
One-Step Effective Diffusion Network for Real-World Image Super-Resolution

Rongyuan Wu^{1,2,*}, Lingchen Sun^{1,2,*}, Zhiyuan Ma^{1,*}, Lei Zhang^{1,2,†}

¹The Hong Kong Polytechnic University ²OPPO Research Institute

{rong-yuan.wu, ling-chen.sun, zm2354.ma}@connect.polyu.hk, cslzhang@comp.polyu.edu.hk

*Equal contribution [†]Corresponding author

Abstract

The pre-trained text-to-image diffusion models have been increasingly employed to tackle the real-world image super-resolution (Real-ISR) problem due to their powerful generative image priors. Most of the existing methods start from random noise to reconstruct the high-quality (HQ) image under the guidance of the given low-quality (LQ) image. While promising results have been achieved, such Real-ISR methods require multiple diffusion steps to reproduce the HQ image, increasing the computational cost. Meanwhile, the random noise introduces uncertainty in the output, which is unfriendly to image restoration tasks. To address these issues, we propose a one-step effective diffusion network, namely OSEDiff, for the Real-ISR problem. We argue that the LQ image contains rich information to restore its HQ counterpart, and hence the given LQ image can be directly taken as the starting point for diffusion, eliminating the uncertainty introduced by random noise sampling. We finetune the pre-trained diffusion network with trainable layers to adapt it to complex image degradations. To ensure that the one-step diffusion model could yield HQ Real-ISR output, we apply variational score distillation in the latent space to conduct KL-divergence regularization. As a result, our OSEDiff model can efficiently and effectively generate HQ images in just one diffusion step. Our experiments demonstrate that OSEDiff achieves comparable or even better Real-ISR results, in terms of both objective metrics and subjective evaluations, than previous diffusion model-based Real-ISR methods that require dozens or hundreds of steps. The source codes are released at <https://github.com/cswry/OSEDiff>.

1 Introduction

Image super-resolution (ISR) [13, 66, 29, 65, 6, 24, 46, 27, 61] is a classical yet still active research problem, which aims to restore a high-quality (HQ) image from its low-quality (LQ) observation suffering from degradations of noise, blur and low-resolution, *etc*. While one line of ISR research [13, 66, 29, 65, 6] simplifies the degradation process from HQ to LQ images as bicubic downsampling (or downsampling after Gaussian blur) and focus on the study on network architecture design, the trained models can hardly be generalized to real-world LQ images, whose degradations are often unknown and much more complex. Therefore, another increasingly popular line of ISR research is the so-called real-world ISR (Real-ISR) [61, 45] problem, which aims to reproduce perceptually realistic HQ images from the LQ images captured in real-world applications.

There are two major issues in training a Real-ISR model. One is how to build the LQ-HQ training image pairs, and another is how to ensure the naturalness of restored images, *i.e.*, how to ensure that the restored images follow the distribution of HQ natural images. For the first issue, some researchers

This work is supported by the PolyU-OPPO Joint Innovation Lab.

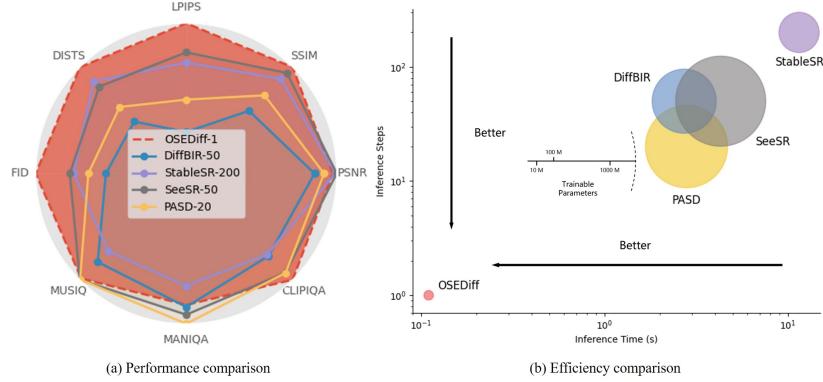


Figure 1: Performance and efficiency comparison among SD-based Real-ISR methods. (a). Performance comparison on the DrealSR benchmark [51]. Metrics like LPIPS and NIQE, where smaller scores indicate better image quality, are inverted and normalized for display. OSEDiff achieves leading scores on most metrics with only one diffusion step. (b). Model efficiency comparison. The inference time is tested on an A100 GPU with 512×512 input image size. OSEDiff has the fewest trainable parameters and is over 100 times faster than StableSR [42].

have proposed to collect real-world LQ-HQ image pairs using long-short camera focal lenses [3, 51]. However, this is very costly and can only cover certain types of real-world image degradations. Another more economical way is to simulate the real-world LQ-HQ image pairs by using complex image degradation pipelines. The representative works include BSRGAN [61] and Real-ESRGAN [45], where a random shuffling of basic degradation operators and a high-order degradation model are respectively used to generate LQ-HQ image pairs.

With the given training data, how to train a robust Real-ISR model to output perceptually natural images with high quality becomes a key issue. Simply learning a mapping network between LQ-HQ paired data with pixel-wise losses can lead to over-smoothed results [24, 46]. It is crucial to integrate natural image priors into the learning process to reproduce HQ images. A few methods have been proposed to this end. The perceptual loss [18] explores the texture, color, and structural priors in a pre-trained model such as VGG-16 [38] and AlexNet [23]. The generative adversarial networks (GANs) [14] alternatively train a generator and a discriminator, and they have been adopted for Real-ISR tasks [24, 46, 45, 61, 27, 53]. The generator network aims to synthesize HQ images, while the discriminator network aims to distinguish whether the synthesized image is realistic or not. While great successes have been achieved, especially in the restoration of specific classes of images such as face images [56, 44], GAN-based Real-ISR tends to generate unpleasant details due to the unstable adversarial training and the difficulties in discriminating the image space of diverse natural scenes.

The recently developed generative diffusion models (DM) [39, 16], especially the large-scale pre-trained text-to-image (T2I) models [37, 36], have demonstrated remarkable performance in various downstream tasks. Having been trained on billions of image-text pairs, the pre-trained T2I models possess powerful natural image priors, which can be well exploited to improve the naturalness and perceptual quality of Real-ISR outputs. Some methods [42, 57, 31, 52, 40, 59] have been developed to employ the pre-trained T2I model for solving the Real-ISR problem. While having shown impressive results in generating richer and more realistic image details than GAN-based methods, the existing SD-based methods have several problems to be further addressed. First, these methods typically take random Gaussian noise as the start point of the diffusion process. Though the LQ images are used as the control signal with a ControlNet module [63], these methods introduce unwanted randomness in the output HQ images [40]. Second, the restored HQ images are usually obtained by tens or even hundreds of diffusion steps, making the Real-ISR process computationally expensive. Though some one-step diffusion based Real-ISR methods [48] have been recently proposed, they fail in achieving high-quality details compared to multi-step methods.

To address the aforementioned issues, we propose a **One-Step Effective Diffusion** network, **OSEDiff** in short, for the Real-ISR problem. The UNet backbone in pre-trained SD models has strong capability to transfer the input data into another domain, while the given LQ image actually has rich information to restore its HQ counterpart. Therefore, we propose to directly feed the LQ images into the pre-trained SD model without introducing any random noise. Meanwhile, we integrate trainable

LoRA layers [17] into the pre-trained UNet, and finetune the SD model to adapt it to the Real-ISR task. On the other hand, to ensure that the one-step model can still produce HQ natural images as the multi-step models, we utilize variational score distillation (VSD) [49, 58, 10] for KL-divergence regularization. This operation effectively leverages the powerful natural image priors of pre-trained SD models and aligns the distribution of generated images with natural image priors. As illustrated in Fig. 1, our extensive experiments demonstrate that OSEDiff achieves comparable or superior performance measures to state-of-the-art SD-based Real-ISR models, while it significantly reduces the number of inference steps from N to 1 and has the fewest trainable parameters, leading to more than $\times 100$ speedup in inference time over previous methods such as StableSR [42].

2 Related Work

Starting from SRCNN [13], deep learning-based methods have become prevalent for ISR. A variety of methods have been proposed [30, 66, 67, 9, 5, 29, 65, 6, 7] to improve the accuracy of ISR reconstruction. However, most of these methods assume simple and known degradations such as bicubic downsampling, limiting their applications to real-world images with complex and unknown degradations. In recent years, researchers have been exploring the potentials of generative models, including GAN [14] and diffusion networks [16], for solving the Real-ISR problem.

