LPNSR: LEARNABLE NOISE PREDICTION FOR DIFFUSION-BASED IMAGE SUPER-RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion-based image super-resolution (SR) aims to reconstruct high-resolution (HR) images from low-resolution (LR) observations. A key property of diffusion models is that, once the starting point of the reverse chain and the denoising network are fixed, the quality of the generated image is dictated by the noise maps sampled at the intermediate steps, which are drawn from an unconstrained standard Gaussian distribution in conventional pipelines. This property raises a natural question: does there exist a noise sampling distribution better than the standard Gaussian that improves the quality of the generated images? To this end, we use a parameterized deep neural network to predict the mean and the variance of the Gaussian noise sampling distribution at each intermediate step, and design two training schemes, one supervised by the quality of the final generated image and the other by aligning each reverse step with the forward posterior. Experiments show that, compared with the standard Gaussian distribution, the learnable noise sampling distribution improves the quality of the generated images.

1 INTRODUCTION

Image super-resolution (SR) aims to reconstruct a high-resolution (HR) image from its low-resolution (LR) observation, a severely ill-posed problem since the degradation is unknown and multiple HR images are consistent with the same LR input. Since SRCNN (Dong et al., 2016) first formulated SR as regression with a deep network, learning-based methods have dominated this task. Early regression-based methods (Kim et al., 2016; Wang et al., 2015; Ahn et al., 2018) steadily improved reconstruction fidelity, yet they inherently yield over-smoothed results, since pixel-wise objectives push the prediction toward the conditional mean over plausible HR images (Blau & Michaeli, 2018). GAN-based methods (Ledig et al., 2017; Sajjadi et al., 2017; Menon et al., 2020) instead recover realistic textures with adversarial and perceptual losses, and their successors improve the generator design and the degradation modeling for real-world inputs (Wang et al., 2018; Zhang et al., 2021; Wang et al., 2021a), but adversarial training remains unstable and may hallucinate unfaithful textures. More recently, diffusion models (Ho et al., 2020; Song et al., 2020; Rombach et al., 2022) have become the dominant paradigm for SR, generating the HR image by an iterative denoising process that is conditioned on the LR input (Saharia et al., 2023) or guided by it during sampling (Choi et al., 2021; Chung et al., 2022b; Kawar et al., 2022; Chung et al., 2022a).

A key property of diffusion models is that, once the starting point of the reverse chain and the denoising network are fixed, the noise maps sampled at the intermediate steps are the only degree of freedom of the generated image, and the quality of the generated image is dictated by them (see Fig. 1). In conventional pipelines, however, these noise maps are drawn from a fixed standard Gaussian distribution. This raises a natural question: does there exist a noise sampling distribution better than the standard Gaussian that improves the quality of the generated images?

A closely related idea is diffusion inversion, which seeks the noise maps with which a pretrained diffusion model reconstructs a given target image, and has been developed mainly for image editing and personalization. Early works optimize the conditioning signal used by the sampler, such as the textual or visual embedding (Gal et al., 2022; Mokady et al., 2023; Miyake et al., 2025; Nguyen et al., 2023), whereas later works operate directly in the noise space, correcting the approximation error of the inversion through coupled transformations, fixed-point iterations, or re-designed injection schedules (Wallace et al., 2023; Meiri et al., 2023; Ju et al., 2023; Kang et al., 2024). In parallel,

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

diffusion models are widely used as priors for inverse problems, where the intermediate latents are guided toward the observation during sampling (Choi et al., 2021; Chung et al., 2022b; Kawar et al., 2022; Chung et al., 2022a; Fei et al., 2023; Song et al., 2023). Recently, this idea has also been extended to SR: InvSR (Yue et al., 2025) trains a network to predict a noise map that serves as the starting point of a frozen text-to-image diffusion model.

In contrast to these methods, which determine the injected noise either by per-input optimization or by predicting a single noise map that only serves as the starting point of a frozen model, we learn the noise sampling distribution at each intermediate step as a conditional distribution that can be evaluated in a single forward pass, parameterized as a diagonal Gaussian whose mean and variance are predicted by a deep neural network conditioned on the current latent, the clean-image estimate of the frozen denoiser, the low-resolution latent, and the time step. We then design two training schemes for this noise predictor. The first is an end-to-end supervised training: the predictor is plugged into every step of the reverse chain, the chain is unrolled, and the predictor is optimized by supervising the final generated image with pixel-wise, perceptual (Zhang et al., 2018), and adversarial (Goodfellow et al., 2014) losses. The second trains the predictor step by step without unrolling the chain: aligning the model reverse distribution with the forward posterior, the network regresses the prediction residual of the frozen denoiser, i.e., the gap between the ground-truth clean latent and its denoised estimate, which amounts to a mean-correction of the reverse distribution.

The main contributions of this work are summarized as follows. First, we model the injected noise as a learnable conditional distribution, and propose two training schemes, i.e., the end-to-end supervised training and the mean-correction training. Second, we propose LPNSR (Learnable Prediction of Noise for Super-Resolution) on the residual-shifting diffusion paradigm (Yue et al., 2023), where a multi-input-aware noise predictor estimates the mean and the variance of the injected noise at each sampling step, while the pretrained denoising network and the inference mechanism remain untouched. Third, we conduct experiments and show that the learnable noise sampling distribution improves the quality of the generated images over the standard Gaussian distribution.

2 Related work

Image Super-Resolution. Deep learning has become the dominant paradigm for SR (Dong et al., 2016; Wang et al., 2021b). Early works trained regression models on paired LR-HR data (Ahn et al., 2018; Kim et al., 2016; Wang et al., 2015); while these models capture the expectation of the posterior distribution, they suffer from over-smoothing artifacts (Ledig et al., 2017; Menon et al., 2020; Sajjadi et al., 2017). To improve the perceptual quality, generative SR models have drawn growing interest, including autoregressive architectures (Dahl et al., 2017; Menick & Kalchbrenner, 2018; Van den Oord et al., 2016; Parmar et al., 2018), which incur substantial computational overhead. GAN-based methods attain remarkable perceptual quality (Ledig et al., 2017; Sajjadi et al., 2017; Menon et al., 2020; Wang et al., 2018; 2021a), yet their training remains unstable. More recently, diffusion models have been widely adopted for SR (Choi et al., 2021; Chung et al., 2022b; Kawar et al., 2022; Rombach et al., 2022; Saharia et al., 2023), either by concatenating the LR image to the denoiser's input (Rombach et al., 2022; Saharia et al., 2023) or by adapting the reverse process of a pretrained model (Choi et al., 2021; Chung et al., 2022b; Kawar et al., 2022). Yet they inject unconstrained Gaussian noise at each reverse step rather than meaningful noise maps.

Diffusion Inversion. This paradigm aims to find the noise maps with which a pretrained diffusion model reconstructs a given target image, and has been developed mainly for image editing and personalization. Early works optimized text embeddings for better alignment (Gal et al., 2022; Mokady et al., 2023), and follow-up works further refined inversion via textual/visual prompts (Miyake et al., 2025; Nguyen et al., 2023) or intermediate noise map optimization (Ju et al., 2023; Kang et al., 2024; Meiri et al., 2023; Wallace et al., 2023). However, existing methods determine the noise either by per-input optimization or by predicting a single noise map per input, and none of them attempts to formulate the noise sampling distribution at each intermediate step as a parametric distribution. For SR, InvSR (Yue et al., 2025) trains a noise predictor that produces the starting point of a frozen text-to-image diffusion model, while the noise maps injected at the intermediate steps are still drawn from a standard Gaussian distribution, as in conventional pipelines.



Figure 1: Qualitative comparison of SR results generated with different random seeds. (Zoom in for best view)

3 METHODOLOGY

3.1 MOTIVATION

We first briefly review the residual-shifting diffusion framework (ResShift) (Yue et al., 2023), which underlies our method. Let $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ denote the latent codes of the HR and LR images encoded by the pretrained VQGAN (Esser et al., 2021), and let $\boldsymbol{e}_0 = \boldsymbol{y}_0 - \boldsymbol{x}_0$ denote their residual. Let $\{\eta_t\}_{t=1}^T$ be a monotonically increasing shifting sequence with $\eta_1 \to 0$ and $\eta_T \to 1$.

