TUDSR: Twice Upsampling-Diffusion for Higher Super-Resolution

Zhiqiang Wu[1] Yitong Dong[2] Xian Wei[1]*
[1]East China Normal University [2]Zhejiang University
51265902095@stu.ecnu.edu.cn

Abstract

Diffusion-based generative models have achieved remarkable success in real-world image super-resolution (SR). With tiled diffusion techniques, these models can produce high-resolution images that exceed their native-supported resolution. However, the quality of such high-resolution (e.g 2048^2*) outputs often remains extremely poor, primarily due to two factors we consider: the image upsampling ratio (e.g* $\times 8$*) exceeding the model's native-supported upsampling ratio (e.g* $\times 4$*), and the model's native-supported resolution. In practice, training a native high-resolution model requires larger architectures, which incur significant computational overhead and GPU memory costs, making it hard on limited-resource equipment. Thus, we present* ***TUDSR****, a* ***T****wice* ***U****psampling–****D****iffusion framework for higher SR. The TUDSR framework mainly consists of two stages: the first involves training at* R*-resolution, and the second introduces a looped chunk-based training strategy at* $N\,R$*-resolution. Each stage adapts a one-step GAN architecture comprising a generator and a discriminator. Based on SD2.1-base, we develop TUDSR-S, which achieves state-of-the-art performance across multiple benchmarks. Extensive experiments further demonstrate that TUDSR-S generates high-quality images at the resolutions of* 1024^2 *and even* 2048^2*, significantly outperforming existing approaches. Code is available at* https://github.com/wuer5/TUDSR.

1. Introduction

Image super-resolution (SR) [7–9, 38] is a fundamental low-level vision task that aims to reconstruct high-quality (HQ) images from their low-quality (LQ) counterparts. Unlike the traditional SR tasks, current SR tasks mainly focus on the real-world scene [4, 5, 35], where the types of degradations are complex and varying. Such a situation requires the SR model to have strong generalization ability, rather than simply fitting ability for supervised tasks. The recent emergence of large-scale diffusion models [29] has revolutionized SR, leveraging powerful generative priors [28, 31, 36, 40] to adapt to the real-world degradations.

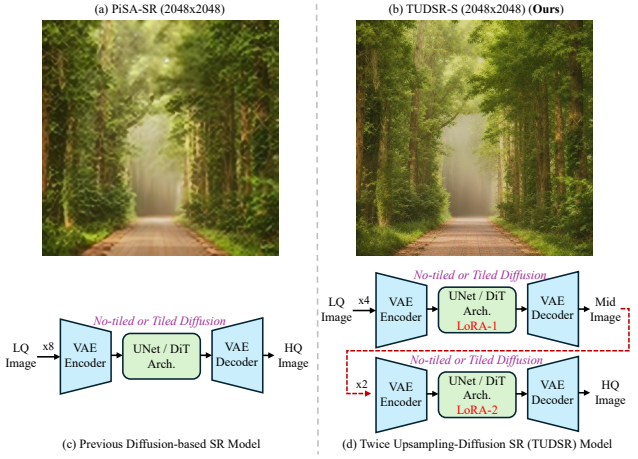


Figure 1. Comparison of (c) Previous Diffusion-based SR Model *vs*. (d) Twice Upsampling-Diffusion SR (TUDSR) Model. We present a case ($256^2 \rightarrow 2048^2$) from RealLQ250 on (a) PiSA-SR and (b) TUDSR-S (SD2.1-base). Please **zoom in**.

There are two main types of diffusion-based SR models: multi-step and one-step models. They typically use Stable Diffusion (SD) [29] as the foundational model and perform SR downstream tasks via LoRA [16] fine-tuning or ControlNet [46] methods. A common practice among these models is to operate within a native-supported resolution of 512^2 (*e.g*. [31, 36, 40]). For the images of varying resolution (height and width may be different), the tiled diffusion [26] is introduced during the inference.

However, the native resolution limit of widely used SD models (*e.g*. SD2.1-base and SD2.1), typically at 512^2 or 768^2, severely limits their applicability to high-resolution upsampling tasks such as $128^2 \rightarrow 1024^2$ or $256^2 \rightarrow 2048^2$. The required $\times 8$ upsampling ratio and the target resolution far exceed the capabilities of these SD models.

One way is to use larger generative models (*e.g*. SD3.5 [11] or FLUX.1-dev [20]) for training SR tasks with the resolution of 1024^2. Recent work, FluxSR [22], has employed FLUX.1-dev as the generative model for SR tasks.

*Corresponding author.

However, this way may result in significant computational overhead and GPU memory costs, making the higher SR difficult to train and deploy on limited devices.

For a higher SR task, it is common practice to first upsample a low-resolution image (*e.g.* 256^2) to a high-resolution image (*e.g.* 2048^2) before applying an SR model. However, direct upsampling at high ratios often exceeds the capacity of existing SR models. To address this limitation, we propose **TUDSR**, a **T**wice **U**psampling–**D**iffusion framework. Specifically, we employ two LoRA adapters trained separately within the same generative model. We divide the training process into two stages. The first stage involves training a LoRA adapter at R-resolution. In the second stage, we keep the first-stage LoRA SR model fixed and use its outputs as inputs to train the second-stage LoRA adapter. Before the second-stage training, the input is $\times N$ upsampled, then we adapt a for-loop chunked training strategy to reduce GPU memory costs. During inference, both LoRA adapters are utilized sequentially: the input is first $\times M$ upsampled and processed by the first-stage LoRA SR model; the output is then $\times N$ upsampled again and passed through the second-stage LoRA SR model to produce the final image. During inference, we only need to switch between two LoRA adapters (lower GPU memory usage) in the backbone network of a generative model, ***without*** initializing or loading the backbone network for both LoRAs. Note that $\times MN$ is your target upsampling ratio.

As shown in Fig. 1, we provide a comparison between our TUDSR framework and traditional SR models, and also present a comparison (PiSA-SR [31] *vs*. TUDSR-S) at the resolution of 2048^2. PiSA-SR produces low-quality images, which appear blurry and lack detail. On the contrary, our TUDSR-S generates forest scenes with clean, rich details, fully leveraging the SD model's image priors.

Under the TUDSR framework, we circumvent the challenges of one-step high-ratio upsampling by decomposing the process into a two-stage upsampling–diffusion pipeline, thereby fully leveraging the generative model's high-resolution generation capability.

