# SHORT CIRCUIT VAE: USING PRE-TRAINED VARIATIONAL AUTOENCODERS FOR LATENT UPSCALING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Scaling modern diffusion models to high resolutions introduces a computational bottleneck that grows quadratically ($\Omega(N^2)$) with the spatial dimensions and frequently degrades structural realism. Pixel-space super-resolution (SR) models inherently require heavy architectures to process uncompressed RGB data, and risk semantic drift when generating fine details from scratch. Existing latent upscalers either require slow, multi-step denoising loops, or rely on external architectures that ignore the pre-trained features of the base model, demanding extensive training and often producing smoothed outputs. To overcome these limitations, we introduce Short-Circuit VAE (SC-VAE). Instead of building a complex upscaler from scratch, SC-VAE acts as a shortcut directly inside the pre-trained Variational Autoencoder (VAE). Specifically, we reuse the early layers of the VAE decoder to extract features and spatially upscale the latents. These upscaled features are passed through a small, trainable adapter, and then fed into the final layers of the VAE encoder to project them back into the standard latent format. By relying on the VAE's own boundary layers to format the data, our method safely upscales images while remaining fully compatible with the generative model.

## 1 INTRODUCTION

Modern diffusion transformers (DiTs) (Peebles & Xie, 2023) and flow-matching models currently dominate text-to-image synthesis, achieving impressive visual fidelity at 1K resolution. However, scaling these architectures to higher resolutions introduces severe computational and qualitative bottlenecks. Computationally, these models process images as flattened sequences of patches using massive self-attention mechanisms. The required memory scales quadratically with the sequence length ($\Omega(N^2)$), making direct high-resolution generation very slow and resource-intensive. Qualitatively, pushing these models beyond the resolutions they were trained on (native resolutions) severely degrades image realism.

Recent works attempt to solve these scaling issues through extensive architectural re-engineering, such as modifying positional encoding or training natively at massive resolutions. While these adaptations push the pixel limits, they do not fully resolve structural consistency. As we demonstrate in later visual comparisons, pushing models to natively generate at high resolutions often leads to fundamental generation issues, such as frequent anatomical artifacts and incoherent image compositions.

A common workaround is to generate a low-resolution base image and then apply a traditional pixel-space super-resolution (SR) network. Processing uncompressed RGB images requires more computational resources than operating on compact latents. Additionally, by leaving the generative model's valid data manifold, these methods must synthesize fine details entirely from scratch.

Other methods upscale a low-resolution latent to use as a structural anchor for a second diffusion process of overlapping windows of the upscaled image. Because this upscaling deviates from the valid data manifold, they must run slow, multi-step denoising loops to fix the latent. As we later show, this sacrifices the speed of latent upscaling and frequently introduces structural distortions.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To solve these issues, we propose **Short-Circuit VAE (SC-VAE)**. Instead of training a standalone upscaler from scratch, our method reuses the pre-trained layers of the Variational Autoencoder. We extract features from the VAE decoder, pass them through a lightweight adapter, and project them back into the latent space. Unlike pixel-space models that require high computational resources, SC-VAE operates efficiently on compact latents. Furthermore, unlike progressive latent methods that deviate from the valid data manifold and require slow multi-step denoising, reusing the VAE's native layers keeps our outputs strictly on the correct manifold. As a result, SC-VAE achieves fast, single-step super-resolution without structural distortions.

Finally, to resolve the tradeoff between geometric accuracy and perceptual sharpness, we optimize the adapter using a phased training curriculum. This approach first learns how to model the correct global structure and then learns to inject sharp, photorealistic micro-textures without corrupting the underlying image structure.

In summary, our main contributions are:

- We identify the inherent limitations of current high-resolution generation pipelines (i.e., native spatial scaling, pixel-space super-resolution, and latent upscaling).
- We introduce SC-VAE, a lightweight latent upscaler. It scales images efficiently by keeping the compute-heavy DiT steps at native resolutions and avoids the $\Omega(N^2)$ bottleneck by upscaling with a single feed-forward pass.
- We present a phased optimization curriculum that simultaneously achieves high geometric accuracy and perceptual realism.

## 2 Related Work

**Diffusion and Flow-Matching Models.** Diffusion models such as DDPM (Ho et al., 2020) and DDIM (Song et al., 2020) have significantly advanced the quality of image generation. Latent Diffusion Models (LDMs) (Rombach et al., 2022) and recent flow-matching architectures (Lipman et al., 2022) operate within a compressed latent space, achieving better efficiency. Although these models excel at generating images at their native resolutions, they struggle to generate higher-resolution images, often producing structural inconsistencies, lower image quality, and a much slower generation time. UltraFlux (Ye et al., 2026) adapts the core architecture and training process for native 4K outputs. However, despite these architectural changes, it still suffers from severe structural artifacts, such as duplicated objects or distorted anatomy (see appendix). Our method resolves this by combining the fast, high-fidelity generation of these models at native resolutions and moving the heavy spatial scaling out of the iterative generation loop using lightweight upscaler.

**Model Compression and Acceleration.** To reduce the high computational cost of inference, recent work explores model compression using quantization (Chen et al., 2025; Li et al., 2024; Shang et al., 2023), block pruning (Fang et al., 2025; Ma et al., 2024; Xie et al., 2025), and distillation (Feng et al., 2024; Zhang & Ma, 2024; Li et al., 2023) to reduce complexity without retraining, along with temporal acceleration methods that skip layers or reuse cached features. Although these strategies effectively reduce generation time, they fail to resolve the spatial bottleneck. Even a one-step distilled model often exceeds memory capacity when generating 4K images, as the quadratic computational cost of spatial scaling remains too massive for standard hardware. RALU (Jeong et al., 2026) spatially accelerates latent diffusion transformers by introducing a mixed-resolution sampling strategy. It identifies artifact-prone edge regions and upsamples them early, combining this with noise-timestep matching to prevent boundary artifacts while speeding up generation. However, RALU suffers from structural discontinuities and a dramatic degradation of realism at higher resolutions. In contrast, our method combines the efficient high-fidelity generation of DiTs at their native resolutions with a fast, lightweight upscaler.

**Pixel-Space Super-Resolution.** A common alternative is to generate a small image first and upscale it using external pixel-space super-resolution (SR) networks (Wang et al., 2021; Ho et al., 2021; Saharia et al., 2022). Processing uncompressed RGB data naturally requires significantly more computation and heavier network architectures compared to latent-space methods. Furthermore, because they leave the structural guidance of the data manifold, these networks have to learn how to generate fine details from scratch. This demands longer training times and risks semantic drift in fine details. Unlike these methods, SC-VAE operates within the compact latent space and reuses

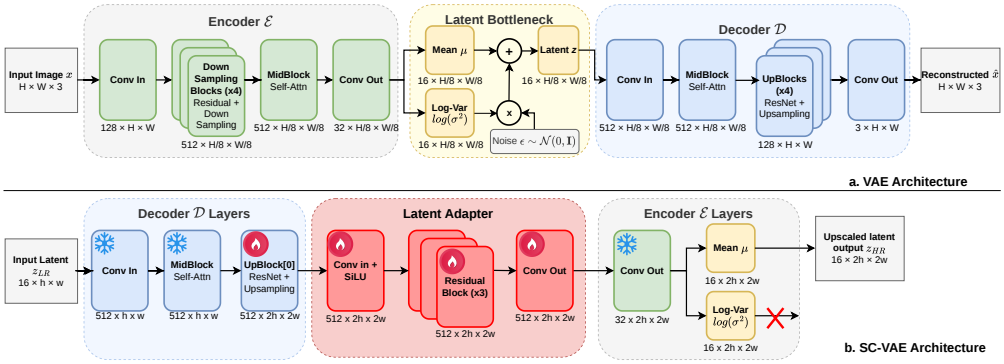



Figure 1: **Architecture overview**: (a) VAE architecture. The Encoder internal layers are in green, Decoder layers in blue, and the Encoder projection layers in yellow. (b) **SC-VAE** architecture. The red layers are the latent adapter layers, while the other colors mark the pre-trained VAE layers and where they were taken from. The dimensions under each layer describe the output sizes.

pre-trained VAE layers. This allows us to train a compact upscaler that preserves details much faster without the massive compute requirements.

**Reference-Based Latent Upscaling.** To preserve details while saving memory, recent training-free methods perform upsampling directly in the latent space. Approaches like DemoFusion (Du et al., 2024) and Pixelsmith (Tragakis et al., 2024) use an upsampled low-resolution latent reference to guide the high-resolution generation. However, simple latent upsampling disrupts the data manifold, which requires long, multi-step denoising loops to fix the structural distortions. While PixelRush (Lai et al., 2026) accelerates this process using one-step distilled models, it still needs to operate separately on many overlapping patches. This requires a lot of computation and applying a boundary blending mechanism. Our method keeps the upscaled features on the valid data manifold by reusing the VAE's pre-trained layers. This allows us to achieve high-quality super-resolution in a single feed-forward step, completely avoiding slow denoising loops, heavy spatial patching, and boundary blending.