Forward process. ResShift constructs a Markov chain of length T that gradually shifts the residual $\boldsymbol{e}_0$, whose transition distribution is formulated as

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_0) = \mathcal{N}\left(\boldsymbol{x}_t;\ \boldsymbol{x}_{t-1} + \alpha_t\, \boldsymbol{e}_0,\ \kappa^2 \alpha_t\, \boldsymbol{I}\right), \tag{1}$$

where $\alpha_t = \eta_t - \eta_{t-1}$ for $t > 1$ and $\alpha_1 = \eta_1$, and the hyperparameter κ controls the noise variance: each step shifts away an α_t portion of the residual while injecting $\kappa^2 \alpha_t$ of Gaussian noise. Notably, the marginal distribution at any time step is analytically integrable:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{y}_0) = \mathcal{N}\left(\boldsymbol{x}_t;\ \boldsymbol{x}_0 + \eta_t\, \boldsymbol{e}_0,\ \kappa^2 \eta_t\, \boldsymbol{I}\right), \tag{2}$$

namely the noisy sample deterministically interpolates from $\boldsymbol{x}_0$ toward $\boldsymbol{y}_0$ with a monotonically increasing noise level.

Reverse process. The reverse process starts from $p(\boldsymbol{x}_T \mid \boldsymbol{y}_0) \approx \mathcal{N}\left(\boldsymbol{x}_T;\ \boldsymbol{y}_0,\ \kappa^2 \boldsymbol{I}\right)$ and iteratively recovers $\boldsymbol{x}_0$ from $\boldsymbol{x}_t$ under the guidance of $\boldsymbol{y}_0$, with the inverse transition kernel assumed to be Gaussian:

$$p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}_0) = \mathcal{N}\left(\boldsymbol{x}_{t-1};\ \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t),\ \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t)\right). \tag{3}$$

Training objective. The optimization for θ is achieved by minimizing the Kullback-Leibler (KL) divergence between the forward posterior and the model reverse distribution, i.e., the negative evidence lower bound:

$$\min_\theta \sum_t D_{KL}\left[\, q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}_0) \,\middle\|\, p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}_0)\,\right], \tag{4}$$

where the targeted distribution is rendered tractable by combining the forward transition and the marginal in equation 2:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}_0) = \mathcal{N}\left(\boldsymbol{x}_{t-1};\ \frac{\eta_{t-1}}{\eta_t}\, \boldsymbol{x}_t + \frac{\alpha_t}{\eta_t}\, \boldsymbol{x}_0,\ \kappa^2 \frac{\eta_{t-1}}{\eta_t}\, \alpha_t\, \boldsymbol{I}\right). \tag{5}$$

Considering that the variance parameter is independent of $\boldsymbol{x}_t$ and $\boldsymbol{y}_0$, it is set to $\boldsymbol{\Sigma}_\theta = \Sigma_\theta\, \boldsymbol{I} = \kappa^2 \frac{\eta_{t-1}}{\eta_t}\, \alpha_t\, \boldsymbol{I}$, and the mean parameter is reparameterized as

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t) = \frac{\eta_{t-1}}{\eta_t}\, \boldsymbol{x}_t + \frac{\alpha_t}{\eta_t}\, \boldsymbol{f}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t), \tag{6}$$

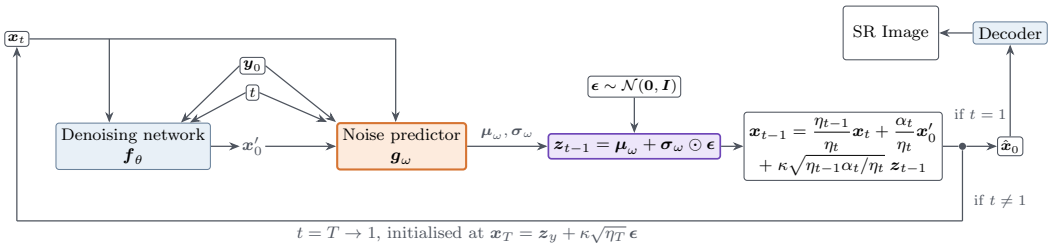


Figure 2: Overview of the LPNSR inference framework. The noise predictor $\boldsymbol{g}_\omega$ outputs a diagonal Gaussian $(\boldsymbol{\mu}_\omega, \boldsymbol{\sigma}_\omega)$ whose sample $\boldsymbol{z}_{t-1}$ replaces the unconstrained Gaussian noise in the reverse iteration of equation 7. The reverse process runs from $t = T$ to $t = 1$, and the final estimate is decoded by the VQGAN decoder into the SR image.

where $\boldsymbol{f}_\theta$ is a deep neural network aiming to predict $\boldsymbol{x}_0$. The KL objective then simplifies to the regression form $\min_\theta \sum_t w_t \left\| \boldsymbol{f}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t) - \boldsymbol{x}_0 \right\|_2^2$, where $w_t = \frac{\alpha_t}{2\kappa^2 \eta_t \eta_{t-1}}$ is the step-dependent weight.

Reverse sampling. At inference, the reverse process is instantiated as the stochastic iteration:

$$\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t) + \sqrt{\Sigma_\theta}\, \boldsymbol{z}_{t-1} = \frac{\eta_{t-1}}{\eta_t}\, \boldsymbol{x}_t + \frac{\alpha_t}{\eta_t}\, \boldsymbol{x}'_0 + \kappa \sqrt{\frac{\eta_{t-1}\, \alpha_t}{\eta_t}}\, \boldsymbol{z}_{t-1}, \tag{7}$$

where $\boldsymbol{x}'_0 = \boldsymbol{f}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t)$ is the predicted clean image, and $\boldsymbol{z}_{t-1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is the intermediate noise injected at step t. For $t = 1$, $\boldsymbol{z}_0 = \boldsymbol{0}$.

Our core observation is that, given the starting point $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{y}_0, \kappa^2 \boldsymbol{I})$, the quality of the generated image is dictated by the noise maps $\{\boldsymbol{z}_{t-1}\}_{t=2}^T$ sampled at the intermediate steps. To validate this observation, we fix the starting point $\boldsymbol{x}_T$ and perform inference with different random seeds, so that only the intermediate noise maps $\{\boldsymbol{z}_{t-1}\}_{t=2}^T$ differ across runs. The results are presented in Fig. 1, where the generated images are evaluated by the MUSIQ score (Ke et al., 2021). It can be observed that the image quality varies considerably across different random seeds, indicating that the noise maps sampled at the intermediate steps directly affect the quality of the generated images.

With the above observation, a natural question arises: does there exist a noise sampling distribution better than $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ that can improve the quality of the generated images? We will analyze this question in Sec. 3.2.

3.2 Method

As revealed in Sec. 3.1, the noise maps sampled at the intermediate steps dictate the quality of the final generated image. Motivated by this observation, we parameterize the noise distribution at each intermediate step as a Gaussian distribution with diagonal covariance, whose mean and standard deviation are predicted element-wise by a learnable noise predictor $\boldsymbol{g}_\omega$:

$$\boldsymbol{z}_{t-1} = \boldsymbol{g}_\omega(\boldsymbol{x}_t, \boldsymbol{x}'_0, \boldsymbol{y}_0, t) = \boldsymbol{\mu}_\omega(\boldsymbol{x}_t, \boldsymbol{x}'_0, \boldsymbol{y}_0, t) + \boldsymbol{\sigma}_\omega(\boldsymbol{x}_t, \boldsymbol{x}'_0, \boldsymbol{y}_0, t) \odot \boldsymbol{\varepsilon}_{t-1}, \qquad \boldsymbol{\varepsilon}_{t-1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{8}$$

where $\boldsymbol{x}'_0 = \boldsymbol{f}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t)$ is the clean-image prediction of the frozen denoiser. In the following, we present the two training schemes of $\boldsymbol{g}_\omega$, together with a per-step optimality analysis that underlies them. The overall inference framework is illustrated in Fig. 2.