The main contributions are as follows:

- To the best of our knowledge, our TUDSR is the first to generate high-quality, high-resolution (*e.g.* 2048^2) images from a native low-resolution (*e.g.* 512^2) generative model, making it deployable on resource-constrained devices without a larger model architecture.
- Our TUDSR framework provides a training and inference strategy for higher SR tasks. For example, TUDSR can be extended to larger-scale generative models (*e.g.* FLUX.1-dev) for higher SR (4096^2).
- Based on SD2.1-base, we instantiate TUDSR-S, achieving state-of-the-art results across multiple datasets and metrics for both $\times 4$ and $\times 8$ SR tasks. In particular, it shows the best performance on the $\times 8$ SR task.

2. Related Work

2.1. Deep Learning-Based Super-Resolution

Early SR research trained networks on bicubically downsampled image pairs, but these models failed to generalize due to complex real-world degradations. Collecting real LQ-HQ pairs is a solution, but the cost is prohibitive. Thus, simulating realistic degradation is still a key research direction. Real-ESRGAN [35] introduces a two-stage degradation process and adversarial training [12], improving perceptual quality significantly. However, these full-parameter training methods remain limited by their training data and cannot fully address diverse real-world degradations.

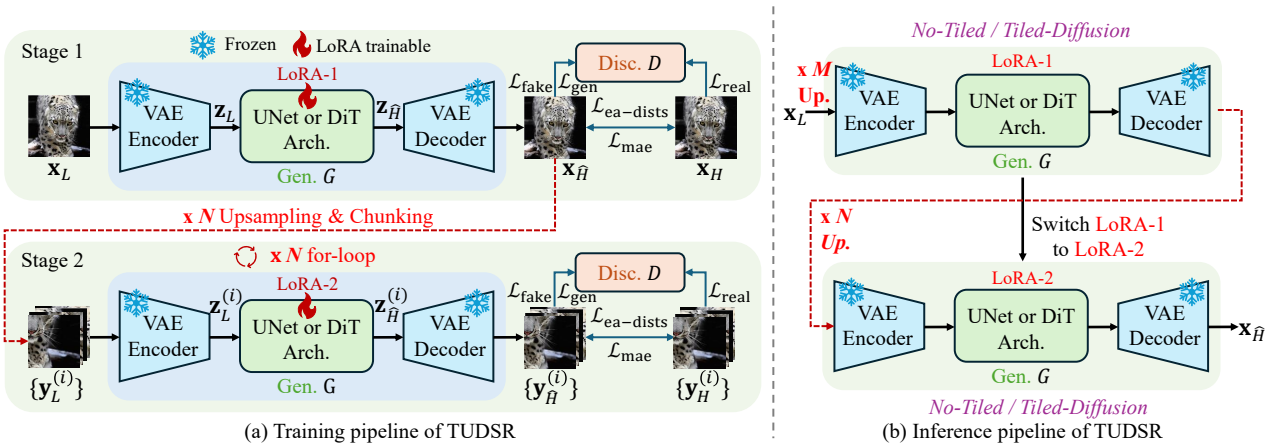
2.2. Diffusion-Based Super-Resolution

Early diffusion-based SR methods often oversimplified real-world degradations and overlooked the value of powerful image priors. With the rise of large-scale diffusion models like SD [29], recent works leverage their strong pretrained priors for SR. For example, StableSR [34] conditions on the LR input; DiffBIR [24] uses a two-stage degradation removal and enhancement pipeline; and SeeSR [41] employs a degradation-aware prompt extractor. These approaches typically rely on standard DDPM [15] training and multi-step sampling, resulting in two key drawbacks: slow inference and potential low fidelity or unrealistic outputs, limiting their suitability for fidelity-critical SR.

To address these issues, recent one-step models have emerged. Methods such as SinSR [36] and OSEDiff [40] use distillation to compress multi-step models into one-step models. SinSR introduces a consistency-preserving loss to shorten the diffusion trajectory, while OSEDiff employs Variational Score Distillation (VSD) to distill the generalization capability of Stable Diffusion. PiSA-SR [31] learns two LoRA modules for an SD model to achieve improved and adjustable SR at both pixel and semantic levels. InvSR [45] designs a Partial Noise Prediction strategy to construct an intermediate state of the SD model as the sampling start point, enabling either multi-step or one-step prediction by configuring the number of intermediate timesteps.

2.3. Higher Super-Resolution

Although these diffusion-based models can generate high-resolution images via tiled diffusion [26], they are inherently constrained by their SD model, which typically supports resolutions like 512^2 or 768^2. For high-resolution (*e.g.* 2048^2) outputs requiring large upscaling factors (*e.g.* $\times 8$), these models (*e.g.* [24, 31, 34, 36, 40, 41, 44, 45]) struggle to produce high-quality results. While scaling to larger generative models (*e.g.* FLUX.1-dev [20]) is possible, the associated training computational cost would be immense. Thus, this paper will focus specifically on the upscaling factor to construct a higher SR framework.



(a) Training pipeline of TUDSR (b) Inference pipeline of TUDSR

Figure 2. Illustration of the (a) training and (b) inference pipelines of TUDSR. In stage 1, we train a one-step LoRA at R-resolution. In stage 2, we freeze the first-stage LoRA SR model to generate an intermediate image, which is then $\times N$ upsampled. Subsequently, we train a second-stage LoRA to process the upsampled image using a for-loop chunk-wise training strategy, where gradients are backpropagated per chunk. Notably, both stages employ a one-step training approach based on a GAN architecture [12]. The inference process employs the two LoRA adapters in sequence: the input first undergoes $\times M$ upsampling by the first-stage LoRA SR model, followed by $\times N$ upsampling by the second-stage model, yielding the final output.

3. Methodology

3.1. Diffusion-based SR Modeling

Given an LQ image $\mathbf{x}_L$ with complex and varying real-world degradations, we aim to generate the corresponding HQ image $\mathbf{x}_H$. Our task mainly focuses on the SD model, which consists of a VAE encoder [19] $\mathcal{E}$ for compressing images into latent space, a backbone for denosing $\mathcal{B}$, and a VAE decoder [19] $\mathcal{D}$ for constructing the latent representation into the image. The problem is formulated as

$$\mathbf{x}_{\widehat{H}} = \mathcal{D}(\mathbf{z}_{\widehat{H}}),\ \mathbf{z}_{\widehat{H}} = \mathrm{f}_G(\mathbf{z}_L),\ \mathbf{z}_L = \mathcal{E}(\mathbf{x}_L), \tag{1}$$

where f_G denotes the denoising process.

3.2. Architeture of TUDSR

We propose TUDSR, a Twice Upsampling-Diffusion SR framework, as illustrated in Fig. 2. TUDSR use twp-stage training, where each stage employs a GAN [12] architecture (Generator G and Discriminator D).

- **Generator** G**:** We employ a pre-trained generative model (training via LoRA) as the generator.
- **Discriminator** D**:** We employ a pre-trained DINOv3-ViT-B [30] model as the feature extractor within a multi-level discriminator, which incorporates multi-level discriminator heads for adversarial learning [12].

