional denoising:

$$\mathcal{L}_{\text{SIII}} = \alpha_3 \mathcal{L}_{\text{L1}}^x + \beta_3 \mathcal{L}_{\text{FFT}}^x + \gamma_3 \mathcal{L}_{\text{EAGLE}}^x, \quad (13)$$

with $\alpha_3=10.0$, $\beta_3=1.0$, $\gamma_3=5\times 10^{-5}$, and

$$\mathcal{L}_{\text{L1}}^x = \|\hat{x} - x_{\text{HR}}\|_1, \quad (14)$$

$$\mathcal{L}_{\text{FFT}}^x = \|\mathcal{F}(\hat{x}) - \mathcal{F}(x_{\text{HR}})\|_1, \quad (15)$$

where $\mathcal{L}_{\text{EAGLE}}^x$ is an edge-aware gradient localization loss that enforces crisp boundaries and reduces staircase artifacts [38].

Weight selection. These weights were set via grid search, with ℓ_1 and FFT terms (latent and image domains) receiving the largest coefficients as primary reconstruction and texture-sensitive objectives and the downsampling and edge-aware terms acting as smaller auxiliary regularizers. The full grid-search procedure is detailed in the supplementary material.

Our multi-stage latent-pixel curriculum adapts LUA to the latent domain and the frozen decoder. Stage I aligns the upscaled latent with high-resolution structure and spectra (Fig. 5, col. 2). Stage II introduces joint latent-image supervision (downsampled and high-frequency terms) to normalize latents before decoding and suppress noise (col. 3). Stage III trains in pixel space only, selectively removing redundant noise and grid artifacts while preserving necessary high-frequency detail (col. 4). The curriculum enables coherent cross-domain training and yields single-decode synthesis without an extra diffusion stage; per-stage loss weights are provided in the supplementary material.

4 Experiments

We evaluate LUA on OpenImages with a unified train/val setup, comparing to representative high-resolution baselines and reporting fidelity (FID/KID and patch variants), text-image alignment (CLIP), and wall-clock latency. We also study cross-model and multi-scale generalization and ablate the curriculum and architecture. Further implementation details and extended results are provided in the supplementary material.

4.1 Experiment Settings

Training data. Following LSRNA [18], we use OpenImages [20]; photos with both sides ≥ 1440 px are tiled into non-overlapping 512×512 crops. HR/LR pairs are made by bicubic downsampling at the target scale ($\times 2$, $\times 4$), following the standard NTIRE super-resolution protocol [9]. Because LUA upscales clean, model-generated latents rather than captured images, the relevant degradation is resolution downsampling rather than sensor noise or compression, and VAE decoding further smooths residual noise. Crops are encoded with the FLUX VAE (stride $s=8$, $C=16$) to produce latent: for $\times 2$, $16 \times 32 \times 32 \rightarrow 16 \times 64 \times 64$; for $\times 4$, $16 \times 16 \times 16 \rightarrow 16 \times 64 \times 64$. The final set contains 3.8M pairs.

Multi-stage training. We train the three stages (Sec. 3.3) with Adam (lr 2×10^{-4} , weight decay 0), EMA 0.999, grad-clip 0.4, and MultiStepLR (decay $\gamma=0.5$ at 62.5k, 93.75k, and 112.5k steps). Each stage runs 125k steps. Hyperparameters follow SwinIR [25]. We keep the same initial lr across stages; Stage III uses a warm-up. Effective batch sizes: 2,048 (Stage I) and 32 (Stages II–III). Losses match Sec. 3.3. Total training takes ~ 270 GPU-hours on 8 NVIDIA H100s.

Scalability: multi-scale and cross-VAE. A single Swin backbone is shared by the $\times 2 / \times 4$ heads ($U_{\times 2}$, $U_{\times 4}$) and trained jointly with balanced sampling (Sec. 3, Fig. 4). For cross-VAE transfer, we build 500k LR/HR latent pairs for SDXL ($C=4$) and SD3 ($C=16$), replace only the first convolution to match channels, and perform a brief Stage III-style fine-tune, reusing the backbone and heads without retraining from scratch.