End-to-end supervised training. A natural idea following this observation is to train the noise predictor $\boldsymbol{g}_\omega$ with the quality of the final generated image as the supervision signal. To this end, $\boldsymbol{g}_\omega$ is injected into every intermediate step of the reverse chain, and the entire chain is unrolled end-to-end: starting from $\boldsymbol{x}_T = \boldsymbol{y}_0 + \kappa\, \boldsymbol{\epsilon}$, each reverse step computes $\boldsymbol{x}'_0 = \boldsymbol{f}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t)$ with the frozen denoiser, injects the noise predicted by $\boldsymbol{g}_\omega$, and proceeds via equation 7. Gradients from the final output are back-propagated through the whole unrolled chain to $\boldsymbol{g}_\omega$. Note that such training is computationally expensive, as it requires unrolling the entire reverse chain and back-propagating through every intermediate step, with both the memory and computation costs growing linearly with the number of sampling steps. The original ResShift framework (Yue et al., 2023) requires only four

inference steps, which makes this training feasible; however, it may not be feasible for diffusion frameworks with considerably more inference steps. The training objective directly supervises the quality of the final generated image:

$$\mathcal{L} = \mathcal{L}_2\big(\hat{\boldsymbol{x}}_0,\, \boldsymbol{x}_0\big) \;+\; \lambda_l\, \mathcal{L}_{\mathrm{LPIPS}}\big(\mathcal{D}(\hat{\boldsymbol{x}}_0),\, \boldsymbol{x}_0\big) \;+\; \lambda_g\, \mathcal{L}_{\mathrm{GAN}}\big(\hat{\boldsymbol{x}}_0,\, \boldsymbol{x}_0\big), \tag{9}$$

where $\hat{\boldsymbol{x}}_0$ denotes the final clean-image prediction produced at the last step of the unrolled chain, and $\mathcal{D}(\cdot)$ is the frozen VQGAN (Esser et al., 2021) decoder that maps latents back to the image space. Here $\mathcal{L}_2$ is the regression loss in the latent space that constrains the fidelity of the generated latent, $\mathcal{L}_{\mathrm{LPIPS}}$ (Zhang et al., 2018) is the perceptual loss that measures the perceptual similarity between the reconstructed and ground-truth images in the image space, and $\mathcal{L}_{\mathrm{GAN}}$ (Goodfellow et al., 2014) is the adversarial loss that distinguishes the generated latents from the real ones, encouraging realistic reconstructions. The complete training procedure is summarized in Alg. 1.

Mean-correction training. We characterize the per-step optimal form of the injected noise from the perspective of distribution alignment: the injected noise is per-step optimal if it minimizes the KL divergence between the model reverse distribution and the forward posterior. By the noise parameterization above and the reparameterization in equation 7, the model reverse distribution becomes $\mathcal{N}\left(\boldsymbol{x}_{t-1};\, \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t) + \sqrt{\Sigma_\theta}\,\boldsymbol{\mu}_\omega,\; \Sigma_\theta\,\mathrm{diag}(\boldsymbol{\sigma}_\omega^2)\right)$, while the forward posterior in equation 5 is Gaussian with mean $\tilde{\boldsymbol{\mu}}_t$ and covariance $\tilde{\boldsymbol{\Sigma}}_t$. The per-step optimal noise is thus obtained by minimizing this KL divergence:

$$(\boldsymbol{\mu}_\omega^*,\; \boldsymbol{\sigma}_\omega^*) = \arg\min_{\boldsymbol{\mu}_\omega,\, \boldsymbol{\sigma}_\omega} D_{KL}\big(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}_0) \;\big\|\; p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}_0)\big). \tag{10}$$

Since both distributions are Gaussian with diagonal covariance, minimizing this KL divergence simply amounts to aligning the mean and the variance of the two distributions. Aligning the means yields the optimal mean prediction (the mean term is a non-negative quadratic in $\boldsymbol{\mu}_\omega$, minimized exactly when the two means coincide):

$$\boldsymbol{\mu}_\omega^* = \frac{\tilde{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t)}{\sqrt{\Sigma_\theta}} = \frac{\alpha_t\,(\boldsymbol{x}_0 - \boldsymbol{x}_0')}{\eta_t\,\sqrt{\Sigma_\theta}}, \tag{11}$$

namely the optimal mean shift is precisely the normalized prediction residual of the denoising network. For the variance, since the reverse variance of ResShift (Yue et al., 2023) inherits the forward posterior variance, i.e., $\tilde{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_\theta$, setting $\boldsymbol{\sigma}_\omega^* = \mathbf{1}$ suffices for the variance alignment. Substituting the per-step optimal mean $\boldsymbol{\mu}_\omega^*$ and variance scale $\boldsymbol{\sigma}_\omega^* = \mathbf{1}$ back into the model reverse distribution yields

$$p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}_0) = \mathcal{N}\left(\boldsymbol{x}_{t-1};\, \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t) + \frac{\alpha_t}{\eta_t}(\boldsymbol{x}_0 - \boldsymbol{x}_0'),\, \Sigma_\theta\, \boldsymbol{I}\right). \tag{12}$$

This indicates that the per-step optimal injection is equivalent to rewriting the mean of p_θ—that is, adding a mean-correction term on top of the frozen prediction. Formally, we express the reverse process as

$$p_{\theta,\omega}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}_0) = \mathcal{N}\left(\boldsymbol{x}_{t-1};\; \boldsymbol{\mu}_{\theta,\omega}(\boldsymbol{x}_t, \boldsymbol{y}_0, t),\; \boldsymbol{\Sigma}_\theta\right), \tag{13}$$

where the mean parameter is reparameterized as

$$\boldsymbol{\mu}_{\theta,\omega}(\boldsymbol{x}_t, \boldsymbol{y}_0, t) = \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t) + \frac{\alpha_t}{\eta_t}\, \boldsymbol{f}_\omega(\boldsymbol{x}_t, \boldsymbol{x}_0', \boldsymbol{y}_0, t). \tag{14}$$

Here $\boldsymbol{f}_\omega$ denotes the residual predictor, and the mean term of the injected noise is given by $\boldsymbol{\mu}_\omega = \frac{\alpha_t}{\eta_t\sqrt{\Sigma_\theta}}\, \boldsymbol{f}_\omega$, i.e., the frozen mean $\boldsymbol{\mu}_\theta$ is corrected by the residual prediction of $\boldsymbol{f}_\omega$. The optimization for ω is achieved by minimizing the negative evidence lower bound, namely:

$$\min_\omega \sum_t D_{KL}\big(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}_0) \;\big\|\; p_{\theta,\omega}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}_0)\big). \tag{15}$$

The detailed mathematical derivation of this objective follows that of DDPM (Ho et al., 2020). Following ResShift (Yue et al., 2023), we transform the optimization objective in equation 15 into the following form:

$$\min_\omega \sum_t \big\| \boldsymbol{f}_\omega(\cdot) - (\boldsymbol{x}_0 - \boldsymbol{x}_0') \big\|_2^2 + \mathcal{L}_{\mathrm{LPIPS}}\big(\mathcal{D}(\boldsymbol{x}_0' + \boldsymbol{f}_\omega(\cdot)),\, \boldsymbol{x}_0\big), \tag{16}$$

where we incorporate the LPIPS perceptual loss (Zhang et al., 2018) as a regularization term. As suggested in ResShift (Yue et al., 2023), this term improves the perceptual quality of the generated images.

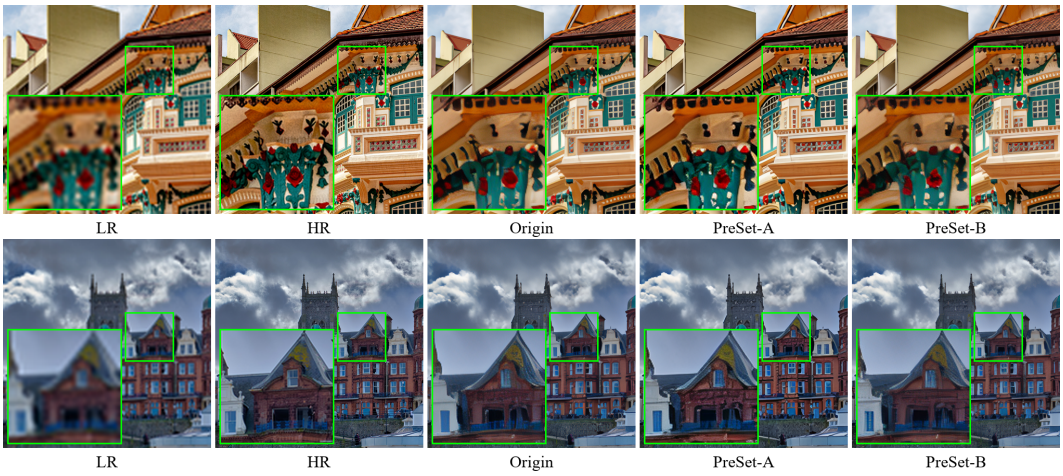


Figure 3: Qualitative comparison of the SR results generated under different noise sampling schemes. (Zoom in for best view)

Table 1: Quantitative comparison on the LSDIR and ImageNet datasets. The last three rows of each group report our noise sampling schemes (Origin, LPNSR-A, and LPNSR-B), and the remaining rows list several representative SR methods for reference. The best and second-best results of each metric are highlighted in red and blue, respectively.