3.3. Rethinkings of GAN Architeture

In traditional GAN training, the difficulties primarily stem from two factors: (1) the common training paradigm of mapping noise to HQ images involves high task complexity and diversity; (2) training both the generator and discriminator from scratch makes it challenging to maintain a balanced optimization dynamic. However, in GANs for SR, the task is simplified to mapping from LQ to HQ images. Furthermore, since designing and training both a discriminator and a generator from scratch is complex and prone to imbalance, we adapt a pre-trained diffusion model as the generator and fine-tune it using LoRA. On the other hand, we leverage a pre-trained DINOv3-ViT-B [30] to extract multi-level features and train multi-level discriminator heads from scratch. As a result, both the generator and the discriminator are equipped with strong prior knowledge, while the number of trainable parameters remains relatively small. This leads to significantly enhanced stability in GAN-based SR training. Moreover, the GAN framework enables the generation of higher-quality images with superior detail rendition compared to other architectures.

3.4. GAN Generator G

In our TUDSR framework, we see a pre-trained generative model (*e.g.* SD2.1-base) with LoRA fine-tuning as the generator. Given a LQ image $\mathbf{x}_L$, we define the generator as the function G

$$G(\mathbf{x}_L, t, c) = \mathbf{x}_{\widehat{H}},\ \text{where} \begin{cases} \mathbf{z}_L = \mathcal{E}(\mathbf{x}_L) \\ \mathbf{z}_{\widehat{H}} = \frac{\mathbf{z}_L - \sqrt{1-\bar{\alpha}_t}\cdot\mathcal{B}(\mathbf{z}_L, t, c)}{\sqrt{\bar{\alpha}_t}} \\ \mathbf{z}_{\widehat{H}} = \mathcal{D}(\mathbf{x}_{\widehat{H}}) \end{cases} \tag{2}$$

Note that t is the fixed timestep for onestep denoising and c is the prompt condition.

3.5. GAN Discriminator D

The GAN discriminator consists of a frozen pre-trained backbone DINOv3-ViT-B for feature extraction and fully

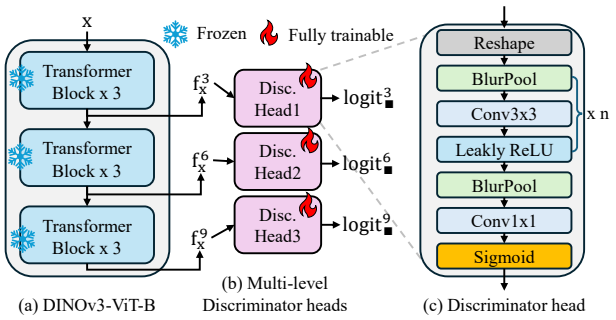


Figure 3. Illustration of Discriminator D (DINOv3-ViT-B + Multi-level Discriminator Heads). Note that BlurPool [47] is a low-pass filter used for anti-aliasing, which is a commonly used method in the design of GAN discriminators.

trainable multi-level discriminator heads to predict the discrimination logits.

3.5.1. DINOv3-ViT-B Features

The DINOv3-ViT-B (86M parameters) adapts the standard ViT [10] architecture for pre-training on large-scale datasets, which contain powerful image priors. It is mainly composed of 12 layers of transform blocks, and a large body of literature [2, 6, 13, 42] shows that the shallow and middle layers contain detailed information, while the higher layers primarily encode global semantic information. For SR tasks, LQ images themselves contain global semantic information, but lack detailed information. Therefore, we use the features in 3, 6, and 9 layers for detailed discrimination here. Given the input image $\mathbf{x}$, we define the extraction process of DINOv3-ViT-B as

$$\text{DINOv3ViTB}(\mathbf{x}, \{3, 6, 9\}) = \{\text{f}^3_{\mathbf{x}}, \text{f}^6_{\mathbf{x}}, \text{f}^9_{\mathbf{x}}\}. \quad (3)$$

3.5.2. Multi-level Discriminator Heads

Once we obtain the features from DINOv3-ViT-B, we design multi-level discriminator heads DHead^l for discrimination prediction, as shown in Fig. 3. We define the discriminator as the function D where

$$D(\mathbf{x}) = \{\text{DHead}^l(\text{f}^l_{\mathbf{x}})\} = \{\text{logits}^l_{\square}\},\ l = 3, 6, 9. \quad (4)$$

3.6. Training Objective

3.6.1. Structural Perception Loss

Although LPIPS [48] has been widely adopted for structural perception, several studies and our own experiments show that it can introduce artifacts, particularly during diffusion-based GAN training. To address this, we employ dists as our structural perception loss. Unlike LPIPS, which computes a weighted L2 distance between multi-level feature maps from a pre-trained network, dists measures the discrepancy in first-order (mean) and second-order (covariance) statistics captured by these feature maps. This approach exhibits greater robustness to geometric distortions and luminance variations, reduces the likelihood of artifacts, and aligns more closely with human visual perception. Furthermore, as demonstrated in [21], edge details are also crucial for structural perception. Following [21], we integrate an edge-aware version of dists to enhance performance:

$$\mathcal{L}_{\text{ea-dists}} = \text{dists}(\mathbf{x}_H, \mathbf{x}_{\widehat{H}}) + \text{dists}(\mathcal{S}(\mathbf{x}_H), \mathcal{S}(\mathbf{x}_{\widehat{H}})), \quad (5)$$

where $\mathcal{S}(\cdot)$ denotes the Sobel operator to extract the edge information of images.

3.6.2. GAN Generator Loss

The GAN generator loss is used to update the generator's parameters. We freeze the discriminator D to obtain the generative logits: $D(\mathbf{x}_{\widehat{H}}) = \{\text{logit}^l_{\text{gen}}\}$ and employ the Binary Cross-Entropy (BCE):

$$\mathcal{L}_{\text{gen}} = \frac{1}{3}\sum_{l}^{\{3,6,9\}} \text{BCE}(\text{label}_{\text{gen}}, \text{logit}^l_{\text{gen}}), \quad (6)$$

where $\text{label}_{\text{gen}}$ is the soft label and set to 0.8.