**Trained Latent Upscalers.** To avoid slow denoising loops entirely, recent methods such as LSRNA (Jeong et al., 2025) and LUA (Razin et al., 2025) train dedicated networks to upscale latents directly. LSRNA trains a latent upsampler for manifold alignment, but it still requires region-wise noise addition (RNA) and multi-step diffusion to recover sharp details. Alternatively, LUA relies on a completely external SwinIR-style backbone, adding significant parameter overhead and bypassing the generative priors of the native autoencoder. This leads to hundreds of GPU hours of training to map the latent manifold. Instead of these external networks or iterative approaches, SC-VAE operates directly inside the frozen VAE. By reusing the domain-aligned layers of the native latent distribution, we construct a highly compact network that bridges the domain gap. This allows us to achieve superior, high-fidelity super-resolution in a single feed-forward step, removing the need for spatial patching, noise injection, or iterative loops, while dramatically reducing training time compared to external architectures.

# 3 Methodology

## 3.1 Preliminary

Variational Autoencoders (VAEs) are a fundamental cornerstone of latent diffusion and flow-matching models. By projecting the high-dimensional RGB pixel space into a lower-dimensional dense latent manifold, VAEs drastically reduce the computational complexity of the iterative generation process.

As illustrated in Figure 1a, the pre-trained FLUX VAE architecture consists of an Encoder ($\mathcal{E}$) and a Decoder ($\mathcal{D}$). The Encoder processes an input image $x \in \mathbb{R}^{H \times W \times 3}$ through a sequence of initial

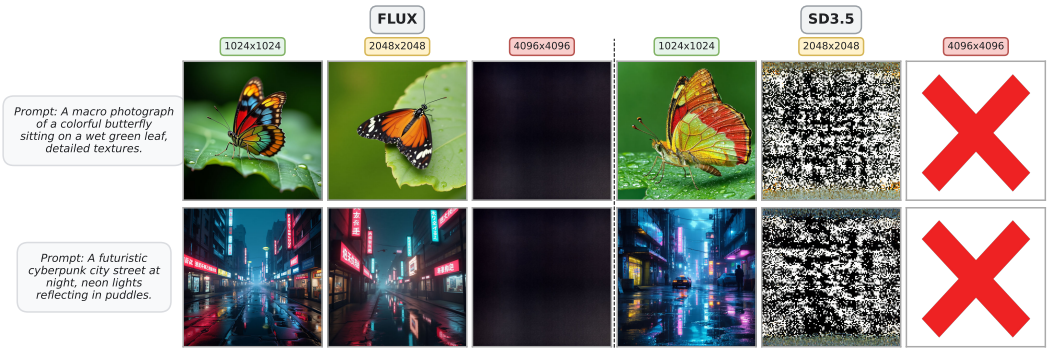



Figure 2: **Base model generation**: Flux and SD3.5 image generations at 1K/2K/4K. The best results are at 1K for both models. Flux is able to generate smoother, less detailed images at 2K and struggles at 4K. SD3.5 is unable to generate good images at 2K and cannot generate 4K images due to architecture limitations.

convolutions, residual downsampling blocks, and a self-attention mid-block to extract rich semantic features. The final convolutional layer projects these features into a 32-channel output, which is split into the mean ($\boldsymbol{\mu}$) and log-variance ($\log(\boldsymbol{\sigma}^2)$). A 16-channel latent representation $\boldsymbol{z}$ is then sampled using the formula:

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ is an element-wise multiplication.

The Decoder $\mathcal{D}$ receives a compressed latent $z \in \mathbb{R}^{c \times h \times w}$, where $c$ is the number of latent channels, $h = H/s, w = W/s$ and $s$ is the spatial downscaling factor of the encoder. In the decoder, $z$ is passing through a self-attention mid-block and a sequence of residual upsampling blocks to reconstruct the original RGB image $\hat{x}$. While highly effective for baseline image generation, utilizing the full decoder for super-resolution tasks necessitates mapping back to the pixel space, which discards the semantic density of the latent space and incurs massive computational overhead.

### 3.2 Short-Circuit VAE

Our goal is to perform high-fidelity super-resolution entirely within the latent space of a pre-trained latent image generation model, specifically leveraging the Autoencoder (VAE) architecture. Given a low-resolution latent tensor $\boldsymbol{z}_{LR} \in \mathbb{R}^{c \times h \times w}$, our objective is to predict a high-resolution latent $\boldsymbol{z}_{HR} \in \mathbb{R}^{c \times 2h \times 2w}$. This approach bypasses the computational bottleneck of generating high-resolution images directly through the Diffusion Transformer (DiT), where the self-attention mechanism creates a strict quadratic lower bound $\Omega(N^2)$, relative to the sequence length $N$. The base models excel at maintaining high-fidelity at the resolutions on which they were trained ($\boldsymbol{z}_{LR}$), but struggle to maintain it when generating higher-resolution images, as seen in Figure 2. This approach obtains the high-fidelity generation of the base model at $\boldsymbol{z}_{LR}$ and a high-resolution output ($\boldsymbol{z}_{HR}$).

A naive approach would decode the latent to RGB pixel space and re-encode it at a higher resolution. However, this adds massive memory overhead and inherently degrades high-frequency details due to the lossy compression bottleneck of the VAE. To avoid these issues, we introduce **Short-Circuit VAE (SC-VAE)**, an efficient latent upscaler comprised of pre-trained VAE layers paired with a residual latent adapter. We reuse the VAE's frozen layers because they are optimized to navigate this specific latent space. Instead of training a new network on how to extract and project latent data from scratch, we repurpose the early decoder layers to extract spatial features, and the terminal encoder layers to project the upscaled features back into a valid latent format. Using the native boundary layers of the VAE for feature extraction and projection, we structurally constrain the output to the exact distribution expected by the downstream diffusion model. This allows the lightweight adapter to focus solely on spatial upscaling without risking manifold drift. The importance of the pre-trained weights of the used VAE layers is demonstrated in the experiments section.

### 3.2.1 ARCHITECTURE

The SC-VAE architecture consists of three sequential stages: feature extraction, residual adaptation, and latent projection. First, the early layers of the frozen VAE decoder take the low-resolution latent $\boldsymbol{z}_{LR} \in \mathbb{R}^{16\times h\times w}$, extract a feature map $\mathbb{F} \in \mathbb{R}^{512\times h\times w}$, and spatially upscale it to $\mathbb{F}' \in \mathbb{R}^{512\times 2h\times 2w}$. While the decoder effectively handles this spatial upscaling, the expansion inherently alters the feature distribution. Because the decoder is optimized to progressively map representations from the latent manifold toward the RGB pixel space, passing through its upsampling blocks naturally shifts $\mathbb{F}'$ away from the latent distribution expected by the encoder. To bridge this semantic gap, we pass $\mathbb{F}'$ through a trainable residual adapter comprised of 3 residual blocks with 512-channel convolutions and SiLU activations. The adapter acts as a translator, realigning the upscaled features back to the target latent distribution. Finally, the layers of the frozen VAE encoder process these adapted features. The encoder compresses the 512-channel feature map back into the standard 16-channel format, projecting it into the final high-resolution latent $\boldsymbol{z}_{HR} \in \mathbb{R}^{16\times 2h\times 2w}$. An overview of our architecture is illustrated in Figure 1b.

### 3.2.2 TRAINING

**Dataset Preparation.** As mentioned above, our model takes a low-resolution latent $z_{LR}$ as input and outputs an upscaled high-resolution latent $z_{HR}$. For that, we would need to have a dataset of $z_{LR}, z_{HR}$ pairs, where each pair consists of low-resolution and high-resolution latents of an image $x_{LR}, x_{HR}$.

To address this, we use high-resolution, real-world images from the OpenImages and DIV2K datasets. To guarantee multi-scale robustness and eliminate the domain gap between macro-structures and micro-textures, we introduce significant scale variability. For each image in the dataset, we: (i) downsample the image to various scales, (ii) apply several random crops at each scale and its respective low-resolution scale, and (iii) encode the LR-HR pair independently using a pre-trained VAE, yielding a highly diverse dataset of both zoomed-in textures and heavily compressed global scenes. For more information, see the supplementary material.