4.2 Evaluation Protocol

Validation data. We evaluate on 1,000 held-out high-resolution OpenImages photos (disjoint from training). Prompts are obtained via captioning and shared across methods; see the supplementary material for details.

Metrics. We report FID [16], KID [3], and CLIP [32]. Images are synthesized at the target resolution ($1024^2/2048^2/4096^2$). To better capture fine detail, we report patch metrics (pFID/pKID) on random 1024×1024 crops with an identical crop policy across methods, following LSRNA and Anyres-GAN [5]. Additional metrics and computation details are provided in the supplementary material.

Table 1: Quantitative comparison on OpenImages (Sec. 4.2). SDXL base 512^2 (1K), 1024^2 (2K/4K). Median runtime on H100, batch 1. Best in **bold**.

Resolution	Method	FID ↓	pFID ↓	KID ↓	pKID ↓	CLIP ↑	Time (s)
1024×1024	HiDiffusion	232.55	230.39	0.0211	0.0288	0.695	1.54
	DemoFusion	195.82	193.99	0.0153	0.0229	0.725	2.04
	LSRNA-DemoFusion	194.55	192.73	0.0151	0.0228	0.734	3.09
	SDXL (Direct)	194.53	192.71	0.0151	0.0225	0.731	1.61
	SDXL + SwinIR	210.40	204.23	0.0313	0.0411	0.694	2.47
	SDXL + LUA (ours)	209.80	191.75	0.0330	0.0426	0.738	1.42
2048×2048	HiDiffusion	200.72	114.30	0.0030	0.0090	0.738	4.97
	DemoFusion	184.79	177.67	0.0030	0.0100	0.750	28.99
	LSRNA-DemoFusion	181.24	98.09	0.0019	0.0066	0.762	20.77
	SDXL (Direct)	202.87	116.57	0.0030	0.0086	0.741	7.23
	SDXL + SwinIR	183.16	100.09	0.0020	0.0077	0.757	6.29
	SDXL + LUA (ours)	180.80	97.90	0.0018	0.0065	0.764	3.52
4096×4096	HiDiffusion	233.65	95.95	0.0158	0.0214	0.698	122.62
	DemoFusion	185.36	177.89	0.0043	0.0113	0.749	225.77
	LSRNA-DemoFusion	177.95	62.07	0.0023	0.0071	0.757	91.64
	SDXL (Direct)	280.42	101.89	0.0396	0.0175	0.663	148.71
	SDXL + SwinIR	183.15	65.71	0.0018	0.0103	0.756	7.29
	SDXL + LUA (ours)	176.90	61.80	0.0015	0.0152	0.759	6.87

Table 2: Cross-model, multi-scale LUA results ($\times 2/\times 4$, $1024^2 \rightarrow 2048^2/4096^2$). Setup matches Table 1. Best in **bold**.

Scale	Diffusion Model	FID ↓	pFID ↓	KID ↓	pKID ↓	CLIP ↑	Time (s)
$\times 2$	FLUX + LUA	180.99	100.40	0.0020	0.0079	0.773	29.829
	SDXL + LUA	183.15	101.18	0.0020	0.0087	0.753	3.52
	SD3 + LUA	184.58	103.94	0.0022	0.0083	0.768	20.292
$\times 4$	FLUX + LUA	181.06	62.30	0.0018	0.0085	0.772	31.908
	SDXL + LUA	182.42	71.92	0.0015	0.0152	0.754	6.87
	SD3 + LUA	183.34	67.25	0.0016	0.0095	0.769	21.843

Baselines. We compare ScaleCrafter [15], HiDiffusion [44] (progressive), DemoFusion [11], LSRNA-DemoFusion [18] (reference-based re-diffusion), SDXL (Direct), and SDXL+SwinIR [25] (pixel-space SR). All SDXL-based methods share the SDXL base checkpoint, a DDIM sampler, classifier-free guidance 7.5, and seed 42, differing only in the high-resolution strategy and using 30–50 sampling steps depending on the method. Full per-method configurations are in the supplementary material.