GAN-based Real-ISR. The use of GAN for photo-realistic ISR can be traced back to SRGAN [24], where the image degradation is assumed to be bicubic downsampling. Later on, researchers found that GAN has the potential to perform real-world image restoration with more complex degradations [61, 45]. Specifically, by using randomly shuffled degradation and high-order degradation to generate more realistic LQ-HQ training pairs, BSRGAN [61] and Real-ESRGAN [45] demonstrate promising Real-ISR results, which trigger many following works [4, 27, 28, 53]. DASR [28] designs a tiny network to predict the degradation parameters to handle degradations of various levels. SwinIR [29] switches the generator from CNNs to stronger transformers, further enhancing the performance of Real-ISR. However, the adversarial training process in GAN is unstable and its discriminator is limited in telling the quality of diverse natural image contents. Therefore, GAN-based Real-ISR methods often suffer from unnatural visual artifacts. Some works such as LDL [27] and DeSRA [53] can suppress much the artifacts, yet they are difficult to generate more natural details.

Diffusion-based Real-ISR. Some early attempts [21, 20, 47] employ the denoising diffusion probabilistic models (DDPMs) [16, 39, 11] to address the ISR problem by assuming simple and known degradations (e.g., bicubic downsampling). These methods are training-free by modifying the reverse transition of pre-trained DDPMs using gradient descent, but they cannot be applied to complex unknown degradations. Recent researches [42, 57, 31, 40, 59] have leveraged stronger pre-trained T2I models, such as Stable Diffusion (SD) [1], to tackle the Real-ISR problem. In general, they introduce an adapter [63] to fine-tune the SD model to reconstruct the HQ image with the LQ image as the control signal. StableSR [42] finetunes a time-aware encoder and employs feature warping to balance fidelity and perceptual quality. PASD [57] extracts both low-level and high-level features from the LQ image and inputs them to the pre-trained SD model with a pixel-aware cross attention module. To further enhance the semantic-aware ability of the Real-ISR model, SeeSR [52] introduces degradation-robust tag-style text prompts and utilizes soft prompts to guide the diffusion process. To mitigate the potential risks of diffusion uncertainty, CCSR [40] leverages a truncated diffusion process to recover structures and finetunes the VAE decoder by adversarial training to enhance details. SUPIR [59] leverages the powerful generation capability of SDXL model and the strong captioning capability of LLaVA [32] to synthesize rich image details.

The above mentioned methods, however, require tens or even hundreds of steps to complete the diffusion process, resulting in unfriendly latency. SinSR shortens ResShift [60] to a single-step inference by consistency preserving distillation. Nevertheless, the non-distribution-based distillation loss tends to obtain smooth results, and the model capacity of SinSR and ResShift are much smaller than the SD models to address Real-ISR problems.

3 Methodology

3.1 Problem Modeling

Real-ISR is to estimate an HQ image $\hat{\mathbf{x}}_H$ from the given LQ image \mathbf{x}_L . This task can be conventionally modeled as an optimization problem: $\hat{\mathbf{x}}_H = \operatorname{argmin}_{\mathbf{x}_H} (\mathcal{L}_{data}(\Phi(\mathbf{x}_H), \mathbf{x}_L) + \lambda \mathcal{L}_{reg}(\mathbf{x}_H))$, where Φ is the degradation function, \mathcal{L}_{data} is the data term to measure the fidelity of the optimization

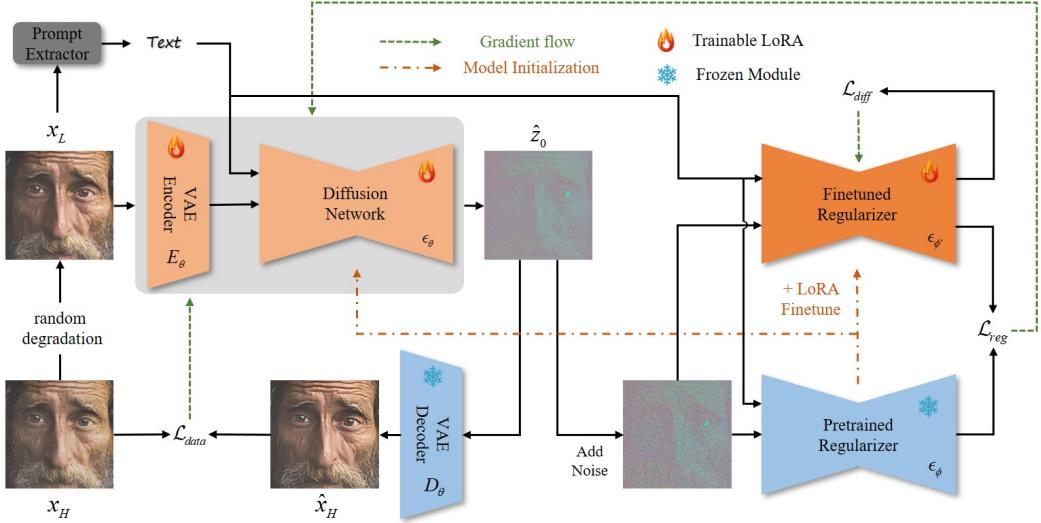


Figure 2: The training framework of OSEDiff. The LQ image is passed through a trainable encoder E_θ , a LoRA finetuned diffusion network ϵ_θ and a frozen decoder D_θ to obtain the desired HQ image. In addition, text prompts are extracted from the LQ image and input to the diffusion network to stimulate its generation capacity. Meanwhile, the output of the diffusion network ϵ_θ will be sent to two regularizer networks (a frozen pre-trained one and a fine-tuned one), where variational score distillation is performed in latent space to ensure that the output of ϵ_θ follows HQ natural image distribution. The regularization loss will be back-propagated to update E_θ and ϵ_θ . Once training is finished, only E_θ , ϵ_θ and D_θ will be used in inference.

output, \mathcal{L}_{reg} is the regularization term to exploit the prior information of natural images, and scalar λ is the balance parameter. Many conventional ISR methods [13, 29, 65] restore the desired HQ image by assuming simple and known degradation models and employing hand-crafted natural image prior models (*i.e.*, image sparsity based prior [54]).

However, the performance of such optimization-based methods is largely hindered by two factors. First, the degradation function Φ is often unknown and hard to model in real-world scenarios. Second, the hand-crafted regularization terms \mathcal{L}_{reg} are hard to effectively model the complex natural image priors. With the development of deep-learning techniques, it has become prevalent to learn a neural network G_θ , which is parameterized by θ , from a training dataset S of $(\mathbf{x}_L, \mathbf{x}_H)$ pairs to map the LQ image to an HQ image. The network training can be described as the following learning problem:

$$\theta^* = \operatorname{argmin}_{\theta} \mathbb{E}_{(\mathbf{x}_L, \mathbf{x}_H) \sim S} [\mathcal{L}_{data}(G_\theta(\mathbf{x}_L), \mathbf{x}_H) + \lambda \mathcal{L}_{reg}(G_\theta(\mathbf{x}_L))], \quad (1)$$

where \mathcal{L}_{data} and \mathcal{L}_{reg} are the loss functions. \mathcal{L}_{data} enforces that the network output $\hat{\mathbf{x}}_H = G_\theta(\mathbf{x}_L)$ can approach to the ground-truth \mathbf{x}_H as much as possible, which can be quantified by metrics such as L_1 norm, L_2 norm and LPIPS [64]. Using only the \mathcal{L}_{data} loss to train the network G_θ from scratch may over-fit the training dataset. In this work, we propose to finetune a pre-trained generative network, more specifically the SD [36] network, to improve the generalization capability of G_θ . In addition, the regularization loss \mathcal{L}_{reg} is critical to improve the generalization capability of G_θ , as well as enhance the naturalness of output HQ images $\hat{\mathbf{x}}_H$. Suppose that we have the distribution of real-world HQ images, denoted by $p(\mathbf{x}_H)$, the KL divergence [8] is an ideal choice to serve as the loss function of \mathcal{L}_{reg} ; that is, the distribution of restored HQ images, denoted by $q_\theta(\hat{\mathbf{x}}_H)$, should be identical to $p(\mathbf{x}_H)$ as much as possible. The regularization loss can be defined as:

$$\mathcal{L}_{reg} = \mathcal{D}_{KL}(q_\theta(\hat{\mathbf{x}}_H) \| p(\mathbf{x}_H)). \quad (2)$$

Existing works [24, 46] mostly instantiate the above objective via adversarial training [14], which involves learning a discriminator to differentiate between the generated HQ image $\hat{\mathbf{x}}_H$ and the real HQ image \mathbf{x}_H , and updating the generator G_θ to make $\hat{\mathbf{x}}_H$ and \mathbf{x}_H indistinguishable. However, the discriminators are often trained from scratch alongside the generator. They may not be able to acquire the full distribution of HQ images and lack enough discriminative power, resulting in sub-optimal Real-ISR performance.

