

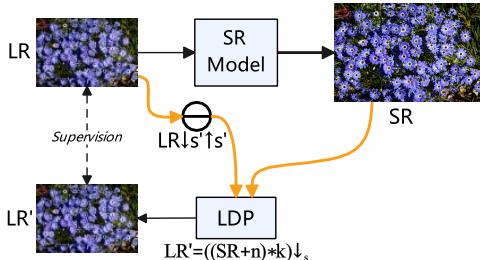
# 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 LDP: A LIGHTWEIGHT DENOISING PLUGIN ENHANCING GENERALIZATION IN SINGLE-IMAGE SUPER-RESOLUTION

006 **Anonymous authors**

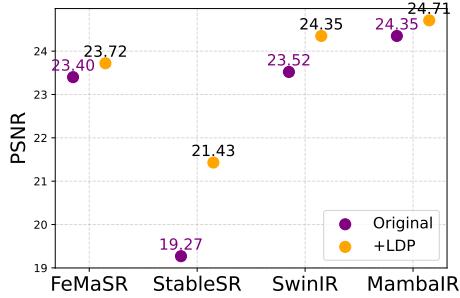
007 Paper under double-blind review

## ABSTRACT

013 Current single-image super-resolution (SISR) models struggle to generalize to  
014 real-world degradations. To address this challenge, we propose LDP, an inno-  
015 vative lightweight denoising autoencoder (DAE) plug-in. It improves the gener-  
016 alization ability of SR models via low-resolution (LR) images prediction-based  
017 cyclic regularization. LDP models the SISR degradation process within the DAE  
018 framework. It leverages a property of diffusion models, where after noise is added,  
019 high-resolution (HR) images and LR features become aligned, so that denoising  
020 noisy HR features is equivalent to denoising noisy LR features. During the cor-  
021 ruption process, noise is added independently to each HR patch. During the de-  
022 noising process, a convolutional denoiser uses learned filters to approximate blur  
023 kernels. In addition, LR degradation is used to distinguish different LR from the  
024 same HR. LDP can be applied to SR models in two modes: as a training loss to  
025 improve reconstruction quality, or as an inference post-processing step to correct  
026 artifacts. Extensive experiments demonstrate that LDP substantially improves the  
027 generalization of existing SR models to unseen degradations.



(a) Arbitrary SR model



(b) Boosts over baselines (DIV2K-Hybrid)

041 Figure 1: Our LDP is a lightweight denoising autoencoder-based plug-in that can be seamlessly  
042 integrated into arbitrary SR models, operating as a training-time loss or an inference-time module.

## 1 INTRODUCTION

047 Single Image Super-Resolution (SISR) aims to reconstruct high-resolution (HR) images from their  
048 low-resolution (LR) counterparts. SISR is widely applied in various fields, such as medical imag-  
049 ing Li et al. (2024a) and remote sensing Dong et al. (2024). Deep learning has advanced SISR  
050 architectures from Convolutional Neural Network (CNN) Dong et al. (2014) to Transformer Liang  
051 et al. (2021); Chen et al. (2023b) and State-Space Model Guo et al. (2024; 2025), achieving higher  
052 reconstruction accuracy. Meanwhile, generative methods, including Generative Adversarial Net-  
053 work (GAN) Chen et al. (2022) and Diffusion Model Wang et al. (2024); Yue et al. (2025); Zhang  
et al. (2025), have been explored to improve perceptual quality.

Despite advances in SR architectures, existing models struggle to generalize to unseen degradations. Recent approaches leverage data augmentation and self-supervised learning techniques to tackle this challenge. Data augmentation approaches typically fall into two categories: generating synthetic distortions Zhang et al. (2021a); Wang et al. (2021), or employing generative models Li et al. (2022); Chen et al. (2025) to synthesize paired data from unpaired LR and HR images. However, these methods may harm performance Zhang et al. (2023) or are limited to in-distribution datasets. Self-supervised approaches rely on either image-specific training Shocher et al. (2018); Ulyanov et al. (2018) or test-time adaptation Hussein et al. (2020); Zhou et al. (2023); Chen et al. (2024), utilizing internal image statistics and priors. However, they suffer from high computational cost or the need for model-specific adaptation. Addressing unseen degradations efficiently remains a key challenge.