Our method. *SDXL+LUA*: generate at 1024^2 , upscale in latent space, decode once. We report SDXL-anchored comparisons in Table 1 and summarize cross-model/multi-scale results (FLUX/SD3/SDXL at $\times 2/\times 4$) in Table 2.

Runtime measurement. Per-image wall-clock time is the median over N runs after 20 warm-ups on a single GPU with AMP and batch size 1. Composite pipelines include generation, upscaling, and decode.

4.3 Quantitative Results

Table 1 summarizes fidelity–efficiency trade-offs across resolutions. At 1024^2 , SDXL+LUA attains the lowest latency (1.42s) but lags strongest single-stage



Fig. 6: Speed-quality trade-off and FLUX upscaling

baselines on fidelity (FID 209.80 vs. 194.53 for SDXL (Direct) and 194.55 for LSRNA–DemoFusion). We attribute this gap to low-resolution input latent (64×64 for 512 px, stride 8), which constrains recoverable detail in $\times 2$ setting; notably, our patch fidelity and text alignment are best in the row (pFID 191.75 and CLIP 0.738), indicating preserved local structure despite lower global scores. Moreover, native sampling is artifact-free at 1024^2 but produces artifacts in 9–19% of images at the $2048^2/4096^2$ scales LUA targets (see supplementary material).

Beyond 1K, LUA consistently improves both quality and speed. At 2048^2 , SDXL+LUA delivers the best fidelity among single-decode pipelines while remaining the fastest: FID 180.80, pFID 97.90, KID 0.0018, CLIP 0.764 in 3.52 s, outperforming SDXL+SwinIR (6.29 s) and substantially undercutting multi-stage re-diffusion (LSRNA–DemoFusion: FID 181.24 in 20.77 s; DemoFusion: 184.79 in 28.99 s). At 4096^2 , SDXL+LUA again sets the best single-pass fidelity (FID 176.90; pFID 61.80) with the lowest runtime (6.87 s), surpassing SDXL+SwinIR (7.29 s) and avoiding the quality collapse of direct high-res sampling (SDXL (Direct) FID 280.42). LSRNA–DemoFusion attains slightly lower pKID (0.0071) but is an order of magnitude slower (91.64 s). Fig. 6(a) confirms that LUA defines the Pareto frontier across 2048^2 – 4096^2 , and (b) shows improved detail and lower runtime over the training-free FreeScale [30].

Table 2 further shows that a single backbone generalizes across models and scales by adapting only the first convolution and briefly fine-tuning: at $\times 2$, FLUX+LUA reaches FID 180.99 and CLIP 0.773; at $\times 4$, SDXL+LUA yields KID 0.0015 while FLUX+LUA attains the strongest pFID (62.30). Fig. 7(a) quantifies the data efficiency of this transfer: with only 50k fine-tuning pairs LUA already matches the baselines, and beyond 250k it surpasses LSRNA, well below the 500k pairs used for our main results.

LUA is trained on latents encoded from real OpenImages crops but applied at inference to diffusion-sampled latents. Measured as the Wasserstein distance (W_1) between the two (Fig. 7(b, c)), this gap is about $4 \times$ larger at the 512^2 base used for 1024^2 outputs ($W_1=0.119$) than at the 1024^2 base used for 2048^2 outputs ($W_1=0.031$). The 2048^2 and 4096^2 pipelines upscale a 1024^2 base sampled at SDXL’s native resolution, whereas the 1024^2 pipeline upscales a 512^2 base sampled below it, where sampled latents drift further from the encoded ones.

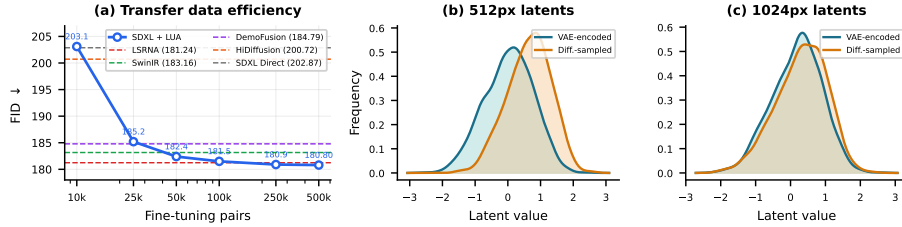


Fig. 7: (a) Transfer data efficiency (2048^2 , OpenImages): quality vs. number of fine-tuning pairs. (b, c) Latent distributions: VAE-encoded vs. diffusion-sampled.