The recently developed T2I diffusion models such as SD [36] offer new options for us to formulate the loss \mathcal{L}_{reg} . These models, trained on billions of image-text pairs, can effectively depict the natural image distribution in latent space. Some score distillation methods have been reported to employ SD to optimize images by using the KL-divergence as the objective [49, 25, 43]. In particular, variational score distillation (VSD) [49, 58, 10] induces such a KL-divergence based objective from particle-based variational optimization to align the distributions represented by two diffusion models. Based on the above discussions, we propose to instantiate the learning objective in Eq. (1) by designing an efficient and effective one-step diffusion network. In specific, we finetune the pre-trained SD with LoRA [17] as our Real-ISR backbone network and employ VSD as our regularizer to align the distribution of network outputs with natural HQ images. The details are provided in the next section.

3.2 One-Step Effective Diffusion Network

Framework Overview. As discussed in Sec. 1, the existing SD-based Real-ISR methods [42, 57, 31, 52, 40] perform multiple timesteps to estimate the HQ image with random noise as the starting point and the LQ image as control signal. These approaches are resource-intensive and will inherently introduce randomness. Based on our formulation in Sec. 3.1, we propose a one-step effective diffusion (OSEDiff) network for Real-ISR, whose training framework is shown in Fig. 2. Our generator G_θ to be trained is composed of a trainable encoder E_θ , a finetuned diffusion network ϵ_θ and a frozen decoder D_θ . To ensure the generalization capability of G_θ , the output of the diffusion network ϵ_θ will be sent to two regularizer networks, where VSD loss is performed in latent space. The regularization loss are back-propagated to update E_θ and ϵ_θ . Once training is finished, only the generator G_θ will be used in inference. In the following, we will delve into the detailed architecture design of OSEDiff, as well as its associated training losses.

Network Architecture Design. Let’s denote by E_ϕ , ϵ_ϕ and D_ϕ the VAE encoder, latent diffusion network and VAE decoder of a pretrained SD model, where ϕ denotes the model parameters. Inspired by the recent success of LoRA [17] in finetuning SD to downstream tasks [34, 35], we adopt LoRA to fine-tune the pre-trained SD in the Real-ISR task to obtain the desired generator G_θ .

As shown in the left part of Fig. 2, to maintain SD’s original generation capacity, we introduce trainable LoRA [17] layers to the encoder E_ϕ and the diffusion network ϵ_ϕ , finetuning them into E_θ and ϵ_θ with our training data. For the decoder, we fix its parameters and directly set $D_\theta = D_\phi$. This is to ensure that the output space of the diffusion network remains consistent with the regularizers.

Recall that the diffusion model diffuses the input latent feature \mathbf{z} through $\mathbf{z}_t = \alpha_t \mathbf{z} + \beta_t \epsilon$, where α_t, β_t are scalars that are dependent to diffusion timestep $t \in \{1, \dots, T\}$ [16]. With a neural network that can predict the noise in \mathbf{z}_t , denoted as $\hat{\epsilon}$, the denoised latent can be obtained as $\hat{\mathbf{z}}_0 = \frac{\mathbf{z}_t - \beta_t \hat{\epsilon}}{\alpha_t}$, which is expected to be cleaner and more photo-realistic than \mathbf{z}_t . Moreover, SD is a text-conditioned generation model. By extracting the text embeddings [36], denoted by c_y , from the given text description y , the noise prediction can be performed as $\hat{\epsilon} = \epsilon_\theta(\mathbf{z}_t; t, c_y)$.

We adapt the above text-to-image denoising process to the Real-ISR task, and formulate the LQ-to-HQ latent transformation F_θ as a text-conditioned image-to-image denoising process as:

$$\hat{\mathbf{z}}_H = F_\theta(\mathbf{z}_L; c_y) \triangleq \frac{\mathbf{z}_L - \beta_T \epsilon_\theta(\mathbf{z}_L; T, c_y)}{\alpha_T}, \quad (3)$$

where we conduct only one-step denoising on the LQ latent \mathbf{z}_L , without introducing any noise, at the T -th diffusion timestep. The denoising output $\hat{\mathbf{z}}_H$ is expected be more photo-realistic than \mathbf{z}_L . As for the text embeddings, we apply some text prompt extractor, such as the DAPE [52], to LQ input \mathbf{x}_L , and obtain $c_y = Y(\mathbf{x}_L)$. Finally, the whole LQ-to-HQ image synthesis can be written as:

$$\hat{\mathbf{x}}_H = G_\theta(\mathbf{x}_L) \triangleq D_\theta(F_\theta(E_\theta(\mathbf{x}_L); Y(\mathbf{x}_L))). \quad (4)$$

As mentioned in Sec. 3.1, to improve the performance for a Real-ISR model, it is required to supervise the generator training with both the data term $\mathcal{L}_{\text{data}}$ and regularization term \mathcal{L}_{reg} . As shown in the right part of Fig. 2, we propose to adapt VSD [49] as the regularization term. Apart from utilizing the SD model as the pre-trained regularizer ϵ_ϕ , VSD also introduces a finetuned regularizer, *i.e.*, a latent diffusion module finetuned on the distribution $q_\theta(\hat{\mathbf{x}}_H)$ of generated images with LoRA. We denote this finetuned diffusion module as $\epsilon_{\phi'}$.

Training Loss. We train the generator G_θ with the data loss $\mathcal{L}_{\text{data}}$ and regularization loss \mathcal{L}_{reg} . We set $\mathcal{L}_{\text{data}}$ as the weighted sum of MSE loss and LPIPS loss:

$$\mathcal{L}_{\text{data}}(G_\theta(\mathbf{x}_L), \mathbf{x}_H) = \mathcal{L}_{\text{MSE}}(G_\theta(\mathbf{x}_L), \mathbf{x}_H) + \lambda_1 \mathcal{L}_{\text{LPIPS}}(G_\theta(\mathbf{x}_L), \mathbf{x}_H), \quad (5)$$

where λ_1 is a weighting scalar. As for \mathcal{L}_{reg} , we adopt the VSD loss via:

$$\mathcal{L}_{\text{reg}}(G_\theta(\mathbf{x}_L)) = \mathcal{L}_{\text{VSD}}(G_\theta(\mathbf{x}_L), c_y) = \mathcal{L}_{\text{VSD}}(G_\theta(\mathbf{x}_L), Y(\mathbf{x}_L)). \quad (6)$$

Given any trainable image-shape feature \mathbf{x} , its latent code $\mathbf{z} = E_\phi(\mathbf{x})$ and encoded text prompt condition c_y , VSD optimizes \mathbf{x} to make it consistent with the text prompt y via:

$$\nabla_{\mathbf{x}} \mathcal{L}_{\text{VSD}}(\mathbf{x}, c_y) = \mathbb{E}_{t, \epsilon} \left[\omega(t) (\epsilon_\phi(\mathbf{z}_t; t, c_y) - \epsilon_{\phi'}(\mathbf{z}_t; t, c_y)) \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right], \quad (7)$$

where the expectation of the gradient is conducted over all diffusion timesteps $t \in \{1, \dots, T\}$ and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Therefore, the overall training objective for the generator G_θ is:

$$\mathcal{L}(G_\theta(\mathbf{x}_L), \mathbf{x}_H) = \mathcal{L}_{\text{data}}(G_\theta(\mathbf{x}_L), \mathbf{x}_H) + \lambda_2 \mathcal{L}_{\text{reg}}(G_\theta(\mathbf{x}_L)), \quad (8)$$

where λ_2 is a weighting scalar. Besides, as required by VSD, the finetuned regularizer $\epsilon_{\phi'}$ should also be trainable, and its training objective is:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t, \epsilon, c_y = Y(\mathbf{x}_L), \hat{\mathbf{z}}_H = F_\theta(E_\theta(\mathbf{x}_L); Y(\mathbf{x}_L))} \mathcal{L}_{\text{MSE}}(\epsilon_{\phi'}(\alpha_t \hat{\mathbf{z}}_H + \beta_t \epsilon; t, c_y), \epsilon). \quad (9)$$

Note that the above $\mathcal{L}_{\text{diff}}$ loss is only applied to update $\epsilon_{\phi'}$. The whole algorithm to illustrate the training pipeline can be found in the **Appendix**.