To address these limitations, we propose LDP, a lightweight denoising autoencoder (DAE) plug-in. It improves the generalization ability of SR models via LR prediction-based cyclic regularization. LDP models the SISR degradation process within the DAE framework. It leverages a property of diffusion models, where after noise is added, high-resolution (HR) images and LR features become aligned Wang et al. (2023b), making denoising noisy HR features equivalent to denoising noisy LR features. LDP takes high-resolution images (ground-truth HR or SR outputs) as input for degradation modeling, with LR high-frequency components as a condition to distinguish different LR images from the same HR. During the corruption process, LDP introduces patch-dependent Gaussian noise. This enables the model to learn fine-grained degradation in local patches, rather than assuming the same degradation for the whole image. During the denoising process, a lightweight convolutional denoiser learns the blur kernels associated with the degradation model. Built on these designs, LDP accurately generates corresponding LR image and generalizes well to unseen degradations. LDP applies to SR models in two modes: as a training-time loss function to improve reconstruction quality, or as an inference-time post-processing step that corrects artifacts independently of training. Extensive experiments verify that LDP significantly improves the generalization ability of existing SR models on unknown complex degradations.

Overall, our contributions are three-fold:

- We propose LDP, an innovative lightweight denoising autoencoder plug-in for single-image super-resolution that enhances the generalization of existing SR models.
- LDP is a conditional degradation model that generates LR images from HR inputs by explicitly conditioning on LR high-frequency components. LDP operates in two modes: as a degradation-aware training-time loss function, or as an inference-time correction module (e.g., Posterior Sampling for diffusion models).
- LDP enhances reconstruction quality during training as a loss function and mitigates artifacts at inference independently of training. Both modes improve SR model generalization to unknown complex degradations.

## 2 RELATED WORK

### 2.1 IMPROVING GENERALIZATION IN SR

The limited generalization ability of SR models to unseen degradations remains a major challenge for real-world applications. Existing SR methods address this issue using two main approaches: data augmentation and self-supervised learning. Data augmentation methods seek to bridge the training–inference gap by creating synthetic data with degradations that approximate real-world scenarios. One line of works explicitly model degradations using predefined operations. BSRGAN Zhang et al. (2021a) generates complex degradations by sequentially combining downsampling, blur, noise, and compression in random order, producing varied LR images for training. RealESRGAN Wang et al. (2021) introduces higher-order degradations to reflect real-world degradation chains. While BSRGAN and RealESRGAN enable non-blind SR models to handle blind scenarios through multi-degradation training, such strategies may compromise performance on in-distribution benchmarks Zhang et al. (2023). Alternatively, implicit modeling methods leverage generative models to synthesize paired data from real LR and unpaired HR images. GAN Yuan et al. (2018); Li et al. (2022); Yin et al. (2023) or diffusion-based Chen et al. (2025) methods learn degradation priors to create realistic training pairs. However, their generalization remains limited to in-distribution data. Self-supervised learning enables SISR training using only LR images without

108 paired HR supervision. ZSSR Shocher et al. (2018) and DIP Ulyanov et al. (2018) exploit internal  
 109 patterns or implicit priors without external data. CorrectFilter Hussein et al. (2020); Zhou et al.  
 110 (2023) aligns inputs with the training distribution of pre-trained models. Lway Chen et al. (2024)  
 111 uses a degradation model to synthesize LR images from SR outputs for test-time fine-tuning. Al-  
 112 though effective, these methods are computationally expensive or require model-specific adaptation.  
 113

## 114 2.2 CONSTRAINING THE SR SOLUTION SPACE VIA DEGRADATION MODELING

115 Degradation modeling, applied jointly with the SR model, introduces structural constraints that en-  
 116 sure reconstructed LR outputs align with the LR input, effectively narrowing the solution space to  
 117 favor LR-consistent reconstructions. DRN Guo et al. (2020) adds a degradation branch that projects  
 118 SR outputs back to the LR domain, enforcing reconstruction consistency and improving stability.  
 119 DualSR Emad et al. (2021) introduces a dual-path framework where a GAN-based downsample  
 120 and an upsample are jointly trained with cycle consistency to model and reverse image-specific  
 121 degradations. SCL-SASR Chen et al. (2023a) adopts a similar bidirectional design under MAP es-  
 122 timation, coupling SR and degradation networks to adapt to test-time degradations. Lway Chen  
 123 et al. (2024) introduces test-time adaptation with pre-trained degradation models to fine-tune SR  
 124 models, increasing generalization to unseen degradations. Despite their benefits, these methods  
 125 face several limitations: DRN handles only bicubic downsampling; DualSR and SCL-SASR re-  
 126 quire image-specific optimization or joint training; and Lway introduces significant computational  
 127 overhead due to its large model size. In contrast, our method supports a wide range of degradations  
 128 through an explicitly modeled degradation process within a lightweight denoising autoencoder  
 129 framework. Our degradation modeling framework is adaptable to various training settings, from  
 130 large-scale supervised learning to image-specific fine-tuning, and can also be applied directly at test  
 131 time. The framework is lightweight and does not incur significant computational cost.