**Multi-Phase Optimization Curriculum.** Training the adapter directly on decoded RGB images is computationally expensive and suffers from significantly slower convergence. Conversely, training solely in the latent space produces decoded images that suffer from grid-like artifacts and lack fine details. To solve this, we split the training into a two-phase curriculum: first, we establish the overall structure in the latent space, and second, we refine the visual details in the pixel space. We empirically validate the need and efficiency of this two-phase approach in our ablation study.

We denote the input and predicted upscaled latents as $\mathbf{z}_{LR}$ and $\hat{\mathbf{z}}_{HR}$, respectively. The ground-truth reference is $\mathbf{z}_{HR}$, and $\hat{\mathbf{x}} = \mathcal{D}(\hat{\mathbf{z}}_{HR})$ and $\mathbf{x}_{HR} = \mathcal{D}(\mathbf{z}_{HR})$ represent their decoded RGB counterparts via the decoder $\mathcal{D}$.

**Phase I: Latent Structure Initialization.** In the first phase, we train the adapter strictly in the latent domain to learn the base geometric mapping. We define the objective as a combination of absolute distance and frequency alignment:

$$\mathcal{L}_{\text{Phase I}} = \lambda_1 \mathcal{L}^z_{\text{L1}} + \lambda_2 \mathcal{L}^z_{\text{FFL}} \tag{2}$$

The $L_1$ loss enforces point-wise accuracy to anchor the global structure. Simultaneously, the Focal Frequency Loss (FFL) (Jiang et al., 2021) minimizes the error between the predicted and target latents in the frequency domain. FFL uses a dynamic weighting matrix that adaptively targets the specific frequencies the network struggles to synthesize. As the network quickly learns the easy low frequencies, FFL automatically down-weights them, shifting the heavy loss penalties entirely onto the harder, high-frequency errors. This focused penalty prevents the $L_1$ loss from collapsing the output into a blurry average by forcing the network to recover fine-grained textures.

**Phase II: Pixel-Space Perceptual Enhancement.** Once the adapter reliably predicts the foundational latent structure, we introduce the frozen VAE decoder $\mathcal{D}$ to compute losses directly on the RGB output space. The goal of this phase is to inject perceptual realism without compromising the previously established geometry. The objective expands to:

$$\mathcal{L}_{\text{Phase II}} = \gamma_1 \mathcal{L}^x_{\text{L1}} + \gamma_2 \mathcal{L}^x_{\text{FFL}} + \gamma_3 \mathcal{L}^x_{\text{HF}} + \gamma_4 \mathcal{L}^x_{\text{LPIPS}} \tag{3}$$

Here, the pixel-space $L_1$ and FFL continue to preserve the overall global structure. To explicitly isolate and improve edge sharpness, we introduce a high-frequency residual loss, $\mathcal{L}^x_{\text{HF}}$. This loss subtracts a Gaussian-blurred (low-frequency) version of the image from the original to isolate the sharp high-frequency edges, and then computes the $L_1$ distance between these residuals:

$$\mathcal{L}^x_{\text{HF}} = \|(\hat{\mathbf{x}} - B(\hat{\mathbf{x}})) - (\mathbf{x}_{HR} - B(\mathbf{x}_{HR}))\|_1 \tag{4}$$

where $B(\cdot)$ denotes a Gaussian blur operation. Finally, $\mathcal{L}^x_{\text{LPIPS}}$ acts as the primary perceptual objective. It guides the network to synthesize visually convincing micro-details—such as film grain or biological textures—that align closely with human visual assessment standards.

We keep most of the pre-trained VAE layers frozen, while fine-tuning the decoder upscaler block. The upscaler block is originally intended to start the spatial upscaling of the latents back to their original pixel-space shape while shifting the latent distribution back to the pixel-space distribution, so we are fine-tuning it to try to preserve the spatial upscaling but reduce the distribution shift before passing through the latent adapter. Information about the frozen and trainable parts is shown in Figure 1.

# 4 Experiments

We detail our experimental setup (Section 4.1) and present both quantitative (Section 4.2) and qualitative (Section 4.3) comparisons with other baselines and a user study for visual human evaluation (Section 4.4). We also demonstrate the importance of various aspects of our method with ablation studies (Section 4.5).

## 4.1 Experiment Settings

We evaluate our method against several high-resolution generation baselines to provide a comprehensive comparison. First, we evaluate direct generation using FLUX at high resolution as a naive baseline. To represent reference-based latent upscaling methods, we compare against DemoFusion and Pixelsmith. We also compare against LSRNA and LUA to evaluate against trained latent upscalers, UltraFlux as a state-of-the-art native high-resolution baseline, and RALU as a representative spatial acceleration framework.

All methods are evaluated using the standard parameters in their official code release, which we detail in the supplementary material. The 2K experiments are conducted on a single NVIDIA RTX 6000 Ada Generation GPU except for the FLUX2 method, and the 4K experiments are conducted on a single NVIDIA RTX PRO 6000 Blackwell Max-Q GPU due to out-of-memory in some of the compared baselines and in the native base generation of FLUX2. PixelRush does not have a public release code at the time of writing this paper.

## 4.2 Quantitative Comparisons

**Metrics.** Following previous work (Jeong et al., 2025; Razin et al., 2025), we use FID (Heusel et al., 2017), KID (Bińkowski et al., 2018), and CLIP (Radford et al., 2021) to evaluate the semantic alignment of the generated images with ground-truth images. FID and KID compress the images to $299 \times 299$ for the input of the Inception network (Szegedy et al., 2015). Since we measure our metrics on high-resolution images, we use pFID and pKID for patch-based evaluation to help evaluate fine details. Specifically, patch-based evaluation extracts multiple local sub-regions (patches) from the high-resolution image at their original pixel scale and evaluates them through the network, preventing aggressive downsampling from discarding high-frequency textures. We also measure the mean generation time of the images.

**Dataset.** Our evaluation at 2K resolution is done with a test set of 755 images. All images are high-resolution (both height and width larger than 2048 pixels) from the OpenImages dataset, which do not appear in our training set. For 4K resolution, we use a test set of 1000 images from the MultiAspect-4K-1M dataset that was introduced in the UltraFlux paper. All images have both height and width larger than 4096 pixels. The images were captioned with JoyCaption (FancyFeast, 2025) and the captions were used as prompts for the baseline generation evaluation. See the supplementary material for more details.

Table 1: Quantitative comparison of generative models and upscalers. All 4K experiments and both FLUX2 experiments are conducted on a single NVIDIA RTX PRO 6000 Blackwell Max-Q GPU, while the remaining 2K experiments are conducted on a single NVIDIA RTX 6000 Ada Generation GPU.

| Resolution | Model | FID ↓ | pFID ↓ | KID ↓ | pKID ↓ | CLIP ↑ | Time (s) |
|---|---|---|---|---|---|---|---|
| $2048 \times 2048$ | FLUX Direct | 84.323 | 92.658 | 0.00601 | 0.00942 | 0.3432 | 114.75 |
| | Pixelsmith | 81.840 | 87.479 | 0.00420 | 0.00397 | 0.3491 | 75.41 |
| | DemoFusion | 83.410 | 88.852 | 0.00482 | 0.00447 | 0.3458 | 133.21 |
| | LSRNA | 81.952 | 87.030 | 0.00424 | **0.00249** | 0.3502 | 63.57 |
| | RALU | 83.226 | 88.681 | 0.00643 | 0.00578 | 0.3378 | 83.95 |
| | LUA | 79.826 | 93.117 | 0.00538 | 0.00901 | 0.3423 | 24.17 |
| | UltraFlux | 78.832 | 87.185 | 0.00489 | 0.00585 | 0.3425 | 40.93 |
| | **Ours (FLUX1)** | 78.334 | 87.062 | 0.00403 | 0.00574 | 0.3431 | **19.92 + 1.02** |
| | **Ours (FLUX2)** | **74.353** | **81.949** | **0.00266** | 0.00323 | **0.3503** | 35.90 + 1.05 |
| $4096 \times 4096$ | Pixelsmith | 60.381 | 84.941 | 0.00199 | 0.00925 | 0.3487 | 226.42 |
| | DemoFusion | 66.904 | 87.080 | 0.00479 | 0.01109 | 0.3439 | 602.81 |
| | LSRNA | 61.174 | 80.481 | 0.00305 | 0.00614 | 0.3483 | 165.72 |
| | RALU | 97.232 | 139.412 | 0.02079 | 0.04795 | 0.2994 | 504.33 |
| | LUA | 59.485 | 87.409 | 0.00327 | 0.01068 | 0.3429 | 18.37 |
| | UltraFlux | 57.414 | 74.342 | 0.00191 | 0.00284 | 0.3430 | 63.13 |
| | **Ours (FLUX1)** | 59.411 | 84.098 | 0.00316 | 0.00890 | 0.3430 | **10.76 + 4.75** |
| | **Ours (FLUX2)** | **54.082** | **72.229** | **0.00184** | **0.00334** | **0.3505** | 35.90 + 4.81 |