3.6.3. GAN Discriminator Loss

The GAN generator loss is used to discriminate between real and fake images to update the discriminator's parameters. We detach $\mathbf{x}_{\widehat{H}}$ and obtain the fake logits: $D(\text{Detach}(\mathbf{x}_{\widehat{H}})) = \{\text{logit}^l_{\text{fake}}\}$ and real logits: $D(\mathbf{x}_H) = \{\text{logit}^l_{\text{real}}\}$. We also use the Binary Cross-Entropy (BCE):

$$\begin{aligned} \mathcal{L}_{\text{real}} &= \frac{1}{3}\sum_{l}^{\{3,6,9\}} \text{BCE}(\text{label}_{\text{real}}, \text{logit}^l_{\text{real}}), \\ \mathcal{L}_{\text{fake}} &= \frac{1}{3}\sum_{l}^{\{3,6,9\}} \text{BCE}(\text{label}_{\text{fake}}, \text{logit}^l_{\text{fake}}), \end{aligned} \quad (7)$$

where $\text{label}_{\text{fake}}$ and $\text{label}_{\text{real}}$ are set to 0 and 0.8.

3.6.4. Total Training Objective

For the generator, we optimize the LoRA parameters of the generative model via the loss:

$$\mathcal{L}_{\text{G}} = \lambda_1\mathcal{L}_{\text{ea-dists}} + \lambda_2\mathcal{L}_{\text{gen}} + \lambda_3\mathcal{L}_{\text{mae}}, \quad (8)$$

where $\lambda_1 = 5$, $\lambda_2 = 0.5$, and $\lambda_3 = 0.5$ in this paper.

For the discriminator, we optimize all the parameters of the multi-level discriminator heads via the loss:

$$\mathcal{L}_{\text{D}} = \lambda_2(\mathcal{L}_{\text{real}} + \mathcal{L}_{\text{fake}}). \quad (9)$$

3.7. Stage 1: R-resolution Training

In the first stage, we train a LoRA of the generative model as R-resolution (*e.g.* 512). We optimize the LoRA parameters θ_1 of a generative model and the parameters ϕ_1 of the multi-level discriminator heads. Given the LQ image $\mathbf{x}_L$ with the resolution of R^2, the stage-1 training is described in Algorithm 1.

Algorithm 1 Stage 1 Training

Require: $\mathbf{x}_L, \mathbf{x}_H, t_1, c, G_{\theta_1}, D_{\phi_1}$
$\mathbf{x}_{\widehat{H}} \leftarrow G_{\theta_1}(\mathbf{x}_L, t_1, c)$
$\{\text{logit}^l_{\text{gen}}\} \leftarrow D^*_{\phi_1}(\mathbf{x}_{\widehat{H}})$ ▷ *: freeze D_{ϕ_1}, for G
$\mathcal{L}^{\theta_1}_{\text{G}} \leftarrow \lambda_1 \mathcal{L}_{\text{ea-dists}}(\mathbf{x}_{\widehat{H}}, \mathbf{x}_H) + \lambda_2 \mathcal{L}_{\text{gen}}(\{\text{logit}^l_{\text{gen}}\}) + \lambda_3 \mathcal{L}_{\text{mae}}(\mathbf{x}_{\widehat{H}}, \mathbf{x}_H)$
$\theta_1 \leftarrow \text{Backward}(\mathcal{L}^{\theta_1}_{\text{G}})$ ▷ Update θ_1
$\{\text{logit}^l_{\text{fake}}\} \leftarrow D_{\phi_1}(\text{Detach}(\mathbf{x}_{\widehat{H}}))$ ▷ For real image
$\{\text{logit}^l_{\text{real}}\} \leftarrow D_{\phi_1}(\mathbf{x}_H)$ ▷ For fake image
$\mathcal{L}^{\phi_1}_{\text{D}} \leftarrow \lambda_2(\mathcal{L}_{\text{fake}}(\{\text{logit}^l_{\text{fake}}\}) + \mathcal{L}_{\text{real}}(\{\text{logit}^l_{\text{real}}\}))$
$\phi_1 \leftarrow \text{Backward}(\mathcal{L}^{\phi_1}_{\text{D}})$ ▷ Update ϕ_1

3.8. Stage 2: NR-resolution Training

In the second stage, we freeze the LoRA of the first stage. Given the LQ image $\mathbf{x}_L$ with the resolution of R^2, we freeze G_{θ_1} and obtain the intermediate image $\mathbf{m} = G_{\theta_1}(\mathbf{x}_L, t_1, c)$. Then, we use the bilinear interpolation to $\times N$ upsample $\mathbf{m}$:

$$\mathbf{y}_L = \text{Upsampling}(\mathbf{m}, N). \tag{10}$$

However, directly training the second LoRA at NR-resolution introduces two main issues: (1) NR-resolution exceeds the model's native-supported resolution (*i.e.* R); (2) gradient computation at NR-resolution consumes substantial GPU memory. We adapt a for-loop chunked training strategy and perform gradient updates per chunk:

$$\{\mathbf{y}^{(i)}_L\} = \text{Chunking}(\mathbf{y}_L, R),\ i \in \{1, \cdots, N^2\}. \tag{11}$$

Note that Chunking divides $\mathbf{y}_L$ into chunks from left to right and from top to bottom without overlap.

Given the HQ chunked image $\{\mathbf{y}^{(i)}_H\}$, we optimize the LoRA parameter θ_2 of the generative model and the multi-level discriminator head parameters ϕ_2. The stage-2 training is described in Algorithm 2.

Algorithm 2 Stage 2 Training

Require: $\mathbf{x}_L, \mathbf{y}^{(i)}_H, t_1, t_2, c, G_{\theta_1}, G_{\theta_2}, D_{\phi_2}, R, l, N$
$\mathbf{m} \leftarrow G^*_{\theta_1}(\mathbf{x}_L, t_1, c)$ ▷ *: freeze G_{θ_1}
$\mathbf{y}_L \leftarrow \text{Upsampling}(\mathbf{m}, N)$ ▷ $\times N$ upsampling
$\{\mathbf{y}^{(i)}_L\} \leftarrow \text{Chunking}(\mathbf{y}_L, R),\ i \in \{1, \cdots, N^2\}$
for $i = 1$ to N^2 **do** ▷ For each chunk
  $\mathbf{y}^{(i)}_{\widehat{H}} = G_{\theta_2}(\mathbf{y}^{(i)}_L, t_2, c)$
  $\{\text{logit}^l_{\text{gen}}\} \leftarrow D^*_{\phi_2}(\mathbf{y}^{(i)}_{\widehat{H}})$ ▷ *: freeze D_{ϕ_2}, for G
  $\mathcal{L}^{\theta_1}_{\text{G}} \leftarrow \lambda_1 \mathcal{L}_{\text{ea-dists}}(\mathbf{y}^{(i)}_{\widehat{H}}, \mathbf{y}^{(i)}_H) + \lambda_2 \mathcal{L}_{\text{gen}}(\{\text{logit}^l_{\text{gen}}\}) + \lambda_3 \mathcal{L}_{\text{mae}}(\mathbf{y}^{(i)}_{\widehat{H}}, \mathbf{y}^{(i)}_H)$
  $\theta_2 \leftarrow \text{Backward}(\mathcal{L}^{\theta_2}_{\text{G}})$ ▷ Update θ_2
  $\{\text{logit}^l_{\text{fake}}\} \leftarrow D_{\phi_2}(\text{Detach}(\mathbf{y}^{(i)}_{\widehat{H}}))$ ▷ For real image
  $\{\text{logit}^l_{\text{real}}\} \leftarrow D_{\phi_2}(\mathbf{y}^{(i)}_H)$ ▷ For fake image
  $\mathcal{L}^{\phi_2}_{\text{D}} \leftarrow \lambda_2(\mathcal{L}_{\text{fake}}(\{\text{logit}^l_{\text{fake}}\}) + \mathcal{L}_{\text{real}}(\{\text{logit}^l_{\text{real}}\}))$
  $\phi_2 \leftarrow \text{Backward}(\mathcal{L}^{\phi_2}_{\text{D}})$ ▷ Update ϕ_2
end for