132 Degradation modeling is also applied during inference in diffusion-based image restoration to en-  
 133 force LR consistency. ILVR Choi et al. (2021) guides the sampling process of DDPM Ho et al.  
 134 (2020) using a reference image to maintain low-frequency consistency across the denoising steps.  
 135 DR2 Wang et al. (2023b) shows that injecting additional Gaussian noise makes LR and HR distribu-  
 136 tions less distinguishable, allowing noise-corrupted LR images to be treated as noise-corrupted HR  
 137 images during sampling. MCG Chung et al. (2022) ensures samples stay close to the data manifold  
 138 by projecting the gradient of the measurement function onto its tangent space. DPS Chung et al.  
 139 (2023) further leverages the degradation process to connect the LR observation to the predicted  
 140 clean image at each step. In our method, LDP degrades each predicted clean image during diffusion  
 141 inference, treating it as SR to produce a predicted LR image. We then enforce LR cyclic consistency  
 142 by applying the tailored loss  $\mathcal{L}_{\text{sym}}^{\text{FT}}$  (Eq. 16), which penalizes the discrepancy between the predicted  
 143 LR and the ground-truth LR. This degradation-aware constraint enhances fidelity by suppressing  
 144 artifacts and promoting structural consistency in the SR results.

## 145 3 PROPOSED METHOD

146 Section 3.1 outlines the motivation behind LDP. Section 3.2 introduces the overall framework of  
 147 LDP. Section 3.3 then details its training and inference modes, describing LDP’s own training, its  
 148 application in fine-tuning SR models, and its role as a post-processing step for diffusion models.  
 149

### 150 3.1 MOTIVATION

151 To improve the generalization of existing SR models on unknown complex degradations, we adopt  
 152 a degradation modeling approach applied jointly with the SR model. This introduces structural  
 153 constraints that ensure the reconstructed LR outputs are aligned with the LR input, effectively nar-  
 154 rowing the solution space to favor LR-consistent reconstructions. Our LDP integrates degradation  
 155 modeling Yue et al. (2022) into the denoising autoencoder, reinterpreting denoising as a controllable  
 156 degradation applied to HR images. In the classical degradation formulation, this can be expressed  
 157 as:

$$y = ((x + n) \otimes k) \downarrow_s, \quad (1)$$

158 where  $x \in \mathbb{R}^{H \times W \times 3}$  is the HR image,  $y \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times 3}$  is the LR image,  $n$  is the noise,  $k$  is the  
 159 blur kernel, and  $s$  is the downsampling scale. We further leverage a property of diffusion models,