**Comparison.** Table 1 presents the numeric results of our method, as well as other baselines. We also used our logic to train a model by reusing the FLUX2 VAE layers with the exact same architecture of SC-VAE, which we evaluate in this table (see more details in the supplementary material). We see that at 2K, all methods beat the FLUX FID. Pixelsmith, DemoFusion and LSRNA achieve low patch-based scores (pFID, pKID) and high CLIP scores. We attribute it to their denoising of individual crops of the image with the same prompt. These methods require long generation times and create severe global artifacts that are discussed in the following sections. RALU accelerates FLUX' generation process, but fails to scale to 4K, showing a large drop of quality. LUA shows good global metrics (FID/KID) and a fast generation, but sacrifices patch-based performance. UltraFlux produces a good combination of patch-based metrics and global metrics, but requires longer generation times and suffers from noticeable artifacts, shown in the following sections. SC-VAE combines good global and patch-based metrics with text-alignment and a fast speed on the original FLUX1 model, and on FLUX2 further improves all metrics except time. We attribute the great improvement on the FLUX2 VAE model mostly to the more expressive latent space of the FLUX2 VAE that has more latent channels (see the supplementary material). The generation times of our methods show the base model native generation time and add the SC-VAE and final VAE decoding times. The FLUX2 long native generation times demonstrate how our lightweight architecture is even more crucial for newer, compute-heavy models.

### 4.3 Qualitative Comparisons

Figure 3 illustrates the visual quality of our proposed SC-VAE framework compared to state-of-the-art reference-based upscalers and direct native generation at 2K/4K resolution.

Pixelsmith, DemoFusion, and LSRNA all rely on separate denoising iterations for local image crops. Looking at the top image for these methods, we can see a lack of a realistic global structure and a poor adherence to the prompt. None of the images has a clear hood of a car. The bottom images have a clearer structure but suffer from catastrophic artifacts like a small third leg for Pixelsmith, and additional small characters at the bottom right for both DemoFusion and LSRNA. We further demonstrate this in the supplementary material.

RALU offers a more efficient generation process than the base FLUX pipeline but fails to produce realistic images at higher 4K, as seen in the bottom row. LUA preserves rigid geometric struc-

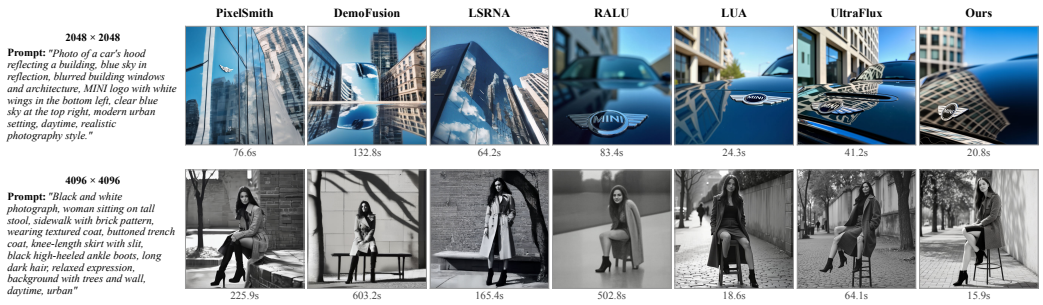



Figure 3: **Qualitative comparison**: Generation comparison at 2K/4K across several baselines. Each row is generated at the same resolution using the same prompt.

ture through its Swin-based architecture, but its reliance on strictly deterministic regression losses suppresses generative hallucination, leaving complex textures appearing slightly out-of-focus and overly safe and introduces some visual glitches, as seen more closely in the supplementary material. UltraFlux excels at producing sharp images but suffers from noticeable artifacts. The top image car seems less realistic than in other baselines and the woman at the bottom image has three legs.

In contrast, our SC-VAE framework achieves a superior balance of geometric stability and perceptual realism. It prevents the color drift and structural distortions seen in other methods, which we attribute to the fact that it operates within the native VAE layers. Furthermore, our targeted high-frequency optimization injects sharp, natural micro-textures directly onto this correct geometry. As seen in the provided examples, SC-VAE accurately preserves the car's hood and the woman's anatomy, producing photorealistic surface details without noise or artificial over-sharpening, visually outperforming competing methods at a high speed.

### 4.4 User Study

We conduct a user study comparing SC-VAE with three leading baselines: LSRNA as the state-of-the-art reference-based refinement method, and LUA and UltraFlux as the top competing baselines. We need this human validation because standard metrics often fail to capture structural distortions and visual artifacts. In each trial, participants are presented with a text prompt and the anonymous output of each method. We ask users to rate each image on a scale from 1 to 5 based on two metrics: overall visual realism and text-to-image alignment. As shown in Table 2, SC-VAE consistently achieves the highest average human preference scores for both metrics, rated by our 42 participants. Further details on the study and extended results are provided in the supplementary material.

Table 2: User study results. We report the average score for both realism and text-alignment for all the compared baselines.

| Metric | LSRNA | UltraFlux | LUA | **Ours** |
|---|---|---|---|---|
| Realism | 2.08 | 3.36 | 3.50 | **3.89** |
| Text-alignment | 2.32 | 3.55 | 3.89 | **4.12** |

### 4.5 Ablation Study

**Importance of Pre-trained Priors and Initialization.** Table 3 demonstrates the importance of the pre-trained VAE weights, upscaler architecture, and our adapter initialization decision. In the **Standard Init** configuration, we discard the pre-trained weights of the FLUX VAE decoder and initialize all layers (including the backbone and adapter) using standard PyTorch distributions. In the **Norm Init** configuration, we initialize the newly introduced convolutional layers by explicitly scaling their weights to match the L2 norm of the upcaler block's preceding boundary layer (Salimans & Kingma, 2016). Although this dynamic scaling may prevent catastrophic gradient explosions, it still introduces immediate structural friction at step zero. By injecting non-zero feature translations from step

Table 3: Quantitative comparison of the ablations done.

| Model | FID ↓ | pFID ↓ | KID ↓ | pKID ↓ | CLIP ↑ |
|---|---|---|---|---|---|
| Standard Init | 96.921 | 109.341 | 0.01825 | 0.02218 | 0.3210 |
| Norm Init | 81.754 | 92.366 | 0.00559 | 0.00892 | 0.3417 |
| Custom Upscaler | 81.931 | 93.229 | 0.00584 | 0.00868 | 0.3420 |
| **SC-VAE** | **78.334** | **87.062** | **0.00403** | **0.00574** | **0.3431** |

Table 4: Ablation of training phases.

| Model | FID ↓ | pFID ↓ | KID ↓ | pKID ↓ | CLIP ↑ |
|---|---|---|---|---|---|
| Phase I Only | 81.971 | 91.226 | 0.00497 | 0.00879 | 0.3417 |
| Phase II Only | 79.803 | 91.404 | 0.00512 | 0.00881 | 0.3421 |
| Full Training | **78.334** | **87.062** | **0.00403** | **0.00574** | **0.3431** |

zero, the network suffers minor structural drift. In the **Custom Upscaler** setting, we replace the VAE's native pre-trained upscaler block with an external, trainable spatial upscaler. While it maintains a high text-alignment (CLIP 0.3420), it has a lower fidelity with real image distribution. The table shows results on 2K images using the same evaluation procedure as in Table 1.

**Curriculum Learning.** We assess our curriculum learning by isolating the individual phases, as detailed in Table 4. The full configuration yields the highest perceptual and geometric fidelity. Training exclusively in Phase I (latent-only) achieves competitive distributional metrics (FID 81.971) but produces severe grid-like screen-door artifacts due to the lack of pixel-space adjustments. Conversely, bypassing Phase I and training exclusively in Phase II (pixel-only) from initial adapter weights cause a gradient shock through the frozen VAE, which makes it harder for the model to converge. Additionally, phase II requires much slower and computationally expensive training due to the need to decode all network outputs and propagate gradients through the decoder. Thus, phase I provides an efficient foundation for the slower refinement in phase II. The dual-phase curriculum resolves these problems, establishing a stable latent geometry before injecting pixel-space micro-textures. The table shows results on 2K images using the same evaluation procedure as in Table 1.

## 5 Limitation and Future Work

Because SC-VAE is designed purely as a fast spatial upscaler, it cannot fix structural mistakes made by the base generative model. If the base model generates distorted limbs or a bad composition, SC-VAE does not remove those errors. Future research could explore hybrid models that use sparse global attention to correct these structural mistakes before upscaling, while still keeping computational costs low.