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	MANIQA↑	CLIPIQA↑	FID↓
LSDIR	VOSR	20.04	0.4917	0.2448	0.1524	72.1925	0.5025	0.6303	47.79
	InvSR	19.19	0.4845	0.2657	0.1636	72.7310	0.5113	0.7156	69.30
	OSEDiff	20.03	0.5094	0.2675	0.1627	71.4740	0.4573	0.6946	58.98
	SinSR	20.39	0.5136	0.2281	0.1447	69.1867	0.4648	0.6687	40.82
	ResShift (Origin)	20.70	0.5281	0.2249	0.1420	67.4741	0.4363	0.6445	36.76
	LPNSR-A (Ours)	19.77	0.5140	0.2099	0.1336	71.7904	0.5618	0.7389	42.39
	LPNSR-B (Ours)	20.80	0.5345	0.2197	0.1394	68.3811	0.4582	0.6722	36.12
ImageNet	VOSR	24.98	0.6721	0.2445	0.1556	72.3900	0.4913	0.6327	45.64
	InvSR	23.98	0.6689	0.2527	0.1616	71.8737	0.4702	0.7152	57.64
	OSEDiff	24.67	0.6917	0.2462	0.1544	71.6051	0.4595	0.6789	50.32
	SinSR	26.53	0.7170	0.2251	0.1520	67.2132	0.4371	0.6655	41.24
	ResShift (Origin)	27.25	0.7397	0.2058	0.1501	65.5388	0.3928	0.6238	34.91
	LPNSR-A (Ours)	26.31	0.7337	0.2026	0.1384	69.9097	0.4897	0.7101	36.93
	LPNSR-B (Ours)	27.45	0.7430	0.2047	0.1481	66.6361	0.4095	0.6476	34.31

4 EXPERIMENTS

In this section, we conduct experiments to validate the role of the noise predictor and compare the performance of different schemes. For clarity, we denote the standard Gaussian distribution serving as the noise sampling distribution as Origin (ResShift), the noise predictor trained with the end-to-end supervised training scheme as PreSet-A (LPNSR-A), and the one trained with the mean-correction training scheme as PreSet-B (LPNSR-B). Following common practice in the field, our experiments mainly focus on the $\times 4$ SR task.

4.1 EXPERIMENTAL SETUP

Training Details. We collect images from public sources including LSDIR (Li et al., 2023), DIV2K (Agustsson & Timofte, 2017), and Flickr2K, as well as web images subjected to quality filtering, totaling 1M images. We train the noise predictor on the collected images for over 200k iterations, randomly cropping an image patch with a resolution of 256×256 from the source image and synthesizing the LR image using the pipeline of Real-ESRGAN (Wang et al., 2021a) at each iteration. We adopt the AdamW (Loshchilov & Hutter, 2017) optimizer with a learning rate of 5×10^{-5} and a batch size of 16. The hyperparameters for the loss function are set as $\lambda_l = 1.0$ and $\lambda_g = 0.1$. During training, we set $T = 4$ to remain consistent with ResShift (Yue et al., 2023), and the noise variance

Table 2: Statistics of the three noise sampling schemes on the LSDIR dataset. We randomly sample 100 random seeds; for each seed, every metric is averaged over the test images to obtain a per-seed mean, and the mean and the standard deviation of these per-seed means across the 100 seeds are reported as mean $\pm$ std.

Method	PSNR↑	LPIPS↓	DISTS↓	MUSIQ↑	CLIPIQA↑
Origin	20.6951 ± 0.0066	0.2239 ± 0.0009	0.1419 ± 0.0003	67.5117 ± 0.0617	0.6436 ± 0.0024
PreSet-A	19.7679 ± 0.0017	0.2103 ± 0.0002	0.1339 ± 0.0002	71.8248 ± 0.0269	0.7393 ± 0.0008
PreSet-B	20.7975 ± 0.0072	0.2195 ± 0.0007	0.1394 ± 0.0003	68.5293 ± 0.0640	0.6726 ± 0.0019

Table 3: Decomposition of the KL divergence between the forward posterior and the model reverse distribution at each intermediate step ($t = 4, 3, 2$); all values are computed per image and then averaged over the test images of the LSDIR dataset. The divergence is split into a mean term D_{KL}^{mean} and a variance term D_{KL}^{var}, where a smaller value indicates a better alignment, and $\bar{\sigma}_\omega^2$ denotes the mean injected noise variance, i.e., the mean of the diagonal entries of $\mathbf{\Sigma}_\omega$.

Method	$t=4$			$t=3$			$t=2$		
	D_{KL}^{mean}	D_{KL}^{var}	$\bar{\sigma}_\omega^2$	D_{KL}^{mean}	D_{KL}^{var}	$\bar{\sigma}_\omega^2$	D_{KL}^{mean}	D_{KL}^{var}	$\bar{\sigma}_\omega^2$
Origin	0.0611	0.0000	1.0000	0.1782	0.0000	1.0000	9.2725	0.0000	1.0000
PreSet-A	0.8988	5.8690	1.47×10^{-4}	0.4707	6.1599	6.97×10^{-5}	15.1685	5.4199	2.49×10^{-4}
PreSet-B	0.0577	0.0000	1.0000	0.1710	0.0000	1.0000	9.2190	0.0000	1.0000

hyperparameter $\kappa = 2.0$ as well as the shifting sequence $\{\eta_t\}_{t=1}^T$ also follow the identical settings. The denoising network $\boldsymbol{f}_\theta$ and the VQGAN autoencoder (Esser et al., 2021) are frozen during training, and only the noise predictor is optimized. The training is conducted on one NVIDIA RTX 5090 GPU. More detailed hyperparameter configurations are provided in the **Appendix**.

Testing Datasets and Metrics. For the testing datasets, we randomly sample 1000 images from the ImageNet test split (Deng et al., 2009), whose HR images are randomly cropped to 512×512 resolution and then degraded through the Real-ESRGAN pipeline (Wang et al., 2021a) to produce 128×128 LR images. For the LSDIR (Li et al., 2023) test split, we randomly crop the paired LR and HR images to 128×128 and 512×512, respectively. We evaluate our method together with the competing methods with full-reference and no-reference metrics. For distortion fidelity, we report PSNR and SSIM (Wang et al., 2004) on the Y channel of the YCbCr color space. For reference-based perceptual quality, we use LPIPS (Zhang et al., 2018), DISTS (Ding et al., 2022), and FID (Heusel et al., 2017). For no-reference perceptual quality, we report MUSIQ (Ke et al., 2021), MANIQA (Yang et al., 2022), and CLIPIQA (Wang et al., 2023).

Compared Methods. We compare our method with five representative diffusion-based SR approaches: ResShift (Yue et al., 2023), SinSR (Wang et al., 2024), OSEDiff (Wu et al., 2024), InvSR (Yue et al., 2025), and VOSR (Wu et al., 2026). Among them, OSEDiff and InvSR are built upon large-scale pretrained text-to-image diffusion models. All methods are run with their official default settings.

Network Architecture. For the noise predictor $\boldsymbol{g}_\omega$, we follow the default architecture and parameter settings of ResShift (Yue et al., 2023), i.e., the same UNet backbone in which the self-attention layers are replaced by Swin Transformer (Liu et al., 2021) blocks, so as to improve the robustness of the model to arbitrary image resolutions. For the adversarial loss, we adopt the classical PatchGAN discriminator (Isola et al., 2017) trained with the hinge loss (Lim & Ye, 2017).

4.2 Experimental results

Comparison with Other Methods. Tab. 1 compares our noise sampling schemes with the compared methods on the LSDIR and ImageNet datasets. Our LPNSR-B ranks first in PSNR, SSIM, and FID on both datasets, while LPNSR-A ranks first in LPIPS and DISTS on both datasets and additionally in MANIQA and CLIPIQA on LSDIR. The compared methods lead only on MUSIQ (InvSR on LSDIR and VOSR on ImageNet) and, on ImageNet, on MANIQA and CLIPIQA (VOSR and InvSR, respectively). Notably, our schemes achieve these results without any text-to-image pretraining. The qualitative comparison in Fig. 4 further corroborates these results. LPNSR-A and LPNSR-B restore

Table 4: Quantitative ablation studies on the loss functions used for training PreSet-A and PreSet-B, wherein the hyperparameters λ_l and λ_g control the weight importance of the LPIPS loss and the GAN loss, respectively. The results are evaluated on the LSDIR dataset under the 4-step sampling setting, with the last two columns measured at the intermediate step $t = 2$.