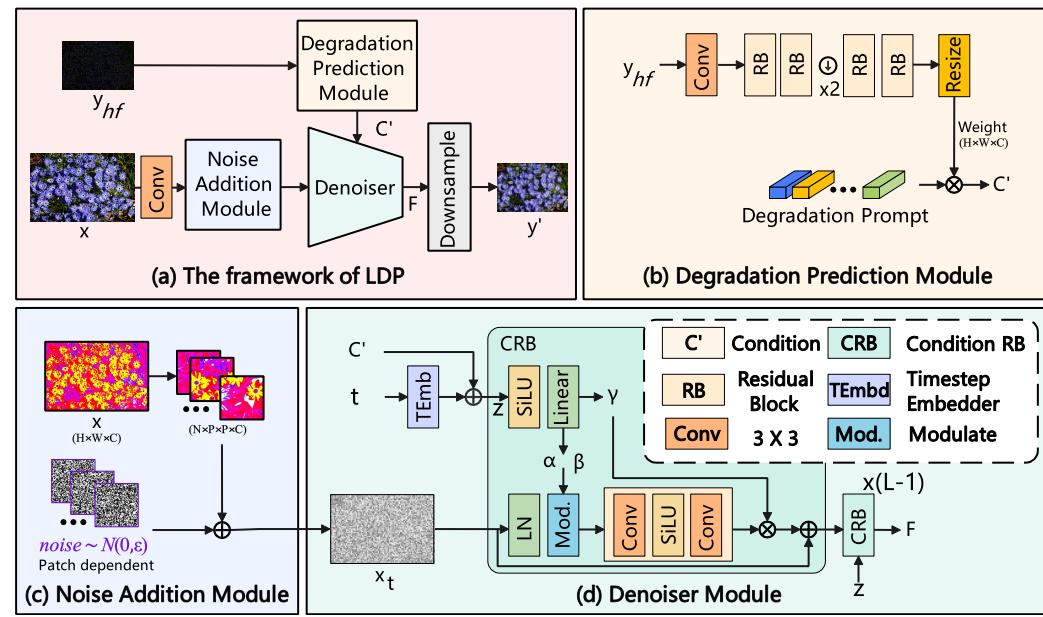


Figure 2: (a) **LDP Framework.**  $LR_{hf}$  predicts degradation  $C'$ , guiding the noise-perturbed HR features to generate the LR output via denoising and downsampling. (b) **Degradation Prediction.** Stacked RB generate weights from  $LR_{hf}$  and multiply them with  $P_D$  to produce  $C'$ . (c) **Noise Addition.** Patch-dependent noise is added to HR features at random timesteps. (d) **Denoiser.** A lightweight CNN denoises  $HR_t$  conditioned on  $z$  using CRBs with AdaLN.

whereby after noise is added, HR features and LR features become aligned Wang et al. (2023b), making denoising noisy HR features equivalent to denoising noisy LR features. This allows us to perform degradation modeling on HR images using a denoising autoencoder. However, there remains a challenge: since the SR task is inherently ill-posed, a condition is required to differentiate between different LR images generated from the same HR image under varying degradations. This condition must satisfy three criteria: (1) it cannot be the LR image itself, otherwise the network might take shortcuts and fail to learn meaningful degradations; (2) it must be discriminative for different LR images corresponding to the same HR image; and (3) it should be simple and easy to obtain. We define this condition as  $LR_{hf}$ , obtained by subtracting the  $s'$ -fold downsampled-then-upsampled LR image from the original LR image. In summary, we use a denoising autoencoder to perform degradation modeling on the input HR image, with the condition  $LR_{hf}$  controlling the type of degradation in the output. During application, this approach constrains the super-resolution (SR) model to produce outputs whose LR reconstructions (via our LDP) are consistent with the original LR input, thus enforcing LR cyclic consistency and effectively guiding the SR model.

### 3.2 FRAMEWORK

Figure 2 (a) illustrates the framework of our proposed LDP, which consists of four main modules: the Degradation Prediction Module (DPM), Noise Addition Module (NAM), Denoiser Module and Downsample Module. Designed as a denoising autoencoder, LDP functions as a conditional degradation model that generates LR images from HR inputs by conditioning on LR high-frequency components. To facilitate both implementation and interpretability, we adopt the noise corruption process from diffusion models Ho et al. (2020). The overall process of LDP is formulated as:

$$x_t = NAM(x, t), \quad (2)$$

$$y' = D(Denoiser(x_t | DPM(y_{hf}), t)), \quad (3)$$

Where  $y'$  is the predicted LR images, and  $y_{hf}$  is the LR high-frequency component.  $t$  is a patch-dependent timestep,  $x_t$  is the noised HR features,  $NAM(\cdot)$  is the Noise Addition Module,  $DPM(\cdot)$  is the Degradation Prediction Module and  $D(\cdot)$  is the Downsample Module.

216 **Degradation Prediction Module.** Figure 2 (b) shows the DPM diagram. Its input is the high-  
 217 frequency component of the LR image, computed by subtracting the  $s'$ -fold downsampled-then-  
 218 upsampled LR image from the original LR image, which can be formulated as:  
 219

$$220 \quad y_{hf} = y - y \downarrow_{s'} \uparrow_{s'}, \quad (4)$$

221 where  $\downarrow_{s'}$  and  $\uparrow_{s'}$  denote the downsampling and upsampling operations with scale factor  $s'$ , re-  
 222 spectively. To extract degradation information, we use prompts to encode degradation-specific de-  
 223 tails Potlapalli et al. (2023). First, a weight map  $w$  is derived from  $y_{hf}$ , and then resized to match the  
 224 spatial dimensions of  $x$  (i.e.,  $H \times W$ ). This resized weight map is multiplied element-wise with the  
 225 Degradation Prompt  $P_D$ . It forms a degradation map  $C' \in \mathbb{R}^{H \times W \times C}$  and serves as the condition  
 226 for the denoiser. The process can be formulated as:  
 227