4. Experiment

4.1. Experimental Settings

Training settings. We adopt the experimental setup from SeeSR [41], using the LSDIR [23] dataset along with the first 10,000 facial images from FFHQ [17]. LQ-HQ pairs are synthesized via the standard Real-ESRGAN degradation pipeline [35]. Within TUDSR, we initialize TUDSR-S from SD2.1-base, setting the rank of UNet LoRA to 32 for both stages. Training uses AdamW [25] with a learning rate of 5×10^{-5}, batch size of 1, and 4 gradient accumulation steps. We set $N = 2$ in Eq. (10), $t_1 = 200$ in Algorithm 1, and $t_2 = 50$ in Algorithm 2. Stage 1 and stage 2 are trained for 5,100 and 3,500 steps, respectively, on 4 RTX 4090 GPUs.

Test Datasets. We evaluate on four real-world datasets: RealSR [3], DrealSR [39], RealLQ250 [1], and RealLR200 [41]. RealSR and DrealSR contain 100 and 93 images at 128^2 with corresponding GT at 512^2, respectively; RealLQ250 consists of 250 256^2 images; RealLR200 includes 200 images of varying resolutions.

Test Metrics. As PSNR and SSIM [37] no longer accurately reflect perceptual quality, we adopt model-based metrics: reference metrics LPIPS [48] and FID [14]; non-reference metrics CLIPIQA [33], CLIPIQA+ [33], NIMA [32], NIQE [27], LIQE [49], MUSIQ [18], and MANIQA [43].

4.2. Comparison Results with Others

We compare TUDSR-S against state-of-the-art diffusion-based multi-step (StableSR [34], DiffBIR [24], SeeSR [41], ResShift [44]) and one-step SR models (SinSR [36], OSEDiff [40], PiSA-SR [31], InvSR [45]) on $\times 4$ and $\times 8$ tasks. Tiled diffusion is used for inputs exceeding 512 pixels. $\times 8$ experiments on RealLR200 are omitted due to OOM errors. Given the high inference cost of multi-step models (over 10 minutes per $\times 8$ image), we restrict $\times 8$ comparisons to one-step models only.

4.2.1. Quantitative Comparisons

Table 1 presents the comprehensive quantitative comparisons ($\times 4$) on four test datasets. Our proposed TUDSR-S demonstrates state-of-the-art performance across multiple real-world SR benchmarks, which achieves overwhelming results on key perceptual quality metrics, including CLIP-IQA, CLIPIQA+, LIQE, MUSIQ, and MANIQA. Such metrics consistently reflect the quality of images generated by TUDSR-S from various perspectives.

Table 1. Quantitative comparisons ($\times 4$) with state-of-the-art multi-step and one-step models on four real-world benchmark datasets. Note that TUDSR-S denotes `M2N2` twice upsampling-diffusion in the table.

Datasets	Metrics	RealESRGAN [35]	StableSR [34]	SeeSR [41]	DiffBIR [24]	ResShift [44]	OSEDiff [40]	PiSASR [31]	InvSR [45]	SinSR [36]	TUDSR-S
RealSR	LPIPS↓	0.2710	**0.2604**	0.3007	0.3470	0.3159	0.2921	0.2672	0.2871	0.3210	0.3217
	FID↓	135.15	132.09	125.51	134.59	149.65	123.50	124.19	138.85	136.78	**111.42**
	CLIPIQA↑	0.4490	0.5426	0.6699	**0.6960**	0.5505	0.6693	0.6699	0.6785	0.6156	0.6846
	CLIPIQA+↑	0.5841	0.6150	0.6909	0.6989	0.5451	0.6964	0.6957	0.6880	0.5370	**0.7135**
	NIMA↑	4.6551	4.6767	4.9191	4.9159	4.7554	4.8951	4.8953	**5.0946**	4.6643	4.8676
	NIQE↓	5.7960	6.6231	5.3984	5.4992	6.8833	5.6474	5.5057	5.6222	6.2998	**4.7149**
	LIQE↑	3.3571	3.2578	4.1354	4.0261	3.1859	4.0690	4.0989	4.0392	3.1466	**4.3738**
	MUSIQ↑	60.3657	61.8058	69.8165	68.3462	60.2181	69.0896	70.1492	68.5372	60.4204	**70.2406**
	MANIQA↑	0.5492	0.5952	0.6445	0.6540	0.5388	0.6331	0.6552	0.6628	0.5391	**0.6786**
DrealSR	LPIPS↓	0.2819	**0.2698**	0.3174	0.4520	0.3526	0.2968	0.2960	0.3538	0.3674	0.3372
	FID↓	147.80	151.27	147.53	177.06	176.70	135.29	**130.43**	171.40	171.88	133.74
	CLIPIQA↑	0.4517	0.4907	0.6913	0.6859	0.5410	0.6963	0.6974	**0.7132**	0.6348	0.6944
	CLIPIQA+↑	0.5544	0.5347	0.6794	0.6828	0.5157	0.6825	0.6920	0.6832	0.5400	**0.6921**
	NIMA↑	4.3261	4.2136	4.6945	4.7847	4.4405	4.6766	4.6250	**4.8566**	4.4639	4.7272
	NIQE↓	6.6927	7.5488	6.4136	6.2409	7.8693	6.4904	6.1759	5.9917	7.1422	**5.5427**
	LIQE↑	2.9259	2.4349	4.1268	3.8930	2.7968	3.9371	4.0440	4.0557	3.0514	**4.1769**
	MUSIQ↑	54.2721	51.3635	65.0935	65.6585	52.3726	64.6537	66.1094	65.9956	54.9825	**66.3650**
	MANIQA↑	0.4899	0.4969	0.6043	0.6279	0.4750	0.5895	0.6146	0.6302	0.4855	**0.6312**
RealLQ250	CLIPIQA↑	0.5434	0.5150	0.7132	0.7255	0.4734	0.6995	0.7095	0.6628	0.6998	**0.7315**
	CLIPIQA+↑	0.6117	0.5811	0.7142	0.7213	0.4642	0.7017	0.7160	0.6722	0.5919	**0.7297**
	NIMA↑	5.2554	5.0700	5.3863	**5.4922**	5.0243	5.2364	5.2429	5.4401	5.1938	5.3268
	NIQE↓	4.1292	4.6345	3.9832	**3.5608**	4.8476	3.9127	3.8751	4.4098	5.7974	3.9440
	LIQE↑	3.3410	2.7533	4.1336	4.0874	2.4609	3.8610	3.9813	3.7113	3.2465	**4.3017**
	MUSIQ↑	62.5169	57.1341	71.1218	70.3687	57.7724	69.6786	71.0710	65.8212	63.8548	**71.9250**
	MANIQA↑	0.5288	0.5203	0.6204	0.6232	0.4657	0.5928	0.6157	0.5914	0.5178	**0.6329**
RealLQ200	CLIPIQA↑	0.5409	0.5731	0.6959	0.7222	0.4942	0.7008	0.7125	0.6830	0.6615	**0.7323**
	CLIPIQA+↑	0.6222	0.6360	0.7171	0.7241	0.5094	0.7136	0.7292	0.7111	0.5888	**0.7388**
	NIMA↑	5.1866	5.2607	5.4154	5.4648	5.0419	5.3530	5.3920	**5.5045**	5.1763	5.3212
	NIQE↓	4.1796	4.3515	3.9996	**3.7803**	4.8432	3.9268	3.9991	3.9936	5.3042	4.0156
	LIQE↑	3.4836	3.4319	4.1806	4.0541	2.6938	3.9967	4.1690	4.0778	3.2175	**4.3648**
	MUSIQ↑	62.9605	63.3346	70.2502	68.7120	57.4580	69.5654	70.8834	68.9061	61.3634	**71.3676**
	MANIQA↑	0.5553	0.5749	0.6360	0.6385	0.4925	0.6143	0.6418	0.6481	0.5343	**0.6545**