Method	Hyperparameters		Metrics						
	λ_l (LPIPS loss)	λ_g (GAN loss)	PSNR↑	LPIPS↓	DISTS↓	MUSIQ↑	CLIPIQA↑	D_{KL}^{mean}↓	$\bar{\sigma}_\omega^2$
A-Baseline1	0.0	0.0	22.05	0.2683	0.1592	64.2017	0.6184	8.8653	1.93×10^{-4}
A-Baseline2	1.0	0.0	20.35	0.2155	0.1372	70.4506	0.7180	12.3537	2.56×10^{-4}
A-Baseline3	0.0	0.1	19.28	0.2230	0.1428	73.0554	0.7495	17.0581	2.91×10^{-4}
B-Baseline1	0.0	\	21.55	0.2602	0.1571	64.3558	0.6202	8.9522	1.0000
PreSet-A	1.0	0.1	19.77	0.2099	0.1336	71.7904	0.7389	15.1685	2.49×10^{-4}
PreSet-B	1.0	\	20.80	0.2197	0.1394	68.3811	0.6722	9.2190	1.0000

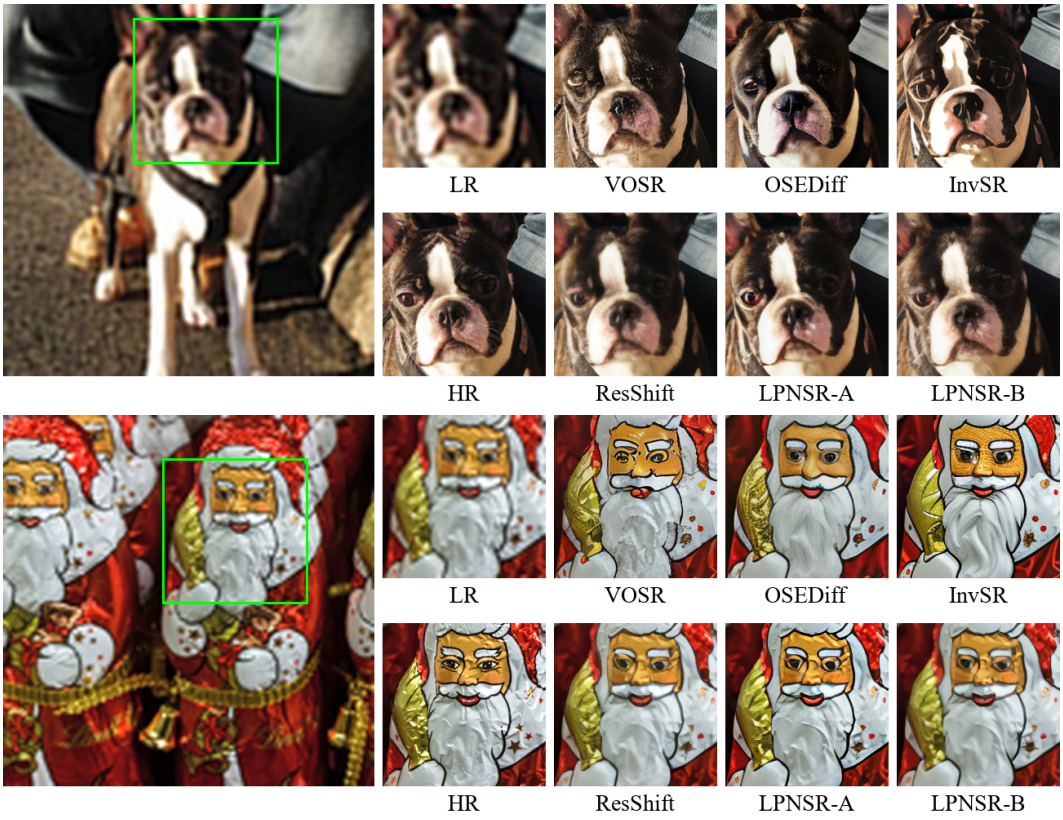


Figure 4: Qualitative comparison with other methods. (Zoom in for best view)

sharper textures and more faithful details, such as the fine fur strands of the dog and the wrinkled foil surface of the Santa Claus. In contrast, VOSR, OSEDiff, and InvSR tend to hallucinate textures and details that are unfaithful to the original image (e.g., the facial features of the dog and of the Santa Claus figure), while ResShift over-smooths the results and thus yields an unclear visual appearance.

Noise Sampling Scheme Comparison. Tab. 1 compares the three noise sampling schemes on the LSDIR and ImageNet datasets, with Origin (ResShift) as the baseline. PreSet-A (LPNSR-A) substantially improves all perceptual metrics at the cost of degraded fidelity and an enlarged FID, whereas PreSet-B (LPNSR-B), derived from the per-step mean-correction analysis in Sec. 3.2, improves fidelity, perceptual quality, and FID simultaneously, though with smaller perceptual gains than PreSet-A. Fig. 3 corroborates these behaviors qualitatively: Origin tends to produce over-smoothed images lacking high-frequency details, PreSet-A recovers richer textures at the cost of slight structural deviation, and PreSet-B achieves a visually appealing balance. Tab. 2 further re-

ports the mean and standard deviation over 100 random seeds on the LSDIR dataset. PreSet-A's standard deviations are consistently much smaller than those of the other two schemes, whereas Origin and PreSet-B fluctuate at similar levels—as expected, since PreSet-B only corrects the mean term of the reverse process, whereas PreSet-A additionally predicts and corrects the variance term. All deviations are remarkably small, confirming that the above observations are not artifacts of a particular seed.

Distribution Alignment Analysis. We further measure the KL divergence between the forward posterior $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}_0)$ and the model reverse distribution $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}_0)$ at each intermediate step ($t = 4, 3, 2$). Both are multivariate Gaussians built from the ground-truth $\boldsymbol{x}_0$ and the denoised estimate $\boldsymbol{x}'_0$, so the divergence follows the standard closed-form solution and decouples into a mean and a variance term, $D_{KL} = D_{KL}^{\text{mean}} + D_{KL}^{\text{var}}$, where $D_{KL}^{\text{mean}} = \frac{1}{2}\boldsymbol{\Delta}^\top \Sigma_\theta^{-1}\boldsymbol{\Delta}$ and $D_{KL}^{\text{var}} = \frac{1}{2}[\operatorname{tr}(\boldsymbol{\Sigma}_\omega - \boldsymbol{I}) - \ln\det(\boldsymbol{\Sigma}_\omega)]$, with $\boldsymbol{\Sigma}_\omega = \operatorname{diag}(\boldsymbol{\sigma}_\omega^2)$ the diagonal covariance of the injected noise and $\boldsymbol{\Delta} = \tilde{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_\theta - \sqrt{\Sigma_\theta}\,\boldsymbol{\mu}_\omega$ the mean deviation between the two distributions. We report $D_{KL}(p \,\|\, q)$ rather than $D_{KL}(q \,\|\, p)$, whose variance term is governed by $\operatorname{tr}(\boldsymbol{\Sigma}_\omega^{-1})$ and thus varies by orders of magnitude across schemes, making the values not directly comparable. Tab. 3 reports the decomposition at each step. PreSet-A achieves the best perceptual quality (Tab. 1) yet deviates the most in its mean term, which explains its degradation in PSNR and SSIM and indicates that the ground-truth image does not necessarily have the best perceptual quality. By contrast, PreSet-B stays closer to the forward posterior mean than Origin, which accounts for its fidelity gains, while its perceptual gains come from approaching the perceptual ceiling of the ground-truth image. PreSet-A's injected noise variance is also orders of magnitude smaller ($\bar{\sigma}_\omega^2 \sim 10^{-4}$ versus 1), consistent with its markedly lower standard deviations across seeds (Tab. 2). The mean term grows as t decreases, since $\boldsymbol{\Delta}$ is scaled by α_t/η_t.

Loss Function Ablation. We ablate the loss functions for training the noise predictors of PreSet-A and PreSet-B, as reported in Tab. 4. For PreSet-B, removing the LPIPS loss (B-Baseline1) improves the fidelity but severely degrades the perceptual quality, indicating that the LPIPS loss is essential for its perceptual quality. For PreSet-A, a similar trend is observed: the L2 loss alone (A-Baseline1) yields the highest pixel fidelity but the weakest perceptual quality, the LPIPS loss (A-Baseline2) markedly improves the perceptual quality at a small cost of fidelity, and the GAN loss (A-Baseline3) further enhances the no-reference realism while degrading the full-reference fidelity. Combining both losses (PreSet-A) achieves the best overall trade-off between pixel-level fidelity and perceptual quality.