## 6 Conclusion

In this paper, we introduced Short-Circuit VAE (SC-VAE), a fast latent super-resolution method that avoids the massive computational cost of high-resolution DiTs. Instead of using slow iterative loops or heavy pixel-space models, SC-VAE acts as a direct shortcut inside the pre-trained VAE. By connecting the VAE's early decoder and late encoder layers with a lightweight adapter, our model only needs a short and efficient training. Our experiments demonstrate that SC-VAE generates photorealistic 2K and 4K images faster and with better visual quality than existing baselines with only a single feed-forward step.

### AI USE STATEMENT

In this work, we used generative AI tools to implement methods (specifically, assistance in implementing the training code) and to clean and reformat the dataset. We have *not* used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, support qualitative and thematic data analysis or interpret results. The tasks of generating synthetic datasets (we use a dataset of real images), formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools to edit scientific figures, summarize existing literature, and edit the manuscript to improve readability. We have reviewed all AI-assisted work. All LLM-generated code was verified and tested for correctness by the main author and a manual literature survey was done in addition to our LLM assistance with the search. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

Our code will be publicly published upon acceptance. Our network architecture is described in Figure 1 and Section 3.2.1. To learn about our training, see Section 3.2.2 which contains information on data preparation and our multi-phase training procedure. We provide more information about the data preparation in Section A.1, and the chosen hyperparameters for the training are in Section A.2 and Table 5.

## REFERENCES

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-dit: Accurate post-training quantization for diffusion transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28306–28315. IEEE, 2025.

Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion: Democratising high-resolution image generation with no $$$. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6159–6168. IEEE, 2024.

FancyFeast. Joycaption: An open-source vision-language model for detailed image captioning. `https://huggingface.co/fancyfeast/llama-joycaption-beta-one-hf-llava`, 2025. Accessed: August 2026.

Gongfan Fang, Kunjun Li, Xinyin Ma, and Xinchao Wang. Tinyfusion: Diffusion transformers learned shallow. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18144–18154. IEEE, 2025.

Weilun Feng, Chuanguang Yang, Zhulin An, Libo Huang, Boyu Diao, Fei Wang, and Yongjun Xu. Relational diffusion distillation for efficient image generation. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 205–213, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021.

Jinho Jeong, Sangmin Han, Jinwoo Kim, and Seon Joo Kim. 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.

Wongi Jeong, Kyungryeol Lee, Hoigi Seo, and Se Young Chun. Training-free mixed-resolution latent upsampling for spatially accelerated diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18673–18682, 2026.

Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *ICCV*, 2021.

Hong-Phuc Lai, Phong Nguyen, and Anh Tran. Pixelrush: Ultra-fast, training-free high-resolution image generation via one-step diffusion. *arXiv preprint arXiv:2602.12769*, 2026.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024.

Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems*, 36:20662–20678, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching. *Advances in Neural Information Processing Systems*, 37: 133282–133304, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

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 *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Aleksandr Razin, Danil Kazantsev, and Ilya Makarov. One small step in latent, one giant leap for pixels: Fast latent upscale adapter for your diffusion models. *arXiv preprint arXiv:2511.10629*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 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.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1972–1981. IEEE, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

Athanasios Tragakis, Marco Aversa, Chaitanya Kaul, Roderick Murray-Smith, and Daniele Faccio. Is one gpu enough? pushing image generation at higher-resolutions with foundation models. In *NeurIPS*, 2024.

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 1905–1914. IEEE, 2021.

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Chengyue Wu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *arXiv preprint arXiv:2501.18427*, 2025.

Tian Ye, Song Fei, and Lei Zhu. Ultraflux: Data-model co-design for high-quality native 4k text-to-image generation across diverse aspect ratios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22070–22079, 2026.

Linfeng Zhang and Kaisheng Ma. Accelerating diffusion models with one-to-many knowledge distillation. *arXiv preprint arXiv:2410.04191*, 2024.

# Supplementary Material

## A Training Information

This section details our training process, starting by describing the preparation of the dataset (Section A.1) and followed by the selection of hyperparameters (Section A.2).

### A.1 Data Preparation

As mentioned in the methodology section, we train our model using a dataset of $z_{LR}, z_{HR}$ latent pairs. We construct this dataset using high-resolution images from the OpenImages and DIV2K datasets, filtering out images smaller than 1024 pixels in either dimension before dividing them into train, validation and test splits. We set a list of image resolutions: native, 4096, 2048, 1024, 512, where native is the original image resolution. For each image, we apply a center-crop based on its shortest dimension (for example, a $3000 \times 2500$ image will be center-cropped at a $2500 \times 2500$ resolution). We then resample all images to all valid lower resolutions ($2048^2, 1024^2, 512^2$ in our example) using `Resampling.LANCZOS`. Finally, we extract $512^2$ non-overlapping crops from each resolution, resample each crop to a $256^2$ resolution and use a VAE encoder to generate our $z_{LR}, z_{HR}$ pairs. We find this diversity of resolutions to be helpful, since it maintains a uniform size for all training samples while extracting zoomed-in textures from the higher resolutions, and broad, global scenes from the lower resolutions. Our final training set consists of approximately 1.1 million samples.

### A.2 Hyperparameter Selection

To optimize our training hyperparameters, we conduct an extensive search using a subset of 20K random training pairs and 250 validation pairs that are processed in the same way as the training pairs. Using Optuna TPE-sampler search, we search for our optimized loss function weights, batch size, and learning rate. Our search consists of 100 trials of 1000 training steps each, with early pruning for quicker trials. Our objective is a combination of LPIPS (AlexNet) and PSNR. We apply our optimized parameters to Phase I of our training, and we further use the weights of Phase I as the baseline for the hyperparameter optimization of Phase II. We use cosine annealing for our learning rate, starting from the optimized learning rate. As shown in the Methodology section, our Phase I training objective is the following:

$$\mathcal{L}_{\text{Phase I}} = \lambda_1 \mathcal{L}^z_{\text{L1}} + \lambda_2 \mathcal{L}^z_{\text{FFL}} \tag{5}$$

We set our optimized hyperparameters for this phase to $\lambda_1 = 1.0$, $\lambda_2 = 0.1$. For Phase II, our training objective is the following:

$$\mathcal{L}_{\text{Phase II}} = \gamma_1 \mathcal{L}^x_{\text{L1}} + \gamma_2 \mathcal{L}^x_{\text{FFL}} + \gamma_3 \mathcal{L}^x_{\text{HF}} + \gamma_4 \mathcal{L}^x_{\text{LPIPS}} \tag{6}$$

We set our optimized hyperparameters for this phase to $\gamma_1 = 10.0$, $\gamma_2 = 30.0$, $\gamma_3 = 1.0$, $\gamma_4 = 1.0$.

Our training was conducted on a single NVIDIA RTX PRO 6000 Blackwell Max-Q GPU with a total training time of approximately 29.4 GPU-hours. Table 5 provides additional training information.

Table 5: Hyperparameters used for the training.

| Phase | Iterations | Batch size | Time | Optimizer | Weight decay | Initial LR |
|---|---|---|---|---|---|---|
| Phase I | 100K | 128 | 13.31h | AdamW | 1e-2 | 2e-4 |
| Phase II | 50K | 16 | 16.11h | AdamW | 1e-2 | 5e-5 |

# B Experiments Information

## B.1 Test Set Preparation

**Test Images.** We evaluate our method against other baselines at both $2048^2$ and $4096^2$ resolutions. For the 2K evaluation, we use our test split from the dataset mentioned in Section A.1. Out of the test split, we use all 755 images that have both dimensions larger than 2048 pixels. Because this test set lacks sufficient 4K samples, we use an additional test set of 1000 images from the MultiAspect-4K-1M dataset (introduced in the UltraFlux paper). All of the images have both dimensions larger than 4096 pixels.

**Image Captioning.** We caption all the test images from both the 2K and 4K datasets using JoyCaption with the following query, inspired by the LUA paper:

```
You are an image captioning model for text-to-image generation
models.  Describe the image in 1-2 sentences, focusing on concrete
objects, their attributes, layout, and photographic style.  Your
caption will be used as a prompt to generate the image with a
text-to-image model.
```

We set a minimum of 15 tokens and a maximum of 60 tokens for the reply.

## B.2 Competing Baselines Generation Details

For our comparisons against other baselines, we use their official code release with their standard generation parameters. In this section, we specify them for each method.

**FLUX Direct.** We use the FLUX1.dev release with the standard parameters, including 28 inference steps and a guidance scale of 3.5.

**Pixelsmith.** We use similar parameters to the given demo in the official code, including 50 inference steps, a guidance scale of 7.5, a slider of 22 for the 2K generation and a slider of 34 for the 4K generation.