Table 3: COCO-100 no-reference quality at 2048^2 . Best in **bold**.

Method	MANIQA $\uparrow$	MUSIQ $\uparrow$	CLIP-IQA $\uparrow$	NIQE $\downarrow$
HiDiffusion	0.425	59.75	0.586	5.06
DemoFusion	0.410	62.69	0.628	3.48
LSRNA–DemoFusion	0.343	59.60	0.513	4.09
SDXL (Direct)	0.414	64.89	0.608	4.55
SDXL + SwinIR	0.389	65.20	0.677	5.01
SDXL + LUA (ours)	0.431	65.34	0.682	5.74

This mirrors the measured trend: LUA is weakest at 1024^2 and strongest at the 2048^2 and 4096^2 scales it targets.

To assess generalization to a different dataset, we evaluate on COCO-100 (val2017) at 2048^2 . Because COCO reference images are low-resolution (~ 500 – 640 px), we report no-reference quality metrics (Table 3). SDXL+LUA attains the best MANIQA (0.431), MUSIQ (65.34), and CLIP-IQA (0.682) among all methods, with DemoFusion best on NIQE (3.48).

4.4 Qualitative Results

Fig. 8 presents comparisons at 2048^2 (top) and 4096^2 (bottom) from identical 1024^2 SDXL bases. Direct high-resolution sampling (*SDXL Direct*) exhibits duplicated structures and geometry drift, while *HiDiffusion* shows similar breakdowns at 4K, indicating that training-free escalation struggles at extreme scales. Among SR methods, pixel-space SwinIR sharpens but introduces ringing/haloing and plastic textures, and LSRNA–DemoFusion restores rich texture but requires a second diffusion stage with much higher latency. SDXL+LUA (ours) preserves edge continuity and microstructure with fewer artifacts and the lowest runtime.

A blind two-alternative forced-choice (2AFC) study corroborates these perceptual advantages. With 20 annotators casting 1,000 votes at 2048^2 , SDXL+LUA was preferred over every baseline: 81.9% against SDXL (Direct), 63.9% against HiDiffusion, 57.9% against LSRNA–DemoFusion, and 56.3% against DemoFusion, each significantly above chance ($p < 0.05$, binomial test).

LUA is content-agnostic: trained only on photographic crops, it upscales diverse domains without retraining (see supplementary material).