Additional Results. More experimental results are provided in the **Appendix**.

5 Conclusion

In this paper, we started from the observation that, once the starting point of the reverse chain and the denoising network are fixed, the quality of the generated image is dictated by the noise maps sampled at the intermediate steps. We therefore parameterized the intermediate noise sampling distribution as a learnable conditional distribution, instantiated it on the residual-shifting diffusion paradigm, and proposed LPNSR, in which a multi-input-aware noise predictor estimates the mean and the variance of the injected noise at each sampling step while leaving the pretrained denoising network and the residual-shifting mechanism untouched. We further developed two training schemes, i.e., the end-to-end supervised training and the mean-correction training, and analyzed the per-step distribution gap to explain their complementary behaviors in fidelity and perceptual quality. Extensive experiments validate the effectiveness of the learnable noise sampling distribution, showing that LPNSR improves the generated image quality over the standard Gaussian distribution.

Future Work. The noise predictor is a plug-and-play module that can be attached to other diffusion frameworks. Applying it to frameworks with richer generative priors, e.g., large-scale pretrained text-to-image models, is expected to further improve the quality of the generated images. Moreover, designing alternative training schemes for the noise predictor is another promising direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools for implementing the proposed method, i.e., for writing the code of the noise predictor and of the associated training and inference pipelines. We have not used generative AI tools for developing the theoretical framework, for formulating the mathematical claims, for providing critical ingredients for the proofs or assisting in their writing, for proposing or refining the hypotheses, for designing the research methodology or the experiments or providing feedback on them, for interpreting the results, or for cleaning and reformatting the datasets. Generating synthetic datasets, assisting with translation, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for editing software code and for editing the manuscript to improve its readability. We have reviewed all AI-assisted work. The AI-assisted code was verified and tested for correctness by the authors, and all references were collected manually and verified against the original publications. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work involves no human subjects and no personally identifiable information, and no copyright concerns arise from the data used. We foresee no ethical concerns arising from the proposed method, which is a general-purpose image super-resolution technique.

REPRODUCIBILITY STATEMENT

The mathematical derivation of the per-step optimal noise and of the mean-correction objective is given in the main text, where all the assumptions and definitions involved are stated. The training details, i.e., the hyperparameters and the architecture of the noise predictor, are described in the main text and the appendix, where the two training procedures are also summarized. All the test datasets used in our experiments are publicly available. The source code of our method is publicly available in a GitHub repository, whose address will be provided in the final version.

REFERENCES

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*, pp. 126–135, 2017.

Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Image super-resolution via progressive cascading residual network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 791–799, 2018.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6228–6237, 2018.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022a.

Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12413–12422, June 2022b.

Ryan Dahl, Mohammad Norouzi, and Jonathon Shlens. Pixel recursive super resolution. In *Proceedings of the IEEE international conference on computer vision*, pp. 5449–5458, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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, 2022.

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, 2016.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, June 2021.

Ben Fei, Zhaoyang Lyu, Liang Pan, Junzhe Zhang, Weidong Yang, Tianyue Luo, Bo Zhang, and Bo Dai. Generative diffusion prior for unified image restoration and enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9935–9946, 2023.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

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. In *Advances in neural information processing systems*, volume 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5967–5976, 2017.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506*, 2023.

Wonjun Kang, Kevin Galim, and Hyung Il Koo. Eta inversion: Designing an optimal eta function for diffusion-based real image editing. In *European Conference on Computer Vision*, pp. 90–106. Springer, 2024.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23593–23606. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/95504595b6169131b6ed6cd72eb05616-Paper-Conference.pdf.

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*, pp. 5148–5157, 2021.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1646–1654, 2016.

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*, pp. 4681–4690, 2017.

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 Workshops*, pp. 1775–1787, 2023.

Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

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

Barak Meiri, Dvir Samuel, Nir Darshan, Gal Chechik, Shai Avidan, and Rami Ben-Ari. Fixed-point inversion for text-to-image diffusion models. *arXiv preprint arXiv:2312.12540*, 2023.

Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. *arXiv preprint arXiv:1812.01608*, 2018.

Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2434–2442, 2020.

Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2063–2072. IEEE, 2025.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6038–6047, 2023.

Thao Nguyen, Yuheng Li, Utkarsh Ojha, and Yong Jae Lee. Visual instruction inversion: Image editing via image prompting. *Advances in Neural Information Processing Systems*, 36:9598–9613, 2023.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pp. 4055–4064. PMLR, 2018.

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 (CVPR)*, pp. 10684–10695, June 2022.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023. doi: 10.1109/TPAMI.2022.3204461.

Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE international conference on computer vision*, pp. 4501–4510, 2017.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 87–103. Springer, 2024.

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

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22532–22541, 2023.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(2):2555–2563, 2023.

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*, pp. 63–79, 2018.

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 Workshops*, pp. 1905–1914, 2021a.

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*, pp. 25796–25805, 2024.

Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE international conference on computer vision*, pp. 370–378, 2015.

Zhihao Wang, Jian Chen, and Steven C. H. Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2021b.

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.

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.

Rongyuan Wu, Lingchen Sun, Zhengqiang Zhang, Xiangtao Kong, Jixin Zhao, Shihao Wang, and Lei Zhang. Vosr: A vision-only generative model for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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 Workshops*, pp. 1191–1200, 2022.

Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 13294–13307. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2ac2eac5098dba08208807b65c5851cc-Paper-Conference.pdf.

Zongsheng Yue, Kang Liao, and Chen Change Loy. Arbitrary-steps image super-resolution via diffusion inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23153–23163, June 2025.

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*, pp. 4791–4800, 2021.

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*, pp. 586–595, 2018.

A Appendix

In the appendix, we provide the following materials:

- The hyperparameter configuration of the noise predictor, including the model architecture and the training hyperparameters.
- The complete training algorithms of our LPNSR framework.
- The inference efficiency analysis of the three noise sampling schemes.
- A comparison between the noise sampling distribution and a deterministic noise map under the end-to-end supervised training scheme.
- A capacity-matched comparison between the mean-correction formulation and a monolithic model of comparable size.
- A comparison between learning the initialization noise and learning the intermediate noise.
- More qualitative comparisons of the SR results.

A.1 Hyperparameter Configuration

The architecture of the noise predictor $\boldsymbol{g}_\omega$ follows the default setting of ResShift (Yue et al., 2023). The key model and training hyperparameters are summarized in Tabs. 5 and 6.

Table 5: Architecture hyperparameters of the noise predictor $\boldsymbol{g}_\omega$.

Hyperparameter	Value
Base channels	160
Channel multipliers	$(1, 2, 2, 4)$
Residual blocks per resolution	2
Attention resolutions	$64, 32, 16, 8$
Channels per attention head	32
Scale-shift normalization	Enabled
Swin Transformer depth	2
Swin Transformer embedding dimension	192
Swin Transformer window size	8
MLP ratio	4.0
Dropout	0.0
LR latent resolution	64×64
Latent channels	3
LR conditioning	Enabled

A.2 Training Algorithms

The end-to-end supervised training and the mean-correction training of the noise predictor are summarized in Alg. 1 and Alg. 2, respectively.

Algorithm 1 End-to-End Noise Predictor Training

Require: HR/LR training pairs, frozen VQGAN $\mathcal{E}/\mathcal{D}$, frozen denoiser $\boldsymbol{f}_\theta$, noise predictor $\boldsymbol{g}_\omega$, total iterations N
Ensure: Trained noise predictor $\boldsymbol{g}_\omega$
for $iteration = 1, \ldots, N$ **do**
  Sample HR/LR images from the training set; $\boldsymbol{x}_0 = \mathcal{E}(\text{HR image})$, $\boldsymbol{y}_0 = \mathcal{E}(\text{LR image})$; $\boldsymbol{x}_T = \boldsymbol{y}_0 + \kappa\,\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  for $t = T, T-1, \ldots, 1$ **do**
    $\boldsymbol{x}'_0 = \boldsymbol{f}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t)$
    if $t > 1$ **then**
      $\boldsymbol{x}_{t-1} = \frac{\eta_{t-1}}{\eta_t}\,\boldsymbol{x}_t + \frac{\alpha_t}{\eta_t}\,\boldsymbol{x}'_0 + \kappa\sqrt{\frac{\eta_{t-1}\,\alpha_t}{\eta_t}}\,\boldsymbol{g}_\omega(\boldsymbol{x}_t, \boldsymbol{x}'_0, \boldsymbol{y}_0, t)$
    end if
    $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}'_0$
  end for
  $\mathcal{L} = \mathcal{L}_2(\hat{\boldsymbol{x}}_0,\, \boldsymbol{x}_0) + \lambda_l\,\mathcal{L}_{\text{LPIPS}}(\mathcal{D}(\hat{\boldsymbol{x}}_0),\, \boldsymbol{x}_0) + \lambda_g\,\mathcal{L}_{\text{GAN}}(\hat{\boldsymbol{x}}_0,\, \boldsymbol{x}_0)$
  Update $\boldsymbol{g}_\omega$
end for
return $\boldsymbol{g}_\omega$