**DemoFusion.** As in the official code, we use 50 inference steps, a guidance scale of 7.5, a stride of 64, cosine scales of $(3, 1, 1)$ and a sigma of 0.8.

**LSRNA.** We utilize the public release of the LSRNA-DemoFusion method. We use the exact same parameters as in the DemoFusion generation, and for the LSRNA-specific parameters we use a stride ratio of 0.5, inversion depth of 30 and RNA minimum and maximum standard deviation of 0.0 and 1.2 respectively.

**RALU.** We use the RALU FLUX pipeline with 'use RALU default' as True, which sets the following parameters: speedup level 4x, N=[5,6,7], e = [0.3, 0.45, 1.0], shift = [5.019, 2.589, 2.226], Z = 6.317, up ratio = 0.3 alongside 28 inference steps and a guidance scale of 3.5.

**LUA.** We use the official LUA code release for FLUX with 28 inference steps and a guidance scale of 3.5. We generate 2K images with the x2 head and 4K images with the x4 head, both with a base resolution of 1K.

**UltraFlux.** We run the official UltraFlux pipeline with 50 inference steps and a guidance scale of 4.0.

**SC-VAE.** For our base generation we generate a 1K latent with 28 inference steps on both FLUX1 and FLUX2, and we use a guidance scale of 3.5 for FLUX1 and 4.0 for FLUX2. We use the dev

model for both. Since our model was trained for 2x upscaling, we pass the output latent through our model once for 2K and twice for 4K generation. No additional parameters are needed. Figure 4 demonstrates the diversity of SC-VAE generations.

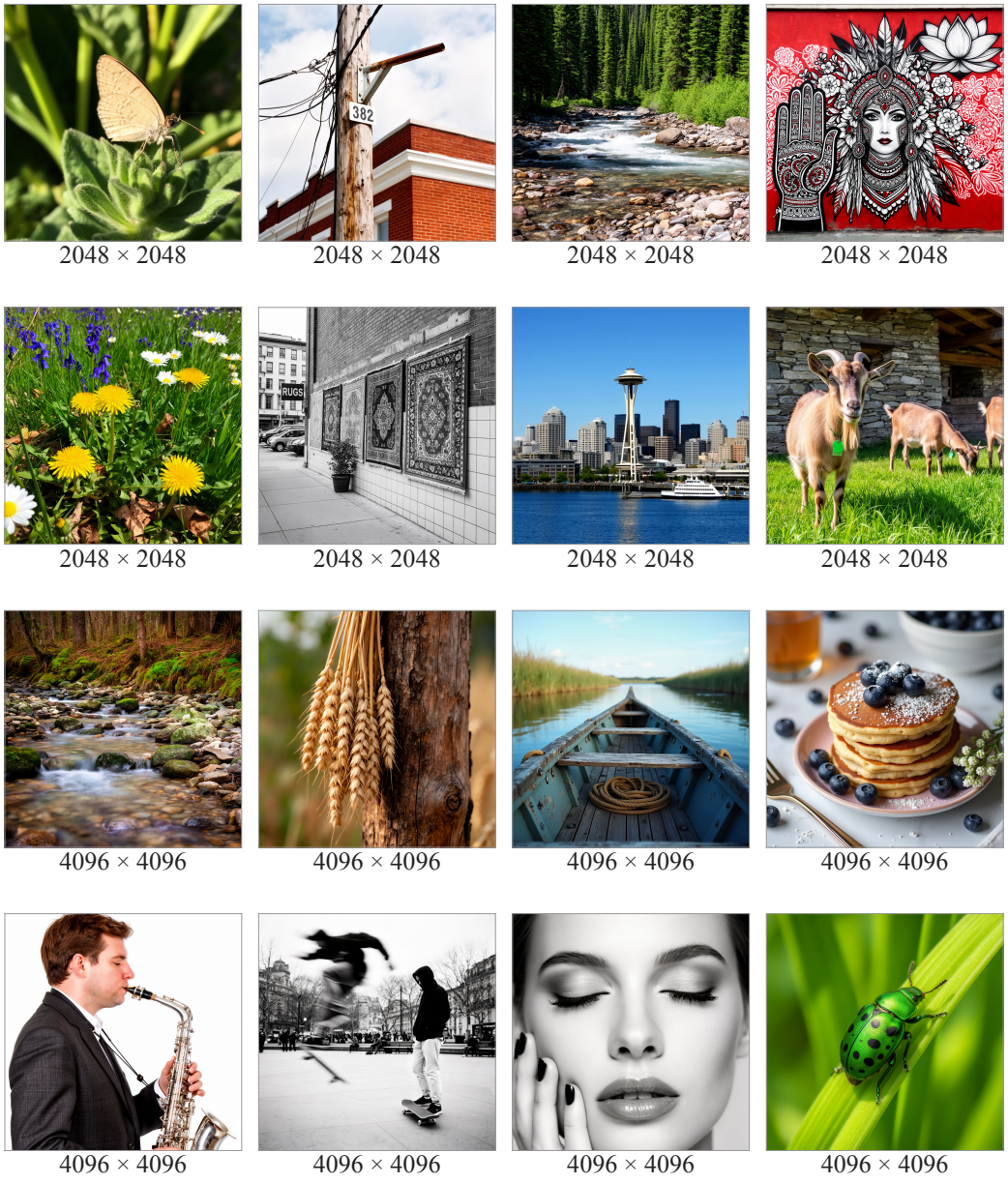



Figure 4: Diverse generations of SC-VAE at 2K/4K resolutions.

### B.3 User Study Details

Our user study provides human evaluation that helps us demonstrate the realism and text-alignment of our method. We randomly select five 4K test images per category: people, landscape, architecture, and objects. The study compares SC-VAE with three leading baselines: LSRNA, LUA and UltraFlux. In each page, the user sees a text prompt and the output images of all methods for the corresponding prompt. Users rate each image on a 1-to-5 scale for both realism and text alignment. The method order is randomized per question. Every user is required to evaluate 10 out of the 20 total chosen images. Figure 5 shows an example page from the user study.

Figure 6 shows the age groups of our participants, along with the frequency in which they use AI to generate images. Our users have varied experience with AI generation, ranging from people with no experience, to those that use generative AI daily.

Table 6 shows the study results in each category. We see that our method was preferred in all categories except *landscape* realism, in which it came a close second. We attribute the poor LSRNA results to its tiled refinement strategy that creates severe structural artifacts and poor alignment with details in the text prompt. Although UltraFlux outputs sharp images at 4K, it often suffers from outputs that look unrealistic like extra limbs or other complex compositions. This is evident in the low UltraFlux realism score on *people*, a category in which it is easier for humans to spot semantic inconsistencies. LUA has come second-best on average, but it outputs smoothed images and often suffers from a color spark artifact. We demonstrate all of these phenomena in Sections C.2 and C.3.

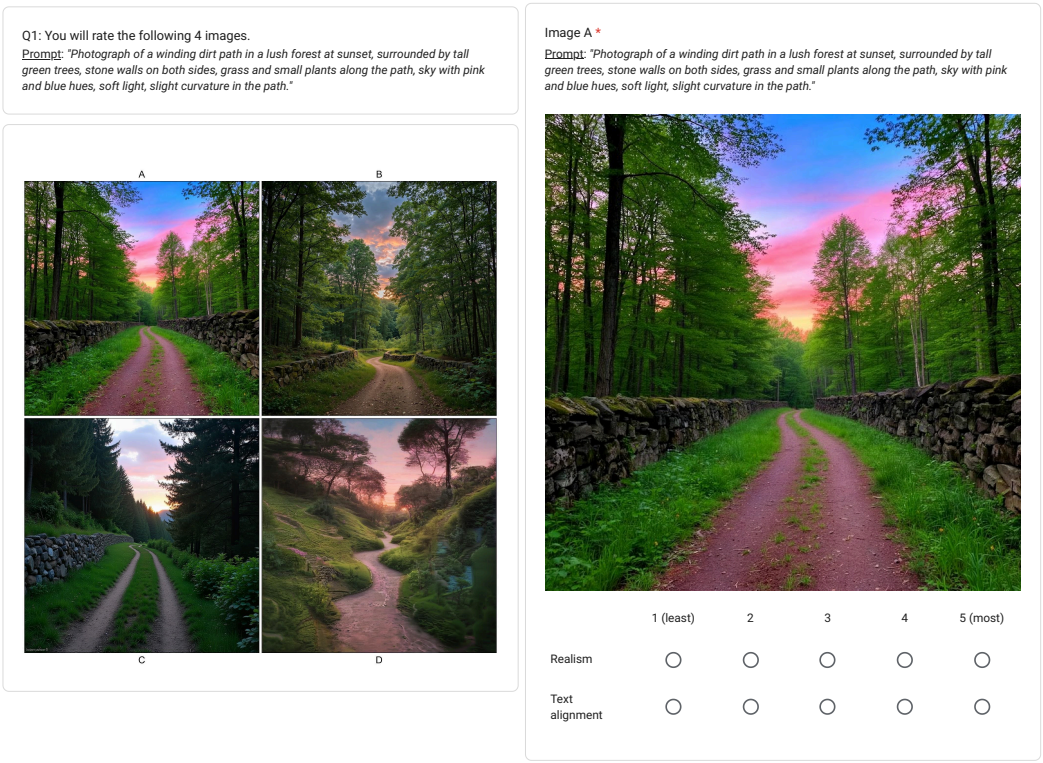



Figure 5: User study layout. **Left**: overview of the prompt and all four methods. **Right**: the user ranks each method.