VSD in Latent Space. The original VSD computes the gradients in the image space. When it is used to train an SD-based generator network, there will be repetitive latent decoding/encoding in computing \mathcal{L}_{reg} . This is costly and makes the regularization less effective. Considering the fact that a well-trained VAE should satisfy $E_\phi(\mathbf{x}) = E_\phi(D_\phi(\mathbf{z})) \approx \mathbf{z}$, we can approximately let $E_\phi(\hat{\mathbf{x}}_H) = \hat{\mathbf{z}}_H$. In this case, we can eliminate the redundant latent encoding/decoding in computing the regularization loss, as we follow DMD [58] to optimize the distribution loss in the latent state space rather than in the noise space. The gradient of the regularization loss w.r.t. the network parameter θ in the latent space is:

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{VSD}}(G_\theta(\mathbf{x}_L), c_y) &= \nabla_{\hat{\mathbf{x}}_H} \mathcal{L}_{\text{VSD}}(\hat{\mathbf{x}}_H, c_y) \frac{\partial \hat{\mathbf{x}}_H}{\partial \theta} \\ &= \mathbb{E}_{t, \epsilon, \hat{\mathbf{z}}_t = \alpha_t E_\phi(\hat{\mathbf{x}}_H) + \beta_t \epsilon} \left[\omega(t) (\epsilon_\phi(\hat{\mathbf{z}}_t; t, c_y) - \epsilon_{\phi'}(\hat{\mathbf{z}}_t; t, c_y)) \frac{\partial \hat{\mathbf{z}}_H}{\partial \hat{\mathbf{x}}_H} \frac{\partial \hat{\mathbf{x}}_H}{\partial \theta} \right] \\ &= \mathbb{E}_{t, \epsilon, \hat{\mathbf{z}}_t = \alpha_t \hat{\mathbf{z}}_H + \beta_t \epsilon} \left[\omega(t) (\epsilon_\phi(\hat{\mathbf{z}}_t; t, c_y) - \epsilon_{\phi'}(\hat{\mathbf{z}}_t; t, c_y)) \frac{\partial \hat{\mathbf{z}}_H}{\partial \theta} \right]. \end{aligned} \quad (10)$$

4 Experiments

4.1 Experimental Settings

Training and Testing Datasets. Prior works [42, 57, 31, 52] employed different training datasets, making unified training standards for Real-ISR difficult to establish. For simplicity, we adopt SeeSR’s setup [52] and train OSEDiff using the LSDIR [26] dataset and the first 10K face images from FFHQ [19]. The degradation pipeline of Real-ESRGAN [45] is used to synthesize LQ-HQ training pairs. We evaluate OSEDiff and compare it with competing methods using the test set provided by StableSR [42], including both synthetic and real-world data. The synthetic data includes 3000 images of size 512×512 , whose GT are randomly cropped from DIV2K-Val [2] and degraded using the Real-ESRGAN pipeline [45]. The real-world data include LQ-HQ pairs from RealSR [3] and DRealSR [51], with sizes of 128×128 and 512×512 , respectively.

Compared Methods. We compare OSEDiff with state-of-the-art DM-based Real-ISR methods, including StableSR [42], ResShift [60], PASD [57], DiffBIR [31], SeeSR [52] and SinSR [48]. Among them, StableSR, PASD, DiffBIR, and SeeSR are all based on the pre-trained SD model. ResShift trains a DM from scratch in the pixel domain, while SinSR is a one-step model distilled

Table 1: Quantitative comparison with state-of-the-art methods on both synthetic and real-world benchmarks. ‘s’ denotes the number of diffusion reverse steps in the method. The best and second best results of each metric are highlighted in **red** and **blue**, respectively.

Datasets	Methods	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓	NIQE↓	MUSIQ↑	MANIQA↑	CLIPQA↑
DIV2K-Val	StableSR-s200	23.26	0.5726	0.3113	0.2048	24.44	4.7581	65.92	0.6192	0.6771
	DiffBIR-s50	23.64	0.5647	0.3524	0.2128	30.72	4.7042	65.81	0.6210	0.6704
	SeeSR-s50	23.68	0.6043	0.3194	0.1968	25.90	4.8102	68.67	0.6240	0.6936
	PASD-s20	23.14	0.5505	0.3571	0.2207	29.20	4.3617	68.95	0.6483	0.6788
	ResShift-s15	24.45	0.6181	0.3349	0.2213	36.11	6.8212	61.09	0.5454	0.6071
	SinSR-s1	24.41	0.6018	0.3240	0.2066	35.57	6.0159	62.82	0.5386	0.6471
	OSEDiff-s1	23.72	0.6108	0.2941	0.1976	26.32	4.7097	67.97	0.6148	0.6683
DrealSR	StableSR-s200	28.03	0.7536	0.3284	0.2269	148.98	6.5239	58.51	0.5601	0.6356
	DiffBIR-s50	26.71	0.6571	0.4557	0.2748	166.79	6.3124	61.07	0.5930	0.6395
	SeeSR-s50	28.17	0.7691	0.3189	0.2315	147.39	6.3967	64.93	0.6042	0.6804
	PASD-s20	27.36	0.7073	0.3760	0.2531	156.13	5.5474	64.87	0.6169	0.6808
	ResShift-s15	28.46	0.7673	0.4006	0.2656	172.26	8.1249	50.60	0.4586	0.5342
	SinSR-s1	28.36	0.7515	0.3665	0.2485	170.57	6.9907	55.33	0.4884	0.6383
	OSEDiff-s1	27.92	0.7835	0.2968	0.2165	135.30	6.4902	64.65	0.5899	0.6963
RealSR	StableSR-s200	24.70	0.7085	0.3018	0.2288	128.51	5.9122	65.78	0.6221	0.6178
	DiffBIR-s50	24.75	0.6567	0.3636	0.2312	128.99	5.5346	64.98	0.6246	0.6463
	SeeSR-s50	25.18	0.7216	0.3009	0.2223	125.55	5.4081	69.77	0.6442	0.6612
	PASD-s20	25.21	0.6798	0.3380	0.2260	124.29	5.4137	68.75	0.6487	0.6620
	ResShift-s15	26.31	0.7421	0.3460	0.2498	141.71	7.2635	58.43	0.5285	0.5444
	SinSR-s1	26.28	0.7347	0.3188	0.2353	135.93	6.2872	60.80	0.5385	0.6122
	OSEDiff-s1	25.15	0.7341	0.2921	0.2128	123.49	5.6476	69.09	0.6326	0.6693

Table 2: Complexity comparison among different methods. All methods are tested with an input image of size 512×512 , and the inference time is measured on an A100 GPU.

	StableSR	DiffBIR	SeeSR	PASD	ResShift	SinSR	OSEDiff
Inference Step	200	50	50	20	15	1	1
Inference Time (s)	11.50	2.72	4.30	2.80	0.71	0.13	0.11
MACs (G)	79940	24234	65857	29125	5491	2649	2265
# Total Param (M)	1410	1717	2524	1900	119	119	1775
# Trainable Param (M)	150.0	380.0	749.9	625.0	118.6	118.6	8.5

from ResShift. Note that we do not compare with the recent method SUPIR [59] because it tends to generate rich yet excessive details, which are however unfaithful to the input image.

For those GAN-based Real-ISR methods, including BSRGAN [61], Real-ESRGAN [45], LDL [27], and FeMaSR [4], we put their results into the **Appendix**.

Evaluation Metrics. To provide a comprehensive and holistic assessment on the performance of different methods, we employ a range of full-reference and no-reference metrics. PSNR and SSIM [50] (calculated on the Y channel in YCbCr space) are reference-based fidelity measures, while LPIPS [64], DISTS [12] are reference-based perceptual quality measures. FID [15] evaluates the distance of distributions between GT and restored images. NIQE [62], MANIQA-pipal [55], MUSIQ [22], and CLIPQA [41] are no-reference image quality measures. We also conduct a user study, which is presented in the **Appendix**.

Implementation Details. We train OSEDiff with the AdamW optimizer [33] at a learning rate of $5e-5$. The entire training process took approximately 1 day on 4 NVIDIA A100 GPUs with a batch size of 16. The rank of LoRA in the VAE Encoder, diffusion network, and finetuned regularizer is set to 4. For the text prompt extractor, although advanced multimodal language models [32] can provide detailed text descriptions, they come at a high inference cost. We adopt the degradation-aware prompt extraction (DAPE) module in SeeSR [52] to extract text prompts. The SD 2.1-base is used as the pre-trained T2I model. The weighting scalars λ_1 and λ_2 are set to 2 and 1, respectively.