$$228 \quad w = (\text{RB}_4 \circ \text{RB}_3 \circ \downarrow_2 \circ \text{RB}_2 \circ \text{RB}_1) \circ \text{Conv}(y_{hf}), \quad (5)$$

$$229 \quad C' = P_D \otimes \text{Resize}(w, H, W), \quad (6)$$

230 where  $\text{RB}(\cdot)$  denotes a residual block consisting of two  $3 \times 3$  convolutional layers with a SiLU  
 231 activation in between,  $\text{Conv}(\cdot)$  represents a convolutional layer,  $\circ$  denotes function composition ap-  
 232 plied sequentially from right to left, and  $\otimes$  denotes element-wise multiplication. The downsampling  
 233 operator  $\downarrow_2$  further reduces spatial resolution and disrupts local structures. The degradation prompt  
 234  $P_D \in \mathbb{R}^{N_p \times C}$  is jointly learned to encode degradation-specific information.  
 235

236 **Noise Addition and Denoiser Module.** Our framework integrates degradation modeling Yue et al.  
 237 (2022) into the denoising autoencoder, reinterpreting denoising as a controllable degradation applied  
 238 to HR images. During the corruption process, we perturb HR images using a patch-wise noise  
 239 schedule. Specifically, following the diffusion noise schedule, each patch  $x_i \in \mathbb{R}^{P \times P \times C}$  is assigned  
 240 a random timestep  $t_i$ , and its noisy version is obtained as:  
 241

$$241 \quad x_i^{(t_i)} = \sqrt{\hat{\alpha}_{t_i}} x_i + \sqrt{1 - \hat{\alpha}_{t_i}} \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \mathbf{I}), \quad (7)$$

242 where  $\hat{\alpha}_{t_i}$  denotes the cumulative product of noise scheduling coefficients at time  $t_i$  and  $\epsilon_i$  is stan-  
 243 dard Gaussian noise. This patch-wise formulation enables each image region to undergo a different  
 244 level of degradation, allowing the model to better capture spatially varying corruption. The final  
 245 noisy image is denoted as  $x_t$ .  
 246

247 During the denoising process, a lightweight CNN acting as the denoiser module estimates the blur  
 248 kernel and extracts intermediate feature  $F$  conditioned on the degradation map  $C'$ . The feature  $F$  are  
 249 then downsampled to produce the predicted LR image. Specifically, the denoiser module comprises  
 250  $L$  Condition Residual Blocks (CRBs) that leverage Adaptive Layer Normalization (AdaLN) Perez  
 251 et al. (2018); Li et al. (2024b) for conditional modulation. For each  $P \times P$  patch, the assigned  
 252 timestep  $t_i$  is embedded and combined with  $C'$  to produce a patch-specific condition  $z$ . This condi-  
 253 tion is passed through a SiLU activation and a linear layer to generate modulation parameters  $\alpha$ ,  $\beta$ ,  
 254 and  $\gamma$  corresponding to scaling, bias, and gating. In the residual path, features are first normalized  
 255 via LayerNorm and modulated by  $\alpha$  and  $\beta$ , then processed by a residual block, gated with  $\gamma$ , and  
 256 finally added back to the input. The CRB can be formulated as:  
 257

$$258 \quad t_{emb} = \text{TEmb}(t), \quad (8)$$

$$259 \quad \alpha, \beta, \gamma = \text{Linear}(\text{SiLU}(C' + t_{emb})), \quad (9)$$

$$260 \quad x'_t = \alpha \otimes (\text{LN}(F_{i-1})) + \beta, \quad (10)$$

$$261 \quad F_i = \gamma \otimes \text{RB}(x'_t) + F_{i-1}, \quad (11)$$

262 where  $\text{TEmb}(\cdot)$  is the timestep embedder,  $F_{i-1}$  is the output of the previous CRB, and the initial  
 263 feature is set as  $F_0 = x_t$ . The  $\text{RB}(\cdot)$  in the final CRB is simplified to a single convolutional layer.  
 264

265 **Downsample Module.** The module adjusts the feature map to match the spatial resolution of the  
 266 original LR image. Features  $F$  are first downsampled by a factor of  $s$ , then processed by a residual  
 267 block and a convolutional layer:  
 268

$$269 \quad y' = \text{Conv}(\text{RB}(F \downarrow_s)). \quad (12)$$

270 Here, RB and the final convolutional layer are used to enhance feature representation and maintain  
 271 smooth transitions between downsampled regions.  
 272