Algorithm 2 Mean-Correction Noise Predictor Training

Require: HR/LR training pairs, frozen VQGAN encoder/decoder $\mathcal{E}/\mathcal{D}$, frozen denoiser $\boldsymbol{f}_\theta$, noise predictor $\boldsymbol{f}_\omega$, total iterations N
Ensure: Trained noise predictor $\boldsymbol{f}_\omega$
for $iteration = 1, \ldots, N$ **do**
  Sample HR/LR images from the training set; $\boldsymbol{x}_0 = \mathcal{E}(\text{HR image})$, $\boldsymbol{y}_0 = \mathcal{E}(\text{LR image})$; sample $t \sim \text{Uniform}\{1, \ldots, T\}$
  Forward: $\boldsymbol{x}_t = \boldsymbol{x}_0 + \eta_t\,(\boldsymbol{y}_0 - \boldsymbol{x}_0) + \kappa\sqrt{\eta_t}\,\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{x}'_0 = \boldsymbol{f}_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t)$
  $\boldsymbol{r} = \boldsymbol{f}_\omega(\boldsymbol{x}_t, \boldsymbol{x}'_0, \boldsymbol{y}_0, t)$
  $\mathcal{L} = \|\boldsymbol{r} - (\boldsymbol{x}_0 - \boldsymbol{x}'_0)\|_2^2 + \mathcal{L}_{\text{LPIPS}}(\mathcal{D}(\boldsymbol{x}'_0 + \boldsymbol{r}),\, \boldsymbol{x}_0)$
  Update $\boldsymbol{f}_\omega$
end for
return $\boldsymbol{f}_\omega$

Table 6: Training hyperparameters of the noise predictor.

Hyperparameter	Value
Training iterations	200k
Batch size	16
Optimizer	AdamW
Learning rate	5×10^{-5}
Weight decay	0
LR schedule	None
EMA rate	0.999
Gradient clipping	1.0
Mixed precision (AMP)	Enabled
Sampling steps T	4
Noise variance κ	2.0
Shift schedule	Exponential
Loss weights ($\mathcal{L}_2$ / LPIPS / GAN)	1.0 / 1.0 / 0.1
Discriminator	PatchGAN with hinge loss

Table 7: Inference efficiency of the three noise sampling schemes on the LSDIR dataset (128×128 LR $\rightarrow$ 512×512 SR), measured on a single NVIDIA RTX 5090 GPU with a batch size of 1 and mixed precision enabled. The number of parameters and the FLOPs are reported per image, and the latency and the GPU memory usage are averaged over all test images.

Method	#Params (M)↓	FLOPs (G)↓	Latency (ms)↓	GPU Memory Usage (MB)↓
Origin	173.92	5410.3	321.8	1968
PreSet-A	292.52	7032.3	442.1	2432
PreSet-B	292.51	7031.8	438.5	2432

A.3 Efficiency Analysis

Tab. 7 reports the inference efficiency of the three schemes. Since the noise predictor only estimates the mean and the variance of the injected noise at each of the 4 sampling steps, the extra cost is moderate, raising the latency and the GPU memory by about $1.4\times$ and $1.2\times$, respectively. PreSet-A and PreSet-B share the same architecture and hence nearly identical efficiency. Overall, the inference cost remains within an acceptable range.

A.4 Distribution Prediction versus Deterministic Noise Map

Under the end-to-end supervised training scheme, the noise predictor admits two different instantiations. In the first one, the predictor estimates the mean and the variance of the noise sampling distribution at each intermediate step, so that the injected noise is sampled from the resulting conditional Gaussian. In the second one, the predictor directly outputs a deterministic noise map at each intermediate step, i.e., the stochastic term is removed and the predicted map itself is injected. Both instantiations are trained with exactly the same configuration. We compare the two instantiations in Tab. 8. Although the variance predicted by the distribution-based instantiation is orders of magnitude smaller than that of the standard Gaussian ($\bar{\sigma}_\omega^2 \sim 10^{-4}$, Tab. 3 in the main paper), it still performs better than its deterministic counterpart on all the reported metrics, indicating that discarding the stochastic term entirely degrades the quality of the generated images.

A.5 Model Capacity versus Residual Correction

The improvement of PreSet-B over the standard Gaussian could in principle be attributed to the extra parameters of its residual predictor rather than to the mean-correction formulation. To separate the two factors, we train two models from scratch that share the architecture of the noise predictor in Table 5, the training data, the losses, the four-step sampling procedure, and the total training budget,

Table 8: Ablation on the form of the noise predictor under the end-to-end supervised training scheme, i.e., predicting the mean and the variance of the noise sampling distribution versus directly predicting a deterministic noise map. The results are evaluated on the LSDIR dataset under the 4-step sampling setting.

Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	MANIQA↑	CLIPIQA↑
Deterministic noise map	19.73	0.5174	0.2315	0.1456	70.0219	0.4491	0.6452
Noise sampling distribution	19.77	0.5140	0.2099	0.1336	71.7904	0.5618	0.7389

Table 9: Model capacity versus residual correction. Both models are trained from scratch with comparable parameter counts and the same total training budget, and are evaluated on the LSDIR dataset under the 4-step sampling setting.

Method	#Params (M)	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	MANIQA↑	CLIPIQA↑
Mono	48.19	20.13	0.5126	0.2428	0.1597	66.1782	0.4197	0.6327
Split	50.16	20.52	0.5248	0.2316	0.1488	66.9657	0.4384	0.6533

and differ in how the capacity is organized: the monolithic model (Mono) is a single network trained directly, whereas the split model (Split) first trains a denoising network, then freezes it, and trains a residual predictor on top of it with the mean-correction objective. Mono uses a base channel count of 96, i.e., 48.19M parameters, whereas Split uses a base channel count of 64 for both its frozen denoising network and its residual predictor, i.e., 25.08M parameters each and 50.16M in total. Tab. 9 reports the results on the LSDIR dataset. Split improves over Mono on all seven metrics, and on the distortion and reference-based perceptual metrics the margin even exceeds the one by which PreSet-B improves over Origin at the full scale (Tab. 1), despite a much smaller capacity gap than the one that the noise predictor introduces there. At this scale the base denoising network is fitted less accurately, and introducing the residual predictor yields a larger improvement.

A.6 Initialization Noise versus Intermediate Noise

The reverse chain receives noise at two places: the initialization $\boldsymbol{x}_T = \boldsymbol{y}_0 + \kappa\,\boldsymbol{\epsilon}$, i.e., a single noise map, and the intermediate steps, i.e., the $T-1$ maps $\{\boldsymbol{z}_{t-1}\}_{t=2}^{T}$; diffusion inversion methods such as InvSR (Yue et al., 2025) learn the former, whereas LPNSR learns the latter. To compare the two choices, we train an initialization noise predictor that has the same architecture and training configuration as that of PreSet-A but takes $(\boldsymbol{x}_T, \boldsymbol{y}_0, t)$ as input, models the initialization noise as a diagonal Gaussian, and is trained end-to-end, while the intermediate noise maps are kept standard Gaussian. This isolates where the learnable noise is injected, i.e., the initialization, the intermediate steps (PreSet-A), or neither (Origin). Tab. 10 reports the results on the LSDIR dataset. Init brings almost no improvement over Origin, whereas PreSet-A improves perceptual quality substantially. This is likely because the gradient signal that end-to-end training delivers back to the initialization predictor is weak: as a multi-step framework, ResShift (Yue et al., 2023) lets the initialization noise enter only at the first step, so its effect is diluted along the remaining reverse steps and further obscured by the stochastic noise injected at the intermediate steps. The same reasoning explains why an initialization noise predictor remains effective in InvSR: it is trained through a single denoising step of a much stronger prior, SD-Turbo (Sauer et al., 2024), so the predicted noise is directly connected to the generated result, whereas in our multi-step chain it is separated from the loss by all the remaining steps.