Figure 4. Qualitative comparisons ($\times 4$ *i.e.* $256^2 \rightarrow 1024^2$) with state-of-the-art multi-step and one-step models. Please **zoom in**.

Table 2 also presents the quantitative comparisons ($\times 8$) to demonstrate the quality of our generation on high-resolution (*i.e.* 1024^2 and 2048^2) images. The one-step models exhibit a substantial decline across most evaluation metrics on the $\times 8$ SR tests conducted on the RealSR, DRealSR, and RealLQ250 datasets, compared to Tab. 1. This indicates that the $\times 8$ upsampling task exceeds the capacity of these one-step approaches. In contrast, TUDSR-S achieves comprehensive performance across most metrics across the three datasets. The results in both Tab. 1 and Tab. 2 demonstrate that our method not only delivers strong performance in the conventional $\times 4$ setting but also excels in the more challenging $\times 8$ SR scenario, showing the effectiveness of our twice upsampling–diffusion strategy.

Figure 5. Qualitative comparisons ($\times 8$ *i.e.* $256^2 \rightarrow 2048^2$) with state-of-the-art one-step models. Please **zoom in**.

Table 2. Quantitative comparisons ($\times 8$) with state-of-the-art one-step models on four real-world benchmark datasets. Note that TUDSR-S denotes `M4N2` twice upsampling-diffusion in the table.

Datasets	Methods	C-IQA↑	C-IQA+↑	NIMA↑	NIQE↓	LIQE↑	MUSIQ↑	MANIQA↑
RealSR	RealESRGAN	0.5061	0.5833	4.9421	5.3701	3.2344	57.6718	0.4858
	OSEDiff	0.6976	0.6673	**5.0711**	5.6951	3.6347	**67.5970**	0.5678
	PiSA-SR	0.6642	0.6562	4.8645	5.0937	3.2765	66.0150	0.5510
	InvSR	0.6581	0.6420	4.9595	**4.3930**	3.0830	64.3007	0.5711
	SinSR	0.6175	0.5151	4.6538	7.2585	2.4228	53.4213	0.4605
	TUDSR-S	**0.6920**	**0.6883**	4.8305	4.6839	**3.6547**	67.2225	**0.6126**
DrealSR	RealESRGAN	0.4642	0.5073	4.5156	6.8732	2.4309	47.5704	0.4226
	OSEDiff	0.6774	0.6265	4.6819	6.1817	2.8764	58.5390	0.5176
	PiSA-SR	0.6572	0.6084	4.5354	6.2137	2.7529	56.7244	0.4960
	InvSR	0.5621	0.5640	4.6442	**5.2770**	2.6230	54.4117	0.5028
	SinSR	0.6118	0.5051	4.4498	7.8526	2.3197	48.7678	0.4236
	TUDSR-S	**0.7186**	**0.6680**	**4.7217**	5.5417	**3.6124**	**63.9383**	**0.5758**
RealLQ250	RealESRGAN	0.5370	0.5629	4.9598	4.6685	2.7754	48.7098	0.4549
	OSEDiff	0.6793	0.6288	4.7853	4.6349	2.7375	54.8024	0.5171
	PiSA-SR	0.6355	0.6110	4.7429	4.7085	2.4839	49.8787	0.4801
	InvSR	0.5233	0.5057	4.8383	5.5685	2.2872	40.2874	0.4358
	SinSR	0.6222	0.4930	4.7463	5.7477	1.9833	40.1009	0.4362
	TUDSR-S	**0.7222**	**0.6904**	**5.3103**	**3.9251**	**3.6112**	**63.2129**	**0.5847**

4.2.2. Qualitative Comparisons

We present qualitative comparisons for $\times 4$ super-resolution in Fig. 4. In the first case (top row), our TUDSR-S produces the best quality for characters and numbers, followed by OSEDiff, InvSR, and PiSA-SR, while the other models perform relatively poorly. In the second case, our method achieves the highest quality in generating teeth and beard details, with the beard appearance being particularly realistic. OSEDiff, PiSA-SR, and SeeSR show lower realism and clarity than our model, while the remaining models exhibit significant quality degradation (especially SinSR and ResShift). In the third case, TUDSR-S generates the most realistic starry sky, complete with fine star details. DiffBIR also produces reasonable results, though inferior to ours, and other models perform noticeably worse.

Table 3. Inference time (seconds per image). All inference times are tested on an H800 GPU.