270 3.3 TRAINING AND INFERENCE MODES OF LDP  
271

272 **Training LDP.** Following Lway Chen et al. (2024), LDP is trained by supervising only the high-  
273 frequency components of the predicted LR images. We apply the Discrete Wavelet Transform  
274 (DWT) to decompose the predicted LR image  $y'$  into four subbands (LL, LH, HL, HH). The high-  
275 frequency subbands (LH, HL, HH) are then summed and normalized to form a weight map  $M$ ,  
276 which is subsequently used to compute both the L1 loss and the LPIPS loss Zhang et al. (2018):  
277

$$\mathcal{L}_{sym}^T = \lambda_1 \mathcal{L}_1(M \otimes y', M \otimes y) + \lambda_2 \mathcal{L}_{LPIPS}(M \otimes y', M \otimes y), \quad (13)$$

278 where  $\lambda_1$  and  $\lambda_2$  are the corresponding loss weights.  
279

280 **Fine-Tuning SR Models with LDP.** In fine-tuning, the original loss of pretrained SR models is aug-  
281 mented with a frequency loss Xie et al. (2023) that supervises the amplitude and phase components  
282 of SR and HR images in the frequency domain:  
283

$$\mathcal{L}_{fre} = \frac{1}{HW} \sum_{u=0}^{H-1} \sum_{v=0}^{W-1} D(\mathcal{F}(x')(u, v), \mathcal{F}(x)(u, v)), \quad (14)$$

$$D(\mathcal{F}(x'), \mathcal{F}(x)) = \left( (\mathcal{R}(\mathcal{F}(x')) - \mathcal{R}(\mathcal{F}(x)))^2 + (\mathcal{I}(\mathcal{F}(x')) - \mathcal{I}(\mathcal{F}(x)))^2 \right)^{\gamma/2}, \quad (15)$$

289 where  $x$  and  $x'$  are the HR image and SR result,  $\mathcal{F}(x)$  denotes the 2D Fourier transform of  $x$ , and  
290  $\mathcal{R}(\cdot)$  and  $\mathcal{I}(\cdot)$  denote its real and imaginary parts.  $\gamma$  controls the sharpness of the frequency distance  
291 and is set to 1 by default.  $(u, v)$  indexes the frequency domain. In addition, LDP enforces cycle  
292 consistency by reconstructing the LR image from the SR output and minimizing a symmetric loss:  
293

$$\mathcal{L}_{sym}^{FT} = \lambda_1 \mathcal{L}_1(M' \otimes y', M' \otimes y) + \lambda_2 \mathcal{L}_{LPIPS}(M' \otimes y', M' \otimes y) + \lambda_3 \mathcal{L}_{fre}(M' \otimes y', M' \otimes y), \quad (16)$$

294 where  $M' = \tau \cdot M$ ,  $\tau$  scales the high-frequency weight map  $M$  by a scalar  $\tau$ .  
295

296 **Diffusion Posterior Sampling with LDP.** Our LDP can also be applied during inference in diffusion  
297 models via Diffusion Posterior Sampling (DPS) Chung et al. (2023), which uses the gradient of a  
298 data fidelity term to guide sampling and better align the results with the LR input:  
299

$$\nabla_{x_t} \log p_t(x_t | y) \simeq s_{\theta^*}(x_t, t) - \rho \nabla_{x_t} \mathcal{L}_{sym}^{FT}(LDP(\hat{x}_0, y_{hf}), y), \quad (17)$$

300 where  $s_{\theta^*}(x_t, t)$  denotes the score function (the noise predictor in DDPM Ho et al. (2020)), and  
301  $LDP(\cdot)$  represents our LDP degradation model.  $\hat{x}_0$  denotes the predicted clean image at each time  
302 step, and we treat it as the SR output. In latent diffusion models,  $\hat{x}_0$  is first decoded into the pixel  
303 space before computing the gradient.  
304