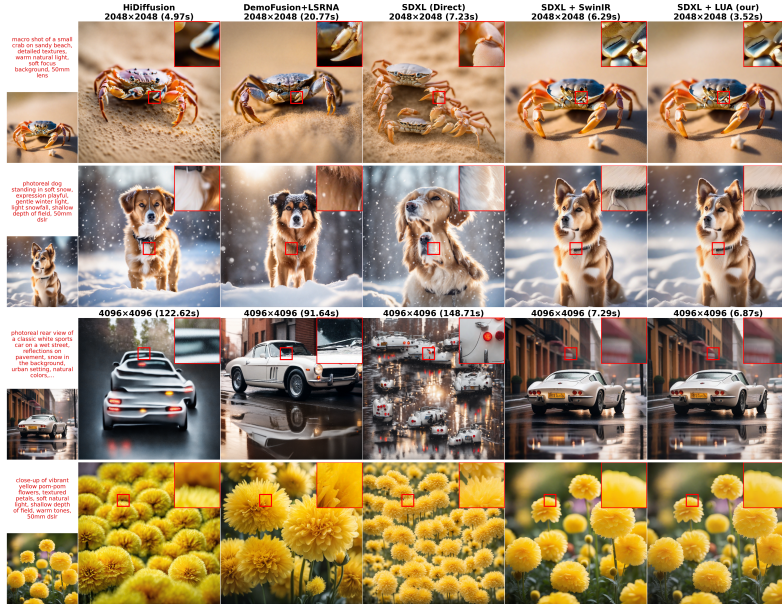


Fig. 8: Qualitative comparison at 2048^2 (top) and 4096^2 (bottom) from identical 1024^2 SDXL bases. Red boxes: $12\times$ crops; headers: per-image runtime. SDXL+LUA produces clean textures at the lowest latency, avoiding high-resolution artifacts of direct sampling and sharpening noise of pixel-space SR.

Table 4: Multi-stage curriculum ablation ($\times 2/\times 4$). Best in **bold**.

Configuration	PSNR $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$
Latent ℓ_1	28.53	0.198	26.16	0.236
I+II (w/o III)	28.96	0.172	26.67	0.213
I+III (w/o II)	31.05	0.150	27.10	0.198
II+III (w/o I)	31.60	0.145	27.40	0.192
Full (I+II+III)	32.54	0.138	27.94	0.184

4.5 Ablation Studies

Multi-stage training effectiveness. We assess the curriculum by removing individual stages, comparing against a latent ℓ_1 -only baseline (Table 4). The full configuration (I+II+III) yields the best fidelity at both scales (PSNR 32.54 / LPIPS 0.138 at $\times 2$, 27.94 / 0.184 at $\times 4$). Removing Stage III reduces edge sharpness and perceptual quality. Removing Stage II weakens latent-pixel consistency and increases artifacts. A pixel-only variant fails to converge with a frozen VAE.

Multi-scale super-resolution adaptation. We compare three design choices per scale factor: (i) an implicit, coordinate-based upsampler (LIIF), (ii) per-scale specialists trained for a single factor ($\times 2$ or $\times 4$), and (iii) a joint multi-head model sharing a backbone with scale-specific heads (Table 5). The joint multi-head design attains the best PSNR/LPIPS at both scales.

Table 5: Upsampling strategy ablation ($\times 2/\times 4$). Best in **bold**.

Variant	PSNR $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$
LIIF	29.10	0.210	26.10	0.235
Separated- $\times 2$	31.92	0.150	—	—
Separated- $\times 4$	—	—	27.71	0.189
Joint Multi-Head	32.54	0.138	27.94	0.184

5 Discussion

As a deterministic adapter, LUA exhibits three failure modes: *preserved hallucinations* (base-latent errors are upscaled, not corrected), *texture over-smoothing* (mitigated by textured training crops), and *dead pixels* (out-of-range latents decoding to localized corruptions, fixable by latent clamping). A promising direction is to jointly refine and upscale in latent space before decoding—e.g., lightweight consistency modules with uncertainty-aware gating to suppress artifacts while preserving semantics. Real-world degraded inputs, beyond the clean latents LUA currently targets, would require modeling such degradations in training. The same mechanism also serves image-to-image tasks such as depth-to-image or semantic maps, where structure preservation is critical. Finally, temporal consistency (e.g., recurrent latent refinement or temporal priors) would extend the adapter to high-resolution video synthesis.

6 Conclusion

We introduced LUA, a single-pass latent upscaler inserted between a pretrained generator and a frozen VAE decoder, and demonstrated that latent-space upscaling is trainable via a multi-stage latent-pixel curriculum with scale-specific $\times 2/\times 4$ heads. On OpenImages at 2048^2 and 4096^2 , SDXL+LUA achieves state-of-the-art single-decode fidelity (FID 180.80 / 176.90; pFID 97.90 / 61.80) while remaining the fastest: at 2048^2 it runs in 3.52 s versus 7.23 s for SDXL (Direct), outperforms pixel-space SR (SDXL+SwinIR), and approaches multi-stage re-diffusion quality at a fraction of the runtime. At 1024^2 , performance trails the strongest baselines due to the 64×64 input latent constraint, though patch-level fidelity remains competitive. A smaller-stride VAE would increase the latent information density and could narrow this gap, but requires retraining the VAE. A single backbone transfers across SDXL, SD3, and FLUX with only a first-layer channel change and brief fine-tuning, demonstrating robust cross-VAE and multi-scale generalization without modifying the generator or adding diffusion stages. Overall, these results establish single-decode latent upscaling as a practical alternative to multi-stage high-resolution pipelines.