4.2 Comparison with State-of-the-Arts

Quantitative Comparisons. The quantitative comparisons among the competing methods on the three datasets are presented in Table 1. We can have the following observations. (1) First, OSEDiff exhibits clear advantages over competing methods in full-reference perceptual quality metrics LPIPS and DISTS, distribution alignment metric FID, and semantic quality metric CLIPQA, especially

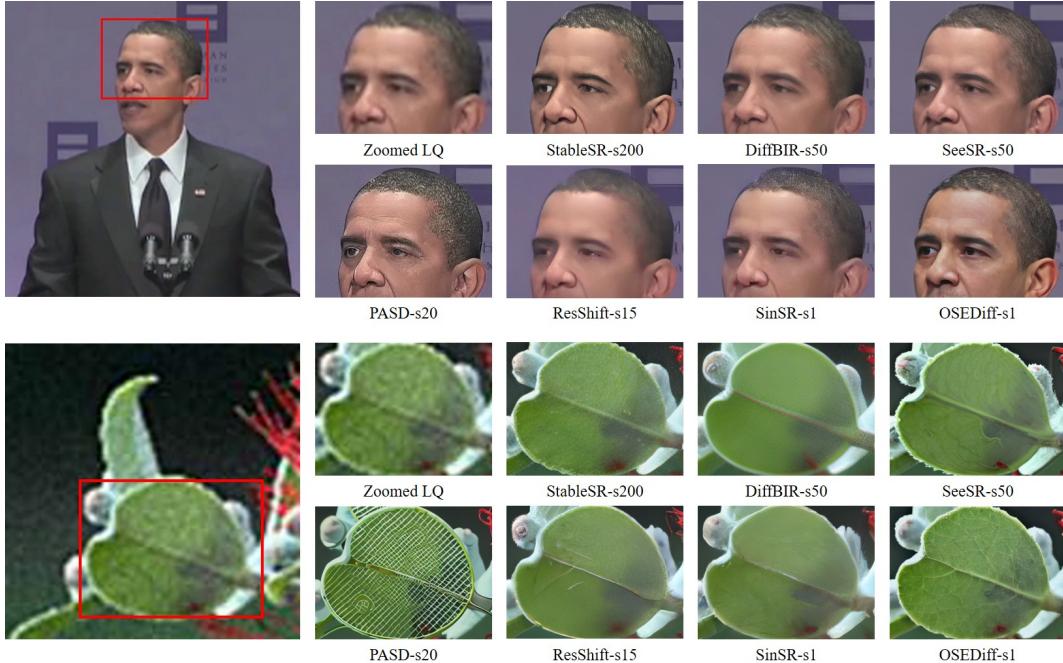


Figure 3: Qualitative comparisons of different Real-ISR methods. Please zoom in for a better view.

on the two real-world datasets DrealSR and RealSR. (2) Second, SeeSR and PASD show better no-reference metrics like NIQE, MUSIQ and MANIQA. This is because these multi-step methods can produce rich image details in the diffusion process, which are preferred by no-reference metrics. (3) Third, ResShift and its distilled version SinSR show better full-reference fidelity metrics such as PSNR. This is mainly because they train a DM from scratch specifically for the restoration purpose, instead of exploring the pre-trained T2I model such as SD. However, ResShift and SinSR show poor perceptual quality metrics than other methods.

Qualitative Comparisons. Fig. 3 presents visual comparisons of different Real-ISR methods. As illustrated in the first example, ResShift and SinSR severely blur the facial details due to the lack of pre-trained image priors. StableSR, DiffBIR and SeeSR reconstruct more facial details by exploring the image prior in pre-trained SD model. PASD generates excessive yet unnatural details. Though OSEDiff performs only one step forward propagation, it reproduces realistic and superior facial details to other methods. Similar conclusion can be drawn from the second example. StableSR and DiffBIR are limited in generating rich textures due to the lack of text prompts. PASD suffers from incorrect semantic generation because its prompt extraction module is not robust to degradation. While SeeSR utilizes degradation-aware semantic cues to stimulate image generation priors, the generated leaf veins are unnatural, which may be influenced by its random noise sampling. In contrast, OSEDiff can generate detailed and natural leaf veins. More visualization comparisons and the results of subjective user study can be found in the **Appendix**.

Complexity Comparisons. We further compare the complexity of competing DM-based Real-ISR models in Table 2, including the number of inference steps, inference time, and trainable parameters. All methods are tested on an A100 GPU with an input image of size 512×512 . OSEDiff has the fewest trainable parameters, and the trained LoRA layers can be merged into the original SD to further reduce the computational cost. By using only one forward pass, OSEDiff has significant advantage in inference time over multi-step methods. Specifically, OSEDiff demonstrates a substantial speed advantage, being approximately 105 times faster than StableSR, 39 times faster than SeeSR, and 6 times faster than ResShift. When compared to the single-step method SinSR, OSEDiff not only achieves faster inference but also delivers significantly higher output quality. In terms of complexity, OSEDiff requires the lowest MACs at just 2265G, as it operates with only a single diffusion step. In contrast, methods like StableSR, which require 200 steps, incur substantially higher MACs (e.g., 79940G). Regarding trainable parameters, OSEDiff is highly parameter-efficient, requiring only 8.5M parameters (LoRA layers), compared to models such as SeeSR, which necessitates 749.9M parameters. This highlights the efficiency of OSEDiff during the training process.

Table 3: Comparison of different losses on the RealSR benchmark.

	PSNR↑	LPIPS↓	MUSIQ↑	CLIPQA↑
w/o VSD loss	25.42	0.2934	65.23	0.5876
GAN loss	25.00	0.2760	67.29	0.6254
VSD loss in image domain	25.05	0.2759	67.90	0.6256
OSEDiff	25.15	0.2921	69.09	0.6693

Table 4: Comparison of different text prompt extractors on the DrealSR benchmark.

Prompt Extraction Methods	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓	NIQE↓	MUSIQ↑	MANIQA↑	CLIPQA↑	Prompt Extraction Time (s)
Null	28.51	0.7910	0.2896	0.2080	126.59	6.6436	62.13	0.5782	0.6599	0
DAPE	27.92	0.7835	0.2968	0.2165	135.30	6.4902	64.65	0.5899	0.6963	0.02
LLaVA-v1.5	27.72	0.7735	0.3109	0.2249	149.45	6.4119	65.70	0.6038	0.7033	3.53

Table 5: Comparison of LoRA in VAE encoder with different ranks.

Rank	PSNR↑	DISTS↓	MUSIQ↑	NIQE↓
2	-	-	-	-
4	25.15	0.2128	69.09	5.6479
8	24.86	0.2134	68.37	5.8184

Table 6: Comparison of LoRA in UNet with different ranks.

Rank	PSNR↑	DISTS↓	MUSIQ↑	NIQE↓
2	25.28	0.2154	68.89	5.7171
4	25.15	0.2128	69.09	5.6479
8	24.87	0.2115	68.39	5.8184

Table 7: Ablation studies on finetuning the VAE encoder and decoder on the RealSR benchmark.

	Train VAE Encoder	Train VAE Decoder	PSNR↑	DISTS↓	LPIPS↓	CLIPQA↑	MUSIQ↑	NIQE↓
(1)	×	×	25.27	0.1966	0.2656	0.5303	58.99	6.5496
(2)	×	✓	25.30	0.2049	0.2829	0.5604	65.83	6.6291
(3)	✓	✓	25.59	0.2141	0.3017	0.5778	65.92	6.9845
OSEDiff	✓	✗	25.15	0.2128	0.2921	0.6693	69.09	5.6479

4.3 Ablation Study

Effectiveness of VSD Loss. To validate the effectiveness of our VSD loss in latent space, we perform ablation studies by removing the VSD loss, replacing it with the GAN loss used in [35], and applying VSD loss in the image domain. The results on the RealSR test set are shown in Table 3. We can see that without using the VSD loss, the perceptual quality metrics are significantly degraded because it is hard to ensure good visual quality using only MSE loss and even LPIPS loss [64]. Using GAN loss and VSD loss in the image domain can improve the performance, but the results are not as good as applying VSD loss in the latent domain. Our proposed OSEDiff can effectively align the distribution of Real-ISR outputs by performing VSD regularization in the latent domain.

Comparison on Text Prompt Extractors. We then conduct experiments to evaluate the effect of different text prompt extractors on the Real-ISR results. We test three options. The first option does not employ text prompts. The second option uses the DAPE module in SeeSR [52] to extract degradation-aware tag-style prompts, as we used in our main experiments. The third option uses LLaVA-v1.5 [32] to extract long text descriptions after removing the degradation of input LQ images, as used in SUPIR [59]. We retrain the models based on different prompt extraction methods. The ablation results are shown in Table 4.

One can see that without using text prompts as inputs, those full-reference metrics such PSNR, SSIM, LPIPS, DISTS and even FID can improve, while those no-reference metrics such as MUSIQ, MANIQA and CLIPQA become worse. By using DAPE and LLaVA to extract text prompts, the generation capability of the pre-trained T2I SD model can be triggered, resulting in richer synthesized details, which however can reduce the full-reference indices. A visual example is shown in Figure 4. We see that while LLaVA extracts significantly longer text prompts than DAPE, they produce a similar amount of visual details. However, it is worth mentioning that the MLLM model LLaVA is very costly, increasing the inference time of DAPE by 170 times. Considering the cost-effectiveness, we ultimately choose DAPE as the text prompt extractor in OSEDiff.