SR task	SinSR	OSEDiff	InvSR	PiSA-SR	TUDSR-S
$\times 4$ ($128^2 \rightarrow 512^2$)	0.0728	0.0892	0.0838	0.0789	**0.0596**
$\times 8$ ($128^2 \rightarrow 1024^2$)	0.5674	0.6532	**0.3693**	0.5601	0.4201
$\times 4$ ($256^2 \rightarrow 1024^2$)	0.5889	0.6776	**0.3723**	0.5776	0.4296
$\times 8$ ($256^2 \rightarrow 2048^2$)	3.2941	3.9241	**1.7844**	2.8337	2.2848

We further provide qualitative comparisons for $\times 8$ SR in Fig. 5. These examples demonstrate the strong performance of our TUDSR-S in high-resolution generation. In the first case (top row), our approach produces a facial image closest to the real person, with rich details and a naturally rendered bokeh background of the vehicle. Real-ESRGAN, SinSR, and InvSR yield the poorest results with missing details, while OSEDiff and PiSA-SR generate faces with inconsistent identity. The second case highlights TUDSR-S's superiority in fine-detail generation, such as roof eaves, where our model surpasses all others. Competing models produce blurry, oversmoothed results in these regions.

These comparisons fully demonstrate that TUDSR-S achieves strong performance on both standard $\times 4$ and challenging $\times 8$ SR tasks, validating the effectiveness of our twice upsampling-diffusion strategy for generating high-fidelity details. We consider that the two diffusion processes can fully utilize the powerful prior of SD.

4.2.3. Inference Time Comparisons

We also provide the inference time in Sec. 4.2.3. On the low-resolution task ($128^2 \rightarrow 512^2$), TUDSR-S exhibits the

fastest inference speed (0.0596 s), significantly outperforming other methods. On high-resolution tasks (*e.g.*, $256^2 \rightarrow 2048^2$), InvSR achieves the best performance (1.7844 s) and demonstrates a significant advantage over other methods. Overall, TUDSR-S and InvSR alternately lead in tasks with varying resolutions, while OSEDiff consistently takes the longest time across all tested scenarios.

4.3. Ablation Study

Since our TUDSR-S is a twice upsampling-diffusion model, it involves two-stage upsampling during inference. Thus, we conduct ablation studies on TUDSR-S, primarily focusing on $\times 4$ and $\times 8$ SR tasks, as presented in Tabs. 4 and 5. Note that in the tables, `M4/N4` denotes $M = 4/N = 4$ using only the stage-1/stage-2 LoRA SR model. Similarly, `M8/N8` denotes $M = 4/N = 4$. `M2N2` indicates $M = 2$ using the first LoRA SR model, and then $N = 2$ using the second LoRA SR model. Likewise, `M4N2` follows suit, and so on. Please refer to Fig. 2(b) for a better understanding.

4.3.1. Ablation Study on TUDSR-S ($\times 4$)

In the $\times 4$ SR task, the resolutions obtained after $\times 4$ upsampling RealSR, DRealSR, and RealLQ250 are 512^2, 512^2, and 1024^2, respectively. RealLR200 has non-fixed resolutions, and after $\times 4$ upsampling, the resulting resolution ranges from 512 to 1728. As shown in Tab. 4, `M2N2` achieves the best overall performance across all metrics and datasets. Meanwhile, `M4`, which follows the conventional $\times 4$ SR approach, performs second best overall performance. In contrast, `N4` yields extremely poor results. The experiments demonstrate that our twice upsampling-diffusion method, which decomposes $\times 4$ SR into two $\times 2$ upsampling stages, also achieves outstanding performance.

4.3.2. Ablation Study on TUDSR-S ($\times 8$)

In the $\times 8$ SR task, the resolutions obtained after $\times 8$ upsampling RealSR, DRealSR, and RealLQ250 are 1024^2, 1024^2, and 2048^2, respectively. In Tab. 5, `M4N2` significantly outperforms `M8` and `N8` across all metrics on RealLQ250. On RealSR and DRealSR, `M4N2` also achieves the best overall performance. Similarly, `N8` produces extremely poor results. This validates the effectiveness of our twice upsampling-diffusion approach, demonstrating that decomposing $\times 8$ SR into two stages ($\times 4$ and $\times 2$) is an effective strategy for achieving higher SR.

4.3.3. Visualization on TUDSR-S ($\times 4$ and $\times 8$)

Figure 6 provides a visualization of the aforementioned ablation study. In both $\times 4$ and $\times 8$ SR tasks, `M2N2` and `M4N2` demonstrate outstanding performance in terms of detail reproduction. In comparison, `M4` and `M8` exhibit relatively fewer details, while `N4` and `N8` fail to generate satisfactory results. The quality of the generated images further demonstrates the effectiveness of our method.

Figure 6. Visualization of twice upsampling-diffusion ($\times 4/\times 8$ *i.e.* $256^2 \rightarrow 1024^2/256^2 \rightarrow 2048^2$). Please **zoom in**.

Table 4. Ablation study on twice upsampling-diffusion (total $\times 4$).

Datasets	Type	C-IQA↑	C-IQA+↑	NIMA↑	NIQE↓	LIQE↑	MUSIQ↑	MANIQA↑
RealSR	M4	0.6657	0.6952	4.8549	5.0086	4.1895	69.2568	**0.6818**
	N4	0.3056	0.4397	3.8149	8.2005	1.3971	28.5303	0.3472
	M2N2	**0.6846**	**0.7135**	**4.8676**	**4.7149**	**4.3738**	**70.2406**	0.6786
DrealSR	M4	**0.6984**	0.6905	4.7232	5.6939	4.1050	65.7759	**0.6387**
	N4	0.2947	0.3790	3.5315	9.8559	1.2533	23.9991	0.3269
	M2N2	0.6944	**0.6921**	**4.7272**	**5.5427**	**4.1769**	**66.3650**	0.6312
RealLQ250	M4	0.7124	0.7177	5.2862	**3.5207**	4.1257	70.4876	**0.6351**
	N4	0.2953	0.3939	4.1043	7.4385	1.3163	31.3223	0.3341
	M2N2	**0.7315**	**0.7297**	**5.3268**	3.9440	**4.3017**	**71.9250**	0.6329
RealLR200	M4	0.7109	0.7290	**5.3926**	**3.6845**	4.2609	70.3921	**0.6611**
	N4	0.3677	0.4853	4.4882	5.4621	1.7350	42.7009	0.4200
	M2N2	**0.7323**	**0.7388**	5.3212	4.0156	**4.3648**	**71.3676**	0.6545

Table 5. Ablation study on twice upsampling-diffusion (total $\times 8$).