305 4 EXPERIMENT  
306308 4.1 IMPLEMENTATION DETAILS  
309

310 **Training LDP.** We train LDP on LSDIR Li et al. (2023) dataset using BSRGAN Zhang et al. (2021a)  
311 to synthesize diverse degradation datasets. For a scale factor of  $s = 4$ , the key hyperparameters are  
312  $s' = 2$ ,  $L = 3$ ,  $P = 16$ ,  $N_p = 32$ ,  $\lambda_1 = \lambda_2 = 1$ , and  $C = 64$ , resulting in 642k parameters. We  
313 use the Adam Kingma & Ba (2015) optimizer with  $\beta_1 = 0.9$  and  $\beta_2 = 0.99$ , with a fixed learning  
314 rate of 0.001. The batch size is 12, with  $256 \times 256$  HR patches. The timesteps  $t_i$  are sampled  
315 from  $[500, 1000]$  to align the noisy HR and LR features. We adopt the diffusion batch multiplier Li  
316 et al. (2024b) with a value of 4 to perform multiple noise realizations per HR image. Training is  
317 conducted on a single NVIDIA RTX A6000 for 60K iterations, taking approximately 16 hours.  
318

319 **Fine-Tuning SR Models.** We fine-tune existing SR models on the DF2K dataset (DIV2K Agustsson  
320 & Timofte (2017) and Flickr2K Lim et al. (2017)) using BSRGAN degradation patterns, with our  
321 LDP employed as an auxiliary loss. Details are provided in the Appendix D.  
322

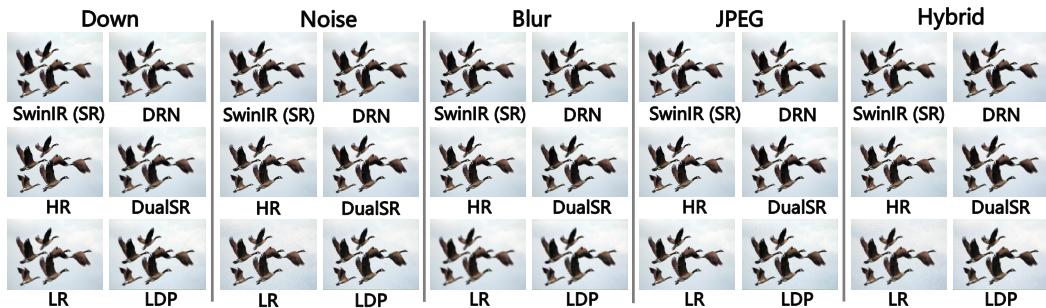
323 **Testing.** For synthetic testing, we generate five distinct datasets from the DIV2K validation set  
324 using `bsrgan_plus` (BSRGAN Zhang et al. (2021a) and Real-ESRGAN Wang et al. (2021)),  
325 corresponding to the following degradation types: (1) downsampling, (2) noise, (3) blur, (4) JPEG  
326 compression, and (5) hybrid degradations following `bsrgan_plus` defaults. For real-world testing,  
327

324  
325  
326  
327  
Table 1: Performance of multiple degra-  
328 dation models in LR prediction on synthetic  
329 multi-degradation datasets.

Methods	Metrics	Down	Noise	Blur	JPEG	Hybrid
<b>DRN</b>	PSNR↑	<b>32.05</b>	<b>27.25</b>	26.38	<b>29.65</b>	27.03
	SSIM↑	<b>0.9539</b>	0.7812	0.8273	<b>0.9270</b>	0.8098
	LPIPS↓	<b>0.0794</b>	0.2474	0.3207	<b>0.0826</b>	0.3360
<b>DualSR</b>	PSNR↑	19.58	18.77	19.36	18.57	19.36
	SSIM↑	0.4814	0.4712	0.4911	0.4612	0.4883
	LPIPS↓	0.1408	0.1399	0.1844	0.1492	0.2130
<b>LDP</b>	PSNR↑	29.15	26.71	<b>28.41</b>	28.01	<b>27.94</b>
	SSIM↑	0.9283	<b>0.8978</b>	<b>0.9159</b>	0.9243	<b>0.9173</b>
	LPIPS↓	0.0985	<b>0.1248</b>	<b>0.1417</b>	0.0877	<b>0.1025</b>

330  
331  
332  
333  
334  
335  
Table 2: Similarity between the LR images  
336 generated by multiple degradation models  
337 and the downsampled SR images.