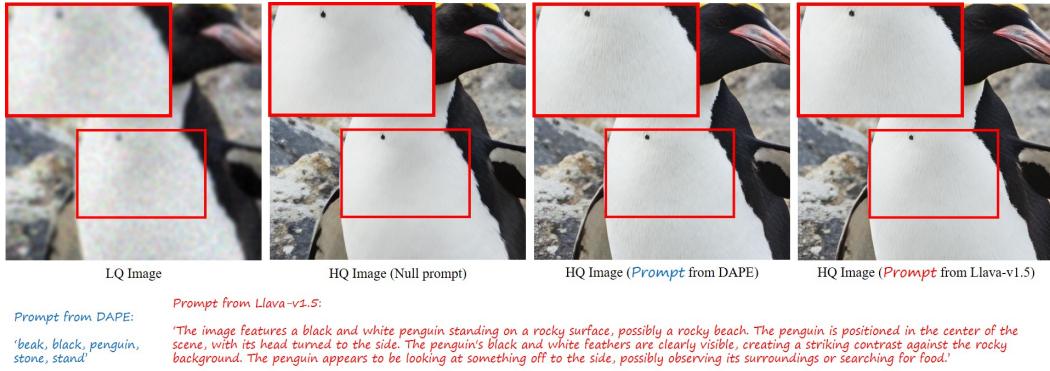


Figure 4: The impact of different prompt extraction methods. Please zoom in for a better view.

Setting of LoRA Rank. When finetuning the VAE encoder and the UNet, we need to set the rank of LoRA layers. Here we evaluate the effect of different LoRA ranks on the Real-ISR performance by using the RealSR benchmark [3]. The results are shown in Tables 5 and 6, respectively. As shown in Table 5, if a too small LoRA rank, such as 2, is set for the VAE encoder, the training will be unstable and cannot converge. On the other hand, if a higher LoRA rank, such as 8, is used for the VAE encoder, it may overfit in estimating image degradation, losing some image details in the output, as evidenced by the PSNR, DISTS, MUSIQ and NIQE indices. We find that setting the rank to 4 can achieve a balanced result for the VAE encoder. Similar conclusions can be made for the setting of LoRA rank on UNet. As can be seen from Table 6, a rank of 4 strikes a good balance. Therefore, we set the rank as 4 for both the VAE encoder and UNet LoRA layers.

The Finetuning on the VAE Encoder and Decoder. We conducted ablation studies to examine the impact of finetuning the VAE encoder and decoder, as shown Table 7. In the first row, where neither the VAE encoder nor decoder is finetuned, the results show poor perception performance. Comparing with OSEDiff, where only the VAE encoder is finetuned, we observe significant improvements in perceptual quality (e.g., MUSIQ improves from 58.99 to 69.09). This demonstrates that finetuning the VAE encoder is important for removing degradation and enhancing overall performance. When comparing the third row, where both the VAE encoder and decoder are finetuned, with OSEDiff, where only the encoder is trained and the decoder is fixed, we note that OSEDiff also achieves better perceptual quality (CLIPiQ improves from 0.5778 to 0.6693). This indicates that fixing the VAE decoder is important to ensure that the UNet output remains in the original VAE latent space, which helps minimizing the VSD loss more effectively. Thus, finetuning the VAE encoder is important to remove degradation, while fixing the VAE decoder helps maintaining stability in the latent space, leading to better perceptual quality.

5 Conclusion and Limitation

We proposed OSEDiff, a one-step effective diffusion network for Real-ISR, by utilizing the pre-trained text-to-image model as both the generator and the regularizer in training. Unlike traditional multi-step diffusion models, OSEDiff directly took the given LQ image as the starting point for diffusion, eliminating the uncertainty associated with random noise. By fine-tuning the pre-trained diffusion network with trainable LoRA layers, OSEDiff can well adapt to the complex real-world image degradations. Meanwhile, we performed the variational score distillation in the latent space to ensure that the model's predicted scores align with those of multi-step pre-trained models, enabling OSEDiff to efficiently produce HQ images in one diffusion step. Our experiments showed that OSEDiff achieved comparable or superior Real-ISR outcomes to previous multi-step diffusion-based methods in both objective metrics and subjective assessments. We believe our exploration can facilitate the practical application of pre-trained T2I models to Real-ISR tasks.

There are some limitations of OSEDiff. First, the details generation capability of OSEDiff can be further improved. Second, like other SD-based methods, OSEDiff is limited in reconstructing fine-scale structures such as small scene texts. We will investigate these problems in further work.

References

- [1] Stability.ai. <https://stability.ai/stable-diffusion>.
- [2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017.
- [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.
- [4] Chaofeng Chen, Xinyu Shi, Yipeng Qin, Xiaoming Li, Xiaoguang Han, Tao Yang, and Shihui Guo. Real-world blind super-resolution via feature matching with implicit high-resolution priors. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1329–1338, 2022.
- [5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12299–12310, 2021.
- [6] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22367–22377, 2023.
- [7] Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xiaokang Yang, and Fisher Yu. Dual aggregation transformer for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12312–12321, 2023.
- [8] Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158, 1975.
- [9] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11065–11074, 2019.
- [10] Trung Dao, Thuan Hoang Nguyen, Thanh Le, Duc Vu, Khoi Nguyen, Cuong Pham, and Anh Tran. Swift-brush v2: Make your one-step diffusion model better than its teacher. *arXiv preprint arXiv:2408.14176*, 2024.
- [11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [12] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.
- [13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014.
- [14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [15] 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.
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[19] 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.

[20] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022.

[21] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021.

[22] 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.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[25] Kyungmin Lee, Kihyuk Sohn, and Jinwoo Shin. Dreamflow: High-quality text-to-3d generation by approximating probability flow. *arXiv preprint arXiv:2403.14966*, 2024.

[26] 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.

[27] Jie Liang, Hui Zeng, and Lei Zhang. Details or artifacts: A locally discriminative learning approach to realistic image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5657–5666, 2022.

[28] Jie Liang, Hui Zeng, and Lei Zhang. Efficient and degradation-adaptive network for real-world image super-resolution. In *European Conference on Computer Vision*, pages 574–591. Springer, 2022.

[29] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021.

[30] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[31] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Ben Fei, Bo Dai, Wanli Ouyang, Yu Qiao, and Chao Dong. Diffbir: Towards blind image restoration with generative diffusion prior. *arXiv preprint arXiv:2308.15070*, 2023.

[32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[34] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023.

[35] Gaurav Parmar, Taesung Park, Srinivasa Narasimhan, and Jun-Yan Zhu. One-step image translation with text-to-image models. *arXiv preprint arXiv:2403.12036*, 2024.

[36] 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.

[37] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[39] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[40] Lingchen Sun, Rongyuan Wu, Zhengqiang Zhang, Hongwei Yong, and Lei Zhang. Improving the stability of diffusion models for content consistent super-resolution. *arXiv preprint arXiv:2401.00877*, 2023.

[41] 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*, volume 37, pages 2555–2563, 2023.

[42] 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*, pages 1–21, 2024.

[43] Peihao Wang, Dejia Xu, Zhiwen Fan, Dilin Wang, Sreyas Mohan, Forrest Iandola, Rakesh Ranjan, Yilei Li, Qiang Liu, Zhangyang Wang, et al. Taming mode collapse in score distillation for text-to-3d generation. *arXiv preprint arXiv:2401.00909*, 2023.

[44] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9168–9178, 2021.

[45] 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.

[46] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[47] Yinhui Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022.

[48] 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. *arXiv preprint arXiv:2311.14760*, 2023.

[49] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

[50] 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.

[51] 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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 101–117. Springer, 2020.

[52] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. *arXiv preprint arXiv:2311.16518*, 2023.

[53] Liangbin Xie, Xintao Wang, Xiangyu Chen, Gen Li, Ying Shan, Jiantao Zhou, and Chao Dong. Desra: Detect and delete the artifacts of gan-based real-world super-resolution models. 2023.

[54] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[55] 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.

[56] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 672–681, 2021.

- [57] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization. *arXiv preprint arXiv:2308.14469*, 2023.
- [58] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024.
- [59] Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild. *arXiv preprint arXiv:2401.13627*, 2024.
- [60] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *arXiv preprint arXiv:2307.12348*, 2023.
- [61] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4791–4800, 2021.
- [62] Lin Zhang, Lei Zhang, and Alan C Bovik. A feature-enriched completely blind image quality evaluator. *IEEE Transactions on Image Processing*, 24(8):2579–2591, 2015.
- [63] 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.
- [64] 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.
- [65] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *European Conference on Computer Vision*, pages 649–667. Springer, 2022.
- [66] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.
- [67] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

A Appendix

In the appendix, we provide the following materials:

- Comparison with GAN-based methods (referring to Section 4.1 in the main paper).
- Results of user study (referring to Section 4.1 in the main paper).
- More real-world visual comparisons under scaling factor $4\times$ (referring to Section 4.2 in the main paper).
- Training algorithm of OSEDiff (referring to Section 3.2 in the main paper).

A.1 Comparison with GAN-based Methods

We compare OSEDiff with four representative GAN-based Real-ISR methods, including BSRGAN [61], Real-ESRGAN [45], LDL [27] and FeMaSR [4]. The results are shown in Table 8. It is not a surprise that GAN-based methods have better fidelity measures such as PSNR and SSIM than OSEDiff. However, OSEDiff has much better perceptual quality metrics. We also provide visual comparisons in Figure 5. Compared to GAN-based methods, OSEDiff is able to generate realistic and reasonable details, such as squirrel hair, textures of petals, buildings, and leaves.