Datasets	Type	C-IQA↑	C-IQA+↑	NIMA↑	NIQE↓	LIQE↑	MUSIQ↑	MANIQA↑
RealSR	M8	**0.6928**	0.6833	**4.8379**	**4.4895**	3.3738	65.7316	0.5970
	N8	0.2672	0.2933	3.8038	11.0800	1.1892	19.6518	0.3013
	M4N2	0.6920	**0.6883**	4.8305	4.6839	**3.6547**	**67.2225**	**0.6126**
DrealSR	M8	0.7056	0.6554	4.6973	**5.3613**	3.1561	60.1592	0.5490
	N8	0.2987	0.2738	3.5905	12.9729	1.3113	19.5022	0.3086
	M4N2	**0.7186**	**0.6680**	**4.7217**	5.5417	**3.6124**	**63.9383**	**0.5758**
RealLQ250	M8	0.6723	0.6347	4.5271	4.1017	2.7134	51.5389	0.5309
	N8	0.2635	0.2712	3.9308	11.1885	1.1709	19.6954	0.2946
	M4N2	**0.7222**	**0.6904**	**5.3103**	**3.9251**	**3.6112**	**63.2129**	**0.5847**

5. Conclusion

In this work, we present TUDSR, a novel Twice Upsampling–Diffusion SR framework that effectively addresses the challenge of high-resolution image generation from a low-resolution generative model. By decomposing the demanding upsampling process into two manageable stages, our approach successfully circumvents the resolution limitations inherent in native SD models. Based on SD2.1-base, we instantiate TUDSR-S, which achieves excellent performance in $\times 4$ and $\times 8$ SR tasks, especially in addressing the shortcomings of existing SR models in the high-resolution SR tasks (*e.g.* 2048^2). Furthermore, the TUDSR framework establishes a generalizable strategy, paving the way for its application to larger generative models (*e.g.* FLUX.1-dev) and future advancements towards 4096^2 resolution.

6. Acknowledgments

This research is supported by the General Program of Shanghai Natural Science Foundation (No.24ZR1419800, No.23ZR1419300), the National Natural Science Foundation of China (No.42130112), the Ministry of Industry and Information Technology of China, Science and Technology Commission of Shanghai Municipality (No.22DZ2229004), and the Shanghai Frontiers Science Center of Molecule Intelligent Syntheses.

References

[1] Yuang Ai, Xiaoqiang Zhou, Huaibo Huang, Xiaotian Han, Zhengyu Chen, Quanzeng You, and Hongxia Yang. Dreamclear: High-capacity real-world image restoration with privacy-safe dataset curation. *Advances in Neural Information Processing Systems*, 37:55443–55469, 2024. 5

[2] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021. 4

[3] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3086–3095, 2019. 5

[4] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3086–3095, 2019. 1

[5] Honggang Chen, Xiaohai He, Linbo Qing, Yuanyuan Wu, Chao Ren, Ray E Sheriff, and Ce Zhu. Real-world single image super-resolution: A brief review. *Information Fusion*, 79:124–145, 2022. 1

[6] Linwei Chen, Lin Gu, and Ying Fu. Frequency-dynamic attention modulation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22620–22632, 2025. 4

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014. 1

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[9] Yitong Dong, Qi Zhang, Minchao Jiang, Zhiqiang Wu, Qingnan Fan, Ying Feng, Huaqi Zhang, Hujun Bao, and Guofeng Zhang. One-shot refiner: Boosting feed-forward novel view synthesis via one-step diffusion. *arXiv preprint arXiv:2601.14161*, 2026. 1

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 4

[11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 1

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2, 3

[13] Pablo Hernández-Cámara, Jose Manuel Jaén-Lorites, Jorge Vila-Tomás, Valero Laparra, and Jesus Malo. Do vision transformers see like humans? evaluating their perceptual alignment. *arXiv preprint arXiv:2508.09850*, 2025. 4

[14] 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. 5

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 1

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 5

[18] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021. 5

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

[20] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1, 2

[21] Jianze Li, Jiezhang Cao, Zichen Zou, Xiongfei Su, Xin Yuan, Yulun Zhang, Yong Guo, and Xiaokang Yang. Unleashing the power of one-step diffusion based image super-resolution via a large-scale diffusion discriminator. *arXiv preprint arXiv:2410.04224*, 2024. 4

[22] Jianze Li, Jiezhang Cao, Yong Guo, Wenbo Li, and Yulun Zhang. One diffusion step to real-world super-resolution via flow trajectory distillation. *arXiv preprint arXiv:2502.01993*, 2025. 1

[23] Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, et al. Lsdir: A large scale dataset for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1775–1787, 2023. 5

[24] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion

prior. In *European conference on computer vision*, pages 430–448. Springer, 2024. 2, 5, 6

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

[26] Or Madar and Ohad Fried. Tiled diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7795–7804, 2025. 1, 2

[27] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012. 5

[28] Brian B Moser, Arundhati S Shanbhag, Federico Raue, Stanislav Frolov, Sebastian Palacio, and Andreas Dengel. Diffusion models, image super-resolution, and everything: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 1

[29] 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*, pages 10684–10695, 2022. 1, 2

[30] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. 3

[31] Lingchen Sun, Rongyuan Wu, Zhiyuan Ma, Shuaizheng Liu, Qiaosi Yi, and Lei Zhang. Pixel-level and semantic-level adjustable super-resolution: A dual-lora approach. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2333–2343, 2025. 1, 2, 5, 6

[32] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE transactions on image processing*, 27(8): 3998–4011, 2018. 5

[33] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2555–2563, 2023. 5

[34] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *International Journal of Computer Vision*, 132(12):5929–5949, 2024. 2, 5, 6

[35] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021. 1, 2, 5, 6

[36] Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C Kot, and Bihan Wen. Sinsr: diffusion-based image super-resolution in a single step. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25796–25805, 2024. 1, 2, 5, 6

[37] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[38] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020. 1

[39] Pengxu Wei, Ziwei Xie, Hannan Lu, Zongyuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution. In *European conference on computer vision*, pages 101–117. Springer, 2020. 5

[40] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *Advances in Neural Information Processing Systems*, 37:92529–92553, 2024. 1, 2, 5, 6

[41] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25456–25467, 2024. 2, 5, 6

[42] Chunlong Xia, Xinliang Wang, Feng Lv, Xin Hao, and Yifeng Shi. Vit-comer: Vision transformer with convolutional multi-scale feature interaction for dense predictions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5493–5502, 2024. 4

[43] Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1191–1200, 2022. 5

[44] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in Neural Information Processing Systems*, 36:13294–13307, 2023. 2, 5, 6

[45] Zongsheng Yue, Kang Liao, and Chen Change Loy. Arbitrary-steps image super-resolution via diffusion inversion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23153–23163, 2025. 2, 5, 6

[46] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 1

[47] Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019. 4

[48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4, 5

[49] Weixia Zhang, Guangtao Zhai, Ying Wei, Xiaokang Yang, and Kede Ma. Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14071–14081, 2023. 5