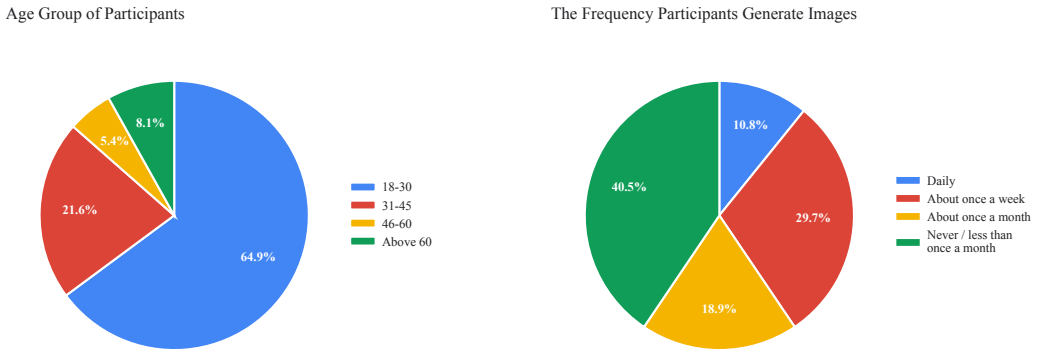



Figure 6: User study participants: age groups and image generation frequency.

Table 6: User study results by category. We report the average score for both realism and text-alignment for all the compared baselines.

| Category | Metric | LSRNA | UltraFlux | LUA | **Ours** |
|---|---|---|---|---|---|
| People | Realism | 1.96 | 2.93 | 3.47 | **4.26** |
| | Text-alignment | 2.67 | 3.46 | 3.79 | **4.06** |
| Landscape | Realism | 2.10 | 3.08 | **3.78** | 3.52 |
| | Text-alignment | 2.16 | 3.60 | 3.98 | **4.10** |
| Architecture | Realism | 1.84 | 3.67 | 3.66 | **3.86** |
| | Text-alignment | 2.26 | 3.47 | 3.95 | **4.08** |
| Objects | Realism | 2.56 | 3.48 | 3.20 | **4.03** |
| | Text-alignment | 2.18 | 3.58 | 3.88 | **4.08** |

# C Additional Experiments

## C.1 Generalization Across Architectures

We further examine our method on other models, using the same logic to train a latent upscaler for other benchmarks. Table 7 shows that our framework extends beyond FLUX models and generalizes to other models with different latent spaces, namely SD3.5. We also show that our approach works in various latent space sizes, as FLUX2 has double the number of latent channels in relation to FLUX1 and SD3.5. Figure 7 provides a qualitative demonstration of this robustness. Despite the differences in base parameter counts, native diffusion mechanics, and latent depths (16-channel vs. 32-channel representations), SC-VAE consistently learns to hallucinate photorealistic micro-textures across both SD3.5 and FLUX.2 pipelines.

Table 7: SC-VAE implementation across various architectures.

| Resolution | Model | FID ↓ | pFID ↓ | KID ↓ | pKID ↓ | CLIP ↑ | Time (s) |
|---|---|---|---|---|---|---|---|
| $2048 \times 2048$ | SC-VAE SD3.5 | 78.937 | 84.682 | 0.00349 | 0.00378 | 0.3449 | 22.12 + 1.02 |
| | SC-VAE FLUX2 | 74.353 | 81.949 | 0.00266 | 0.00323 | 0.3503 | 35.90 + 1.05 |
| $4096 \times 4096$ | SC-VAE SD3.5 | 60.678 | 85.767 | 0.00276 | 0.00910 | 0.3454 | 10.45 + 4.78 |
| | SC-VAE FLUX2 | 54.082 | 72.229 | 0.00184 | 0.00334 | 0.3505 | 35.90 + 4.81 |

## C.2 Further Investigation Of Tiled Refinement Methods

We further investigate the visual artifacts of methods that rely on localized, crop-based denoising iterations, such as Pixelsmith, DemoFusion and LSRNA. As shown in Table 1, these methods present strong patch-based scores (pFID, pKID) and prompt alignment (CLIP). In the example of DemoFusion, which is also used as the refinement technique of LSRNA, the latent image is divided into many overlapping crops that are denoised separately using the same prompt. We hypothesize that independently refining patches naturally favors patch-based metrics, while applying the same text prompt to localized regions inflates the CLIP score despite poor global structure. Figure 8 contains examples of generated images using DemoFusion, Pixelsmith and LSRNA at 2K/4K resolutions. The figure shows that global structure is damaged by the refinement process. Blending overlapping patches cause severe artifacts, such as the man with a metal fence blended with his skin or a bike rider on a tree. It also seems like non-overlapping patches create separate scenes like LSRNA's park image, where it seems like there are a few separate parks in separate parts of the image.

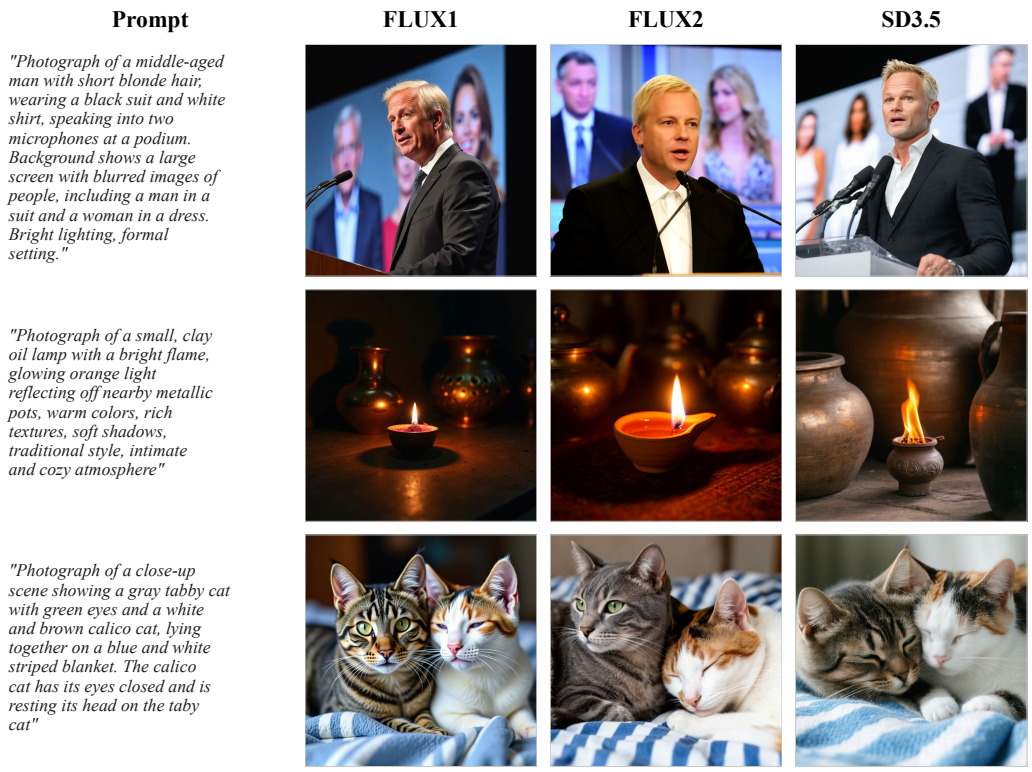



Figure 7: SC-VAE generation across several base models. Each row is generated using the same prompt.

### C.3 Further Investigation Of Competing Baselines

We further investigate other baselines visually. UltraFlux consistently produces sharp 4K images, yet it often suffers from unrealistic appearance. Figure 9 shows some examples of these failures. The top image has a distorted Ferris wheel shape, the second image has a knee joint connected to an arm, the third has many distorted limbs blended together, and the bottom image shows something that appears like a person-shaped attire floating on top of the person in the image. LUA achieves fast generation times and preserves the prompt-alignment of the base generation, yet it suffers from over-smoothing and often suffers from color spark artifacts in its output images, as shown in Figure 10.