A.2 User Study

To further validate the effectiveness of our proposed OSEDiff method, we conducted a user study by using 20 real-world LQ images. An LQ image and its HQ counterparts generated by different Real-ISR methods were presented to volunteers, who were asked to select the best HQ result. The volunteers were instructed to consider two factors when making their decisions: the image perceptual quality and and its content (including structure and texture) consistency with the LQ input, with each factor contributing equally to the final selection.

We randomly selected 20 real-world LQ images from the RealLR200 dataset [52]. Figure 6(a) shows the thumbnails used in the user study, cropped into squares for a convenient layout. We generated the HQ outputs of them by using the DM-based Real-ISR methods StableSR [42], DiffBIR [31], SeeSR [52], PASD [57], ResShift [60], SinSR [48], and OSEDiff. A number of 15 volunteers were invited to participate in the evaluation. The results are shown in Figure 6(b). We see that OSEDiff ranks the second, just lagging slightly behind SeeSR. However, it should be noted that OSEDiff runs over 10 times faster than SeeSR by performing only one-step diffusion.

A.3 More Visual Comparisons

Figure 7 provides more visual comparisons between OSEDiff and other DM-based methods. One can see that OSEDiff achieves comparable to or even better results than the multi-step diffusion methods in scenarios such as portraits, flower patterns, buildings, animal fur, and letters.

Table 8: Quantitative comparison with GAN-based methods on both synthetic and real-world benchmarks. The best results of each metric are highlighted in red.

Datasets	Methods	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	DISTS \downarrow	FID \downarrow	NIQE \downarrow	MUSIQ \uparrow	MANIQA \uparrow	CLIPQA \uparrow
DIV2K-Val	BSRGAN	24.58	0.6269	0.3351	0.2275	44.23	4.7518	61.20	0.5071	0.5247
	Real-ESRGAN	24.29	0.6371	0.3112	0.2141	37.64	4.6786	61.06	0.5501	0.5277
	LDL	23.83	0.6344	0.3256	0.2227	42.29	4.8554	60.04	0.5350	0.5180
	FeMaSR	23.06	0.5887	0.3126	0.2057	35.87	4.7410	60.83	0.5074	0.5997
DrealSR	OSEDiff	23.72	0.6108	0.2941	0.1976	26.32	4.7097	67.97	0.6148	0.6683
	BSRGAN	28.75	0.8031	0.2883	0.2142	155.63	6.5192	57.14	0.4878	0.4915
	Real-ESRGAN	28.64	0.8053	0.2847	0.2089	147.62	6.6928	54.18	0.4907	0.4422
	LDL	28.21	0.8126	0.2815	0.2132	155.53	7.1298	53.85	0.4914	0.4310
	FeMaSR	26.90	0.7572	0.3169	0.2235	157.78	5.9073	53.74	0.4420	0.5464
RealSR	OSEDiff	27.92	0.7835	0.2968	0.2165	135.30	6.4902	64.66	0.5899	0.6963
	BSRGAN	26.39	0.7654	0.2670	0.2121	141.28	5.6567	63.21	0.5399	0.5001
	Real-ESRGAN	25.69	0.7616	0.2727	0.2063	135.18	5.8295	60.18	0.5487	0.4449
	LDL	25.28	0.7567	0.2766	0.2121	142.71	6.0024	60.82	0.5485	0.4477
	FeMaSR	25.07	0.7358	0.2942	0.2288	141.05	5.7885	58.95	0.4865	0.5270
RealSR	OSEDiff	25.15	0.7341	0.2921	0.2128	123.49	5.6476	69.09	0.6326	0.6693

A.4 Algorithm of OSEDiff

The pseudo-code of our OSEDiff training algorithm is summarized as **Algorithm 1**. We follow [49, 58] and use classifier-free guidance (cfg) when calculating z_ϕ . The cfg value is set to 7.5, and the negative prompts we use are: "painting, oil painting, illustration, drawing, art, sketch, oil painting, cartoon, CG Style, 3D render, unreal engine, blurring, dirty, messy, worst quality, low quality, frames, watermark, signature, jpeg artifacts, deformed, lowres, over-smooth."

Algorithm 1 Training Scheme of OSEDiff

Input: Training dataset \mathcal{S} , pretrained SD parameterized by ϕ including VAE encoder E_ϕ , latent diffusion network E_ϕ and VAE decoder ϵ_ϕ , prompt extractor Y , training iteration N

- 1 Initialize G_θ parameterized by θ , including
 $E_\theta \leftarrow E_\phi$ with trainable LoRA
 $\epsilon_\theta \leftarrow \epsilon_\phi$ with trainable LoRA
 $D_\theta \leftarrow D_\phi$
- 2 Initialize $\epsilon_\phi \leftarrow \epsilon_\theta$ with trainable LoRA
- 3 **for** $i \leftarrow 1$ **to** N **do**
 - 4 Sample $\mathbf{x}_L, \mathbf{x}_H$ from \mathcal{S}
 $\text{/* Network forward}$ $*/$
 - 5 $c_y \leftarrow Y(\mathbf{x}_L)$
 - 6 $\mathbf{z}_L \leftarrow E_\theta(\mathbf{x}_L)$
 - 7 $\hat{\mathbf{z}}_H \leftarrow F_\theta(\mathbf{z}_L; c_y)$
 - 8 $\hat{\mathbf{x}}_H \leftarrow D_\theta(\hat{\mathbf{z}}_H)$
 $\text{/* Compute data term objective}$ $*/$
 - 9 $\nabla_\theta \mathcal{L}_{\text{data}} \leftarrow [\mathcal{L}_{\text{MSE}}(\hat{\mathbf{x}}_H, \mathbf{x}_H) + \lambda_1 \mathcal{L}_{\text{LPIPS}}(\hat{\mathbf{x}}_H, \mathbf{x}_H)] \frac{\partial \hat{\mathbf{x}}_H}{\partial \theta}$
 $\text{/* Compute regularization objective, following DMD [58]}$ $*/$
 - 10 Sample ϵ from $\mathcal{N}(0, \mathbf{I})$
 - 11 Sample t from $\{20, \dots, 980\}$
 - 12 $\hat{\mathbf{z}}_t \leftarrow \alpha_t \hat{\mathbf{z}}_H + \sigma_t \epsilon$
 - 13 $\mathbf{z}_\phi \leftarrow \text{stopgrad}(F_\phi(\hat{\mathbf{z}}_t; c_y))$
 - 14 $\mathbf{z}_{\phi'} \leftarrow \text{stopgrad}(F_{\phi'}(\hat{\mathbf{z}}_t; c_y))$
 - 15 $\omega \leftarrow 1/\text{mean}(\|\mathbf{z}_\phi - \hat{\mathbf{z}}_H\|)$
 - 16 $\nabla_\theta \mathcal{L}_{\text{reg}} \leftarrow [\omega(\mathbf{z}_{\phi'} - \mathbf{z}_\phi)] \frac{\partial \hat{\mathbf{z}}_H}{\partial \theta}$
 $\text{/* Compute regularizer funetuning objective}$ $*/$
 - 17 Sample ϵ from $\mathcal{N}(0, \mathbf{I})$
 - 18 Sample t from $\{1, \dots, T\}$
 - 19 $\mathbf{z}_t \leftarrow \alpha_t \text{stopgrad}(\hat{\mathbf{z}}_H) + \sigma_t \epsilon$
 - 20 $\mathcal{L}_{\text{diff}} \leftarrow \mathcal{L}_{\text{MSE}}(\epsilon_{\phi'}(\mathbf{z}_t; t, c_y), \epsilon)$
 $\text{/* Network Parameter Update}$ $*/$
 - 21 Update θ with $\mathcal{L}_{\text{data}} + \lambda_2 \mathcal{L}_{\text{reg}}$
 - 22 Update ϕ' with $\mathcal{L}_{\text{diff}}$
- 23 **end**

Output: Generator G_θ including VAE encoder E_θ , latent diffusion network E_θ and VAE decoder ϵ_θ