Acknowledgements

We thank Pavel Chizhov (CAIRO, Technical University of Applied Sciences Würzburg-Schweinfurt, Germany) for discussions and manuscript revisions.

References

1. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: Multidiffusion: Fusing diffusion paths for controlled image generation (2023), <https://arxiv.org/abs/2302.08113>
2. Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., et al.: Flux. 1 kontekst: Flow matching for in-context image generation and editing in latent space. arXiv e-prints pp. arXiv-2506 (2025)
3. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans (2021), <https://arxiv.org/abs/1801.01401>
4. Cao, J., Wang, Q., Xian, Y., Li, Y., Ni, B., Pi, Z., Zhang, K., Zhang, Y., Timofte, R., Gool, L.V.: Ciaosr: Continuous implicit attention-in-attention network for arbitrary-scale image super-resolution (2023), <https://arxiv.org/abs/2212.04362>
5. Chai, L., Gharbi, M., Shechtman, E., Isola, P., Zhang, R.: Any-resolution training for high-resolution image synthesis (2022), <https://arxiv.org/abs/2204.07156>
6. Chen, W., Ma, Y., Liu, X., Yuan, Y.: Hierarchical generative adversarial networks for single image super-resolution. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 355–364 (2021)
7. Chen, X., Wang, X., Zhang, W., Kong, X., Qiao, Y., Zhou, J., Dong, C.: Hat: Hybrid attention transformer for image restoration (2025), <https://arxiv.org/abs/2309.05239>
8. Chen, Y., Liu, S., Wang, X.: Learning continuous image representation with local implicit image function (2021), <https://arxiv.org/abs/2012.09161>
9. Chen, Z., Liu, K., Gong, J., Wang, J., Sun, L., Wu, Z., Timofte, R., Zhang, Y., et al.: Ntire 2025 challenge on image super-resolution ($\times 4$): Methods and results (2025), <https://arxiv.org/abs/2504.14582>
10. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015)
11. Du, R., Chang, D., Hospedales, T., Song, Y.Z., Ma, Z.: Demofusion: Democratising high-resolution image generation with no \$. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6159–6168 (2024)
12. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
13. Fayyaz, M., Koohpayegani, S.A., Jafari, F.R., Sengupta, S., Joze, H.R.V., Sommerlade, E., Pirsivash, H., Gall, J.: Adaptive token sampling for efficient vision transformers. In: European conference on computer vision. pp. 396–414. Springer (2022)
14. Fuoli, D., Gool, L.V., Timofte, R.: Fourier space losses for efficient perceptual image super-resolution (2021), <https://arxiv.org/abs/2106.00783>
15. He, Y., Yang, S., Chen, H., Cun, X., Xia, M., Zhang, Y., Wang, X., He, R., Chen, Q., Shan, Y.: Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In: The Twelfth International Conference on Learning Representations (2023)
16. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium (2018), <https://arxiv.org/abs/1706.08500>

17. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33**, 6840–6851 (2020)
18. Jeong, J., Han, S., Kim, J., Kim, S.J.: Latent space super-resolution for higher-resolution image generation with diffusion models. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 2355–2365 (2025)
19. Jiang, X., Zhang, X., Gao, N., Deng, Y.: When fast fourier transform meets transformer for image restoration. In: *European Conference on Computer Vision*. pp. 381–402. Springer (2024)
20. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., Kolesnikov, A., Duerig, T., Ferrari, V.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision* **128**(7), 1956–1981 (Mar 2020). <https://doi.org/10.1007/s11263-020-01316-z>, <http://dx.doi.org/10.1007/s11263-020-01316-z>
21. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network (2017), <https://arxiv.org/abs/1609.04802>
22. Lee, J., Jin, K.H.: Local texture estimator for implicit representation function (2022), <https://arxiv.org/abs/2111.08918>
23. Li, H., Zou, Y., Wang, Y., Majumder, O., Xie, Y., Manmatha, R., Swaminathan, A., Tu, Z., Ermon, S., Soatto, S.: On the scalability of diffusion-based text-to-image generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9400–9409 (2024)
24. Li, H., Yang, Y., Chang, M., Feng, H., Xu, Z., Li, Q., Chen, Y.: Srdiff: Single image super-resolution with diffusion probabilistic models (2021), <https://arxiv.org/abs/2104.14951>
25. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 1833–1844 (2021)
26. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution (2017), <https://arxiv.org/abs/1707.02921>
27. Lin, X., He, J., Chen, Z., Lyu, Z., Dai, B., Yu, F., Ouyang, W., Qiao, Y., Dong, C.: Diffbir: Towards blind image restoration with generative diffusion prior (2024), <https://arxiv.org/abs/2308.15070>
28. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 14297–14306 (2023)
29. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis (2023), <https://arxiv.org/abs/2307.01952>
30. Qiu, H., Zhang, S., Wei, Y., Chu, R., Yuan, H., Wang, X., Zhang, Y., Liu, Z.: Freescale: Unleashing the resolution of diffusion models via tuning-free scale fusion (2025), <https://arxiv.org/abs/2412.09626>
31. Rad, M.S., Yu, T., Musat, C., Ekenel, H.K., Bozorgtabar, B., Thiran, J.P.: Benefiting from bicubically down-sampled images for learning real-world image super-resolution (2020), <https://arxiv.org/abs/2007.03053>
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021), <https://arxiv.org/abs/2103.00020>

33. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
34. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement (2021), <https://arxiv.org/abs/2104.07636>
35. Schusterbauer, J., Gui, M., Ma, P., Stracke, N., Baumann, S.A., Hu, V.T., Ommer, B.: FMBoost: Boosting Latent Diffusion with Flow Matching, pp. 338–355. Springer Nature Switzerland (Nov 2024). https://doi.org/10.1007/978-3-031-73030-6_19, http://dx.doi.org/10.1007/978-3-031-73030-6_19
36. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
37. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. In: International Conference on Machine Learning. pp. 32211–32252 (2023)
38. Sun, Y., Huang, Y., Yang, Z., Schneider, L.S., Thies, M., Gu, M., Mei, S., Bayer, S., Zöllner, F.G., Maier, A.: Eagle: an edge-aware gradient localization enhanced loss for ct image reconstruction. *Journal of Medical Imaging* **12**(1), 014001–014001 (2025)
39. Wang, J., Yue, Z., Zhou, S., Chan, K.C.K., Loy, C.C.: Exploiting diffusion prior for real-world image super-resolution (2024), <https://arxiv.org/abs/2305.07015>
40. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C.C., Qiao, Y., Tang, X.: Esrgan: Enhanced super-resolution generative adversarial networks (2018), <https://arxiv.org/abs/1809.00219>
41. Wu, R., Yang, T., Sun, L., Zhang, Z., Li, S., Zhang, L.: Seesr: Towards semantics-aware real-world image super-resolution (2024), <https://arxiv.org/abs/2311.16518>
42. Xie, L., Li, Y., Du, S., Xia, M., Wang, X., Yu, F., Chen, Z., Wan, P., Zhou, J., Dong, C.: Simplegvr: A simple baseline for latent-cascaded video super-resolution (2025), <https://arxiv.org/abs/2506.19838>
43. Yu, F., Gu, J., Li, Z., Hu, J., Kong, X., Wang, X., He, J., Qiao, Y., Dong, C.: Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild (2024), <https://arxiv.org/abs/2401.13627>
44. Zhang, S., Chen, Z., Zhao, Z., Chen, Y., Tang, Y., Liang, J.: Hidiffusion: Unlocking higher-resolution creativity and efficiency in pretrained diffusion models. In: European Conference on Computer Vision. pp. 145–161. Springer (2024)