A.7 More Qualitative Comparisons

We provide additional qualitative comparisons of our method against the compared methods in Figs. 5 and 6.

Table 10: Initialization noise versus intermediate noise. The three models differ only in where the learnable noise is injected: the initialization (Init), the intermediate steps (PreSet-A), or neither (Origin). All are evaluated on the LSDIR dataset under the 4-step sampling setting.

Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	MANIQA↑	CLIPIQA↑
Origin	20.70	0.5281	0.2249	0.1420	67.4741	0.4363	0.6445
Init	20.68	0.5277	0.2242	0.1417	67.5212	0.4366	0.6447
PreSet-A (LPNSR-A)	19.77	0.5140	0.2099	0.1336	71.7904	0.5618	0.7389

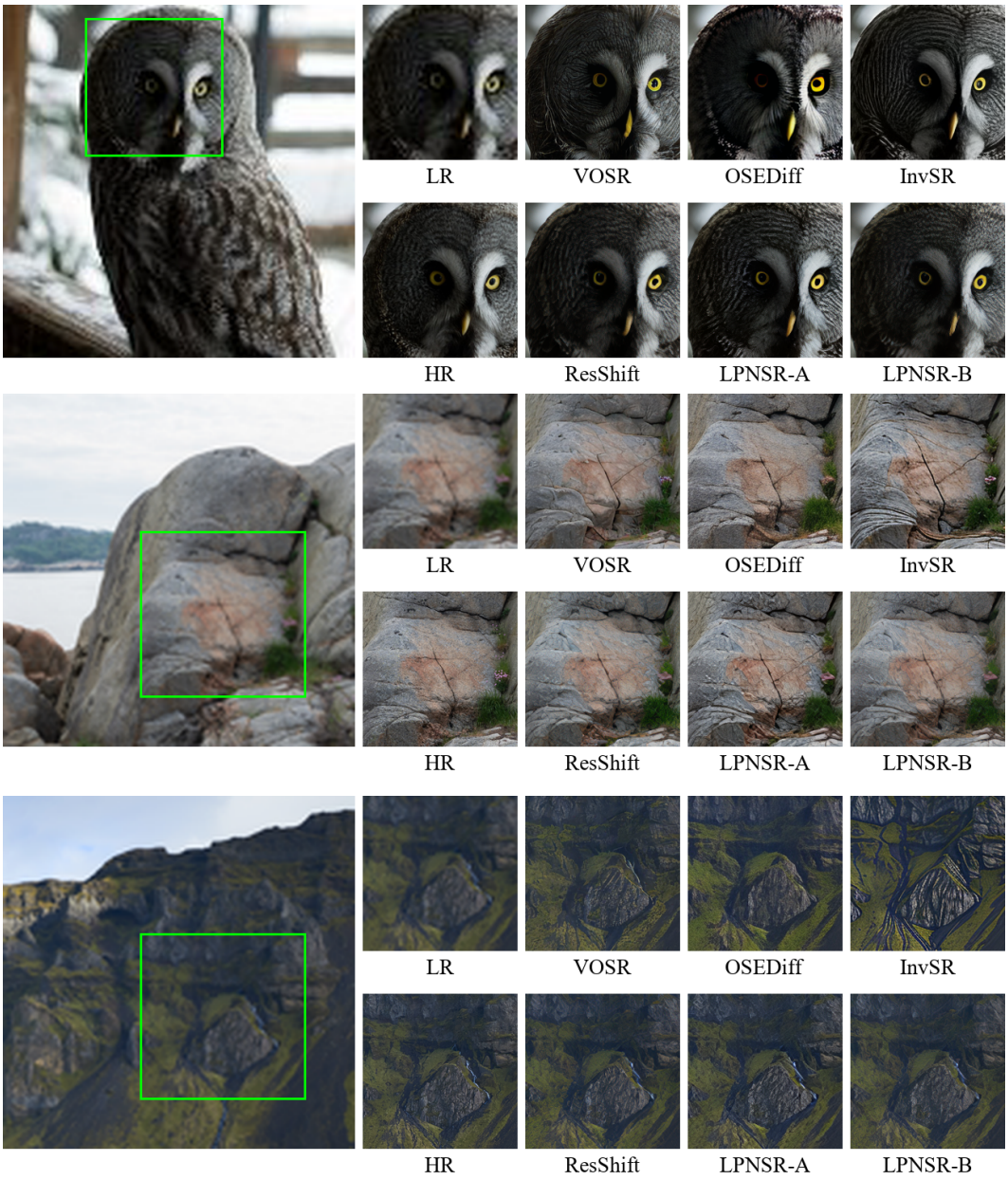


Figure 5: Additional qualitative comparison with other methods. (Zoom in for best view)

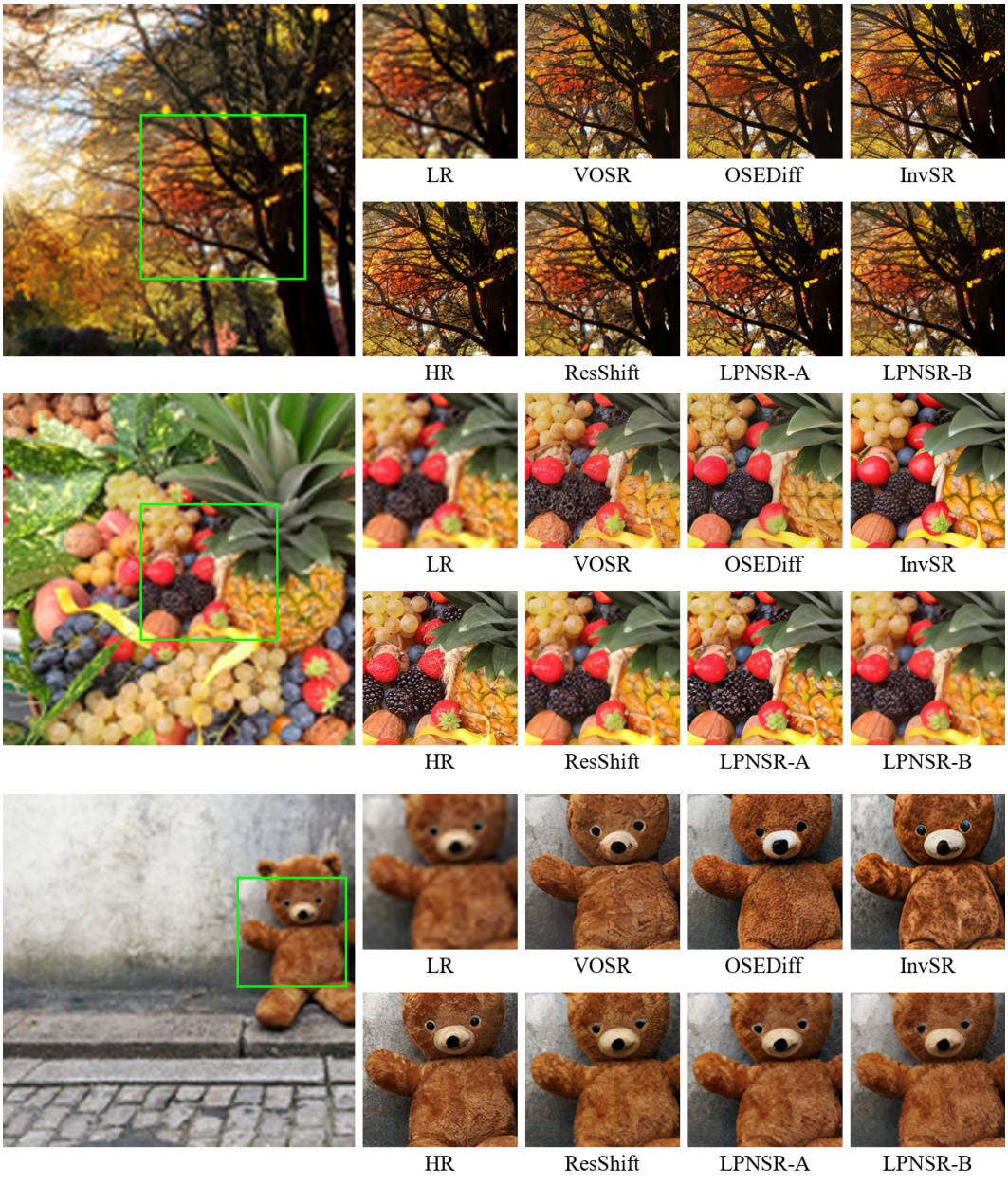


Figure 6: Additional qualitative comparison with other methods. (Zoom in for best view)