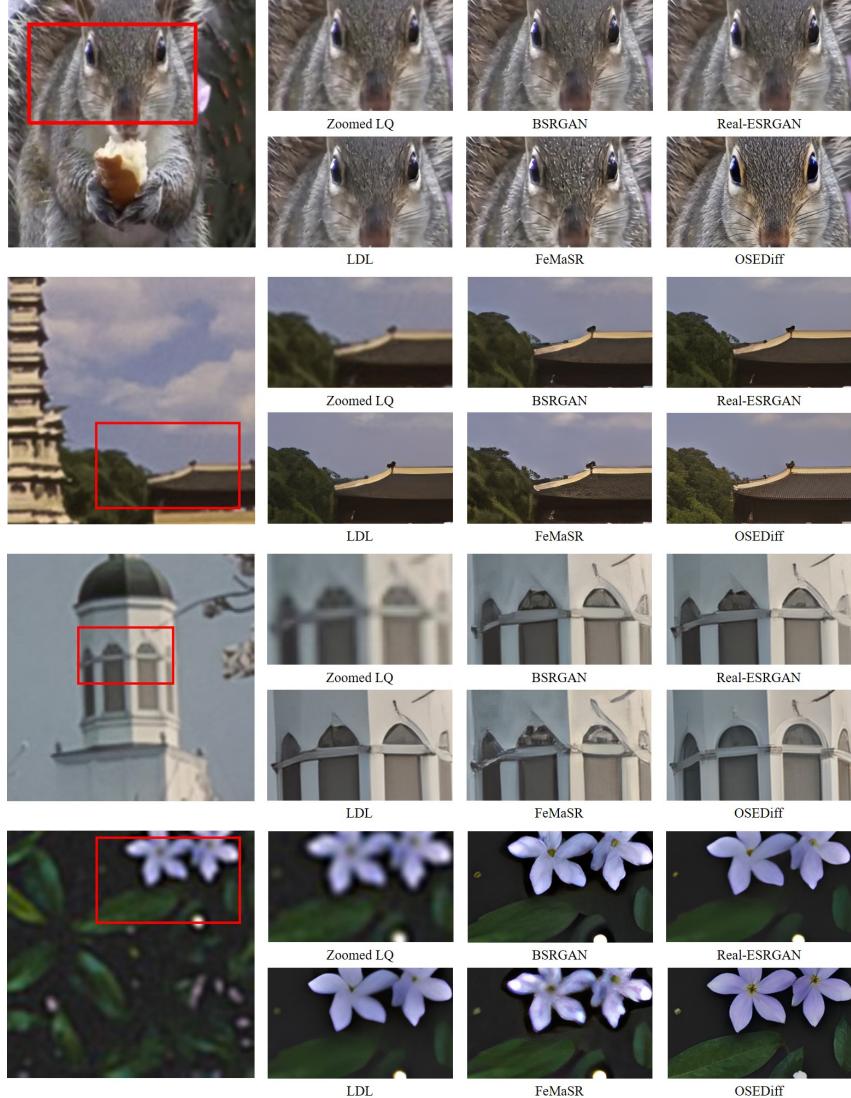
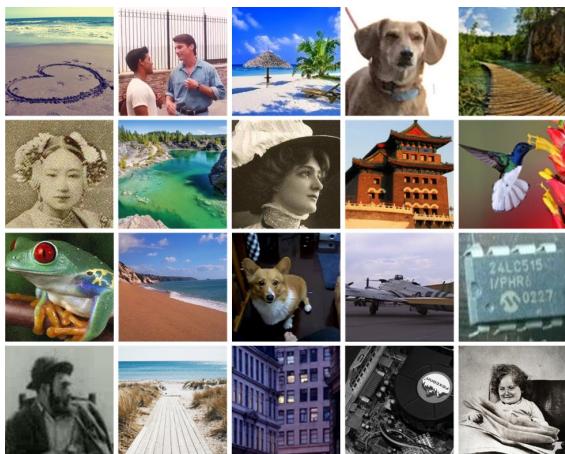
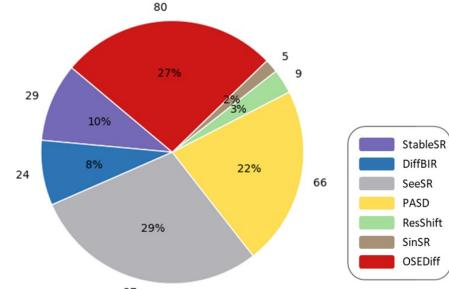


Figure 5: Qualitative comparisons between OSEDiff and GAN-based Real-ISR methods. Please zoom in for a better view.



(a) The LQ images used in user study



(b) The voting results are from 15 volunteers. The corresponding percentages and numerical counts are displayed with the pie chart.

Figure 6: The LQ images used in user study and the voting results.

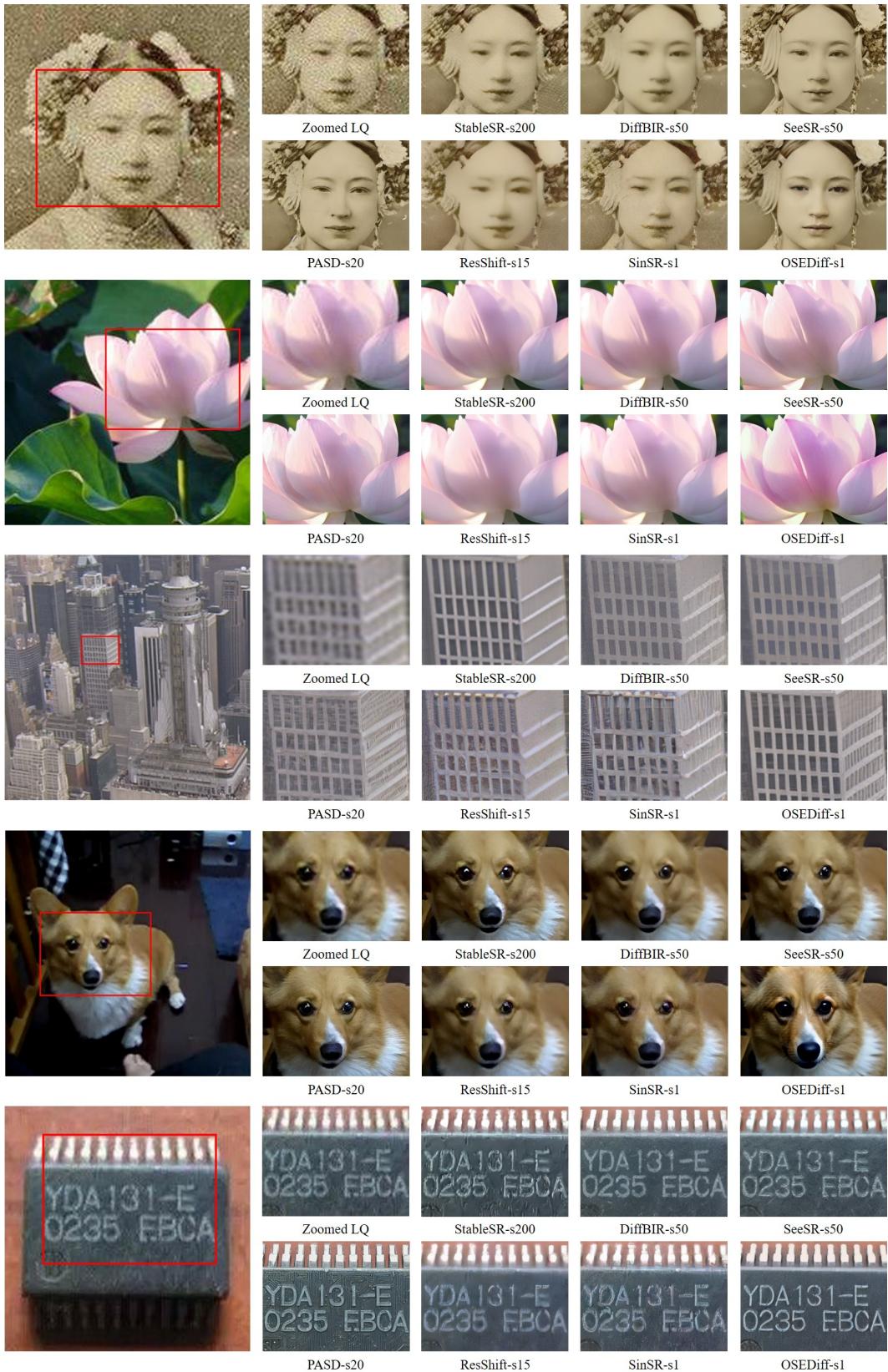


Figure 7: More visualization comparisons of different DM-based Real-ISR methods. Please zoom in for a better view.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: Yes, the main claims in the abstract and introduction accurately reflect the paper's contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss limitations in the final section (Conclusion and Limitation).

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Yes, the paper provides a full set of assumptions and a proof for each theoretical result.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, the paper fully discloses all necessary information to reproduce the main experimental results, supporting the main claims and conclusions.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: All training and testing data are publicly available. We will release the codes and model if the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Yes, the paper provides all necessary training and test details, including data splits, hyperparameters, and optimizer type, to fully understand the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[No\]](#)

Justification: We do not report error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: We detail the type of compute workers, memory, and execution time in the experiment section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: Yes, the research in the paper fully conforms to the NeurIPS Code of Ethics in every respect.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: Yes, we analyze the potential social impact in the final section.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, we clearly indicate the baseline methods and training/testing data used in the paper. Their licenses permit use within academic scope.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: Codes and model will be released if the paper is accepted.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowd sourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

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

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.