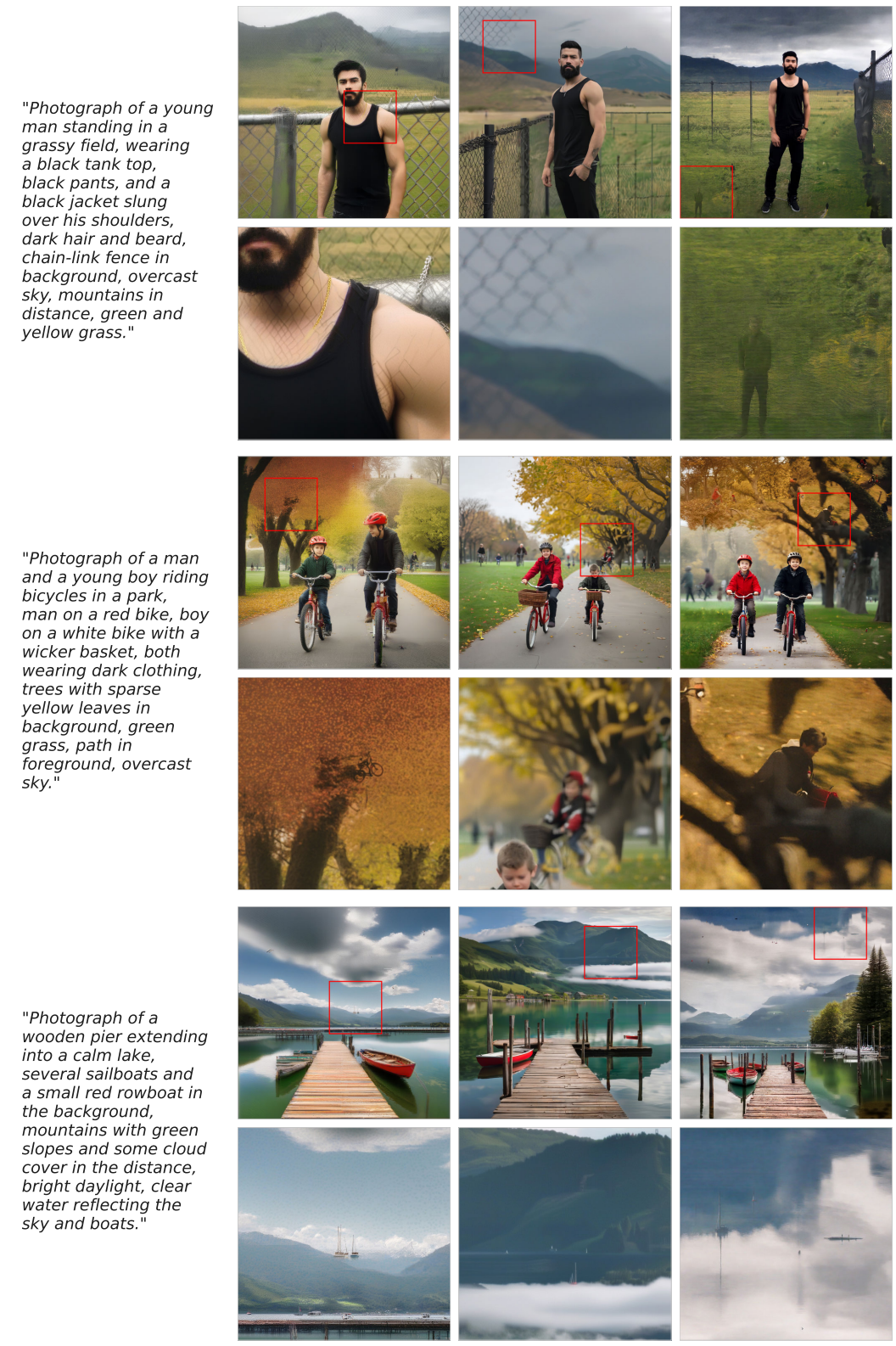



Figure 8: **Windowed-denoising reference-based methods:** structural artifacts caused by the denoising of independent crops of images in several reference-based methods.

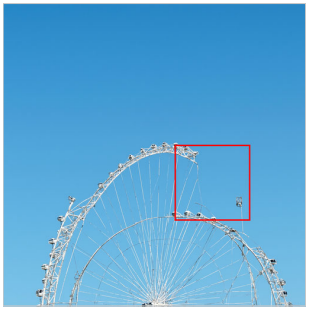
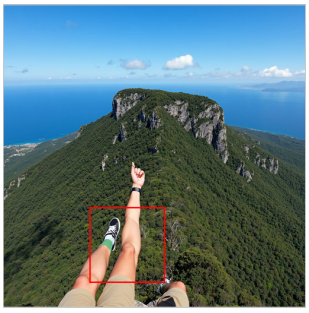
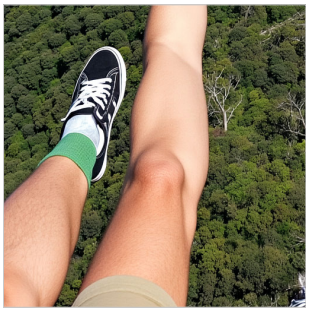
| Prompt | Original Image | Zoomed-in Detail |
|---|---|---|
| *"Photograph of a large white ferris wheel against a clear blue sky, taken from a low angle, showing the curved structure with numerous white, dome-shaped passenger cabins connected by white cables, bright sunlight, sharp shadows, modern and sleek design."* |  |  |
| *"Photograph of a person's perspective from the top of a mountain, with a raised hand pointing towards a rocky peak covered in green forest. The person is wearing khaki shorts, black and white sneakers, and green socks. Clear blue sky with a few white clouds in the background. Ocean visible on the horizon."* |  |  |
| *"Photograph of a classic orange muscle car with the hood open, showing a dark engine bay, parked in an empty lot with a clear blue sky, daytime, chrome bumper, black grille, white license plate, silver wheels, bare tree in the background, lamp post to the left, asphalt road."* |  |  |
| *"Photograph of a smiling woman in traditional Asian attire, wearing a crown, red veil, black and gold patterned skirt, brown shoes, floral garland, tattoos on arms, standing outdoors, trees in background, daytime, colorful, flowing skirt, detailed jewelry, confident pose, natural light."* |  |  |

Figure 9: Structural artifacts of the UltraFlux method.

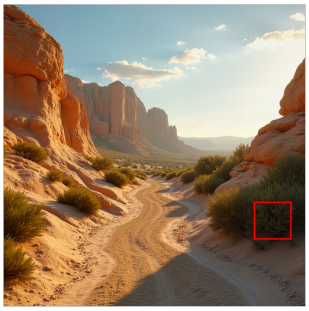
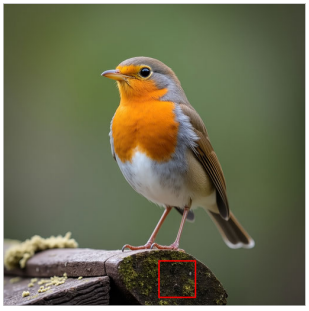
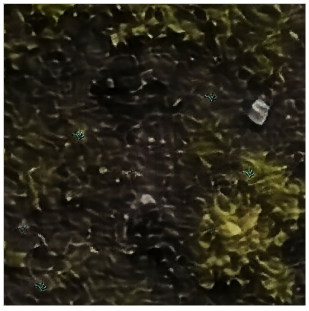
| Prompt | Original Image | Zoomed-in Detail |
|---|---|---|
| *"Black and white photograph, woman sitting on tall stool, sidewalk with brick pattern, wearing textured coat, buttoned trench coat, knee-length skirt with slit, black high-heeled ankle boots, long dark hair, relaxed expression, background with trees and wall, daytime, urban."* |  |  |
| *"Photograph of three green acorns with textured caps hanging from a tree branch, surrounded by green oak leaves, soft bokeh background, close-up, natural light, vibrant green colors, detailed textures on acorn caps."* |  |  |
| *"Photograph of a sunlit, rocky landscape with large, tan and beige erosion formations, rugged cliffs in the background, scattered green shrubs, soft orange and yellow lighting, clear blue sky with some fluffy white clouds, warm golden hour sunlight, detailed textures."* |  |  |
| *"Photograph of a small american robin perched on a mossy, weathered wooden roof, side profile with bright orange chest, grey back, and yellow beak, blurred green and gray background, natural sunlight, realistic detail, shallow depth of field, texture of wood and moss visible."* |  |  |

Figure 10: The color spark artifact of the LUA method.