Methods	Metrics	Down	Noise	Blur	JPEG	Hybrid
<b>DRN</b>	PSNR↑	<b>34.02</b>	<b>31.57</b>	<b>34.99</b>	<b>31.35</b>	<b>35.10</b>
	SSIM↑	<b>0.9638</b>	<b>0.9590</b>	<b>0.9692</b>	<b>0.9587</b>	<b>0.9679</b>
	LPIPS↓	<b>0.0365</b>	<b>0.0436</b>	<b>0.0306</b>	<b>0.0467</b>	<b>0.0296</b>
<b>DualSR</b>	PSNR↑	22.58	20.79	22.57	20.46	22.85
	SSIM↑	0.6689	0.6502	0.7044	0.6356	0.7164
	LPIPS↓	0.1264	0.1040	0.1262	0.1279	0.1175
<b>LDP</b>	PSNR↑	28.41	25.93	25.04	27.42	26.28
	SSIM↑	0.8895	0.7508	0.7596	0.8886	0.7597
	LPIPS↓	0.1551	0.3043	0.3278	0.1293	0.3586



346  
347 Figure 3: Qualitative results of multiple degradations models for LR prediction on synthetic datasets.  
348 **(Zoom in for details)**

349  
350  
351 we evaluate on RealSR Cai et al. (2019), RealSRSet Zhang et al. (2021b), and DPED Ignatov et al.  
352 (2017) datasets. We evaluate using PSNR, SSIM Wang et al. (2004), and LPIPS Zhang et al. (2018)  
353 as reference metrics, and NIQE Mittal et al. (2012), MANIQA Yang et al. (2022), CLIPQA Wang  
354 et al. (2023a), MUSIQ Ke et al. (2021), and QAlign Wu et al. (2024) as non-reference metrics. For  
355 diffusion models, synthetic datasets are center-cropped to  $512 \times 512$ , and real-world datasets follow  
356 the StableSR Wang et al. (2024).

## 358 4.2 EFFECTIVENESS OF LDP IN LR PREDICTION

359  
360 To thoroughly evaluate the effectiveness of the proposed LDP, we conduct extensive experiments  
361 under five degradation scenarios and compare it with two existing degradation models, DRN Guo  
362 et al. (2020) and DualSR Emad et al. (2021). In this experiment, we first generate SR images us-  
363 ing SwinIR Liang et al. (2021), and then apply the degradation models provided by LDP, DRN,  
364 and DualSR to obtain predicted LR images from the SR outputs. These predictions are compared  
365 with the LR inputs to the SR model, and the results are reported in Table 1. In addition, Table 2  
366 reports the similarity between the LR images produced by each degradation model and the down-  
367 sampled SR images. A higher similarity indicates that the degradation model collapses into trivial  
368 downscaling rather than applying the specific degradations implied by the input LR. As shown in  
369 the tables, LDP performs consistently well across all degradation types. Importantly, the similarity  
370 between the LDP-generated LR and the downsampled SR is significantly lower than that between  
371 the LDP-generated LR and the input LR, demonstrating that LDP does not degenerate into simple  
372 downscaling. In contrast, DRN behaves almost identically to bicubic downscaling: because its  
373 inputs include only HR (SR results) images without any conditional signals, it fails to map an SR  
374 image to the multiple possible LR variants implied by different degradations. DualSR also strug-  
375 gles to properly handle diverse degradation types, particularly under complex mixed settings. As  
376 illustrated in **Fig. 3**, LDP effectively degrades high-frequency structures, further validating its abil-  
377 ity to generate perceptually realistic LR images even under challenging degradations. In contrast,  
378 DRN and DualSR largely produce LR outputs that resemble simple downsampled versions of the  
379 SR images, indicating that they fail to apply the intended degradations.

378  
379  
380  
381  
382  
383 Table 3: Performance improvements of blind SR models across diverse architectures using our  
384 proposed LDP on synthetic multi-degradation benchmarks. We generate synthetic benchmarks from  
385 the DIV2K validation set using five types of degradation: (1) Downsampling (Down), (2) Noise, (3)  
386 Blur, (4) JPEG, and (5) Hybrid degradations following bsrgan\_plus defaults.

Datasets	Scale	Metrics	FeMaSR	+LDP	StableSR	+LDP	SwinIR	+LDP	MambaIR	+LDP
Down	×4	PSNR↑	24.22	<b>25.06 (+0.84)</b>	20.35	<b>21.73 (+1.38)</b>	25.44	<b>25.86 (+0.42)</b>	26.58	<b>26.63 (+0.05)</b>
	×4	SSIM↑	0.6793	<b>0.7105 (+0.0312)</b>	0.4998	<b>0.5642 (+0.0644)</b>	0.7210	<b>0.7242 (+0.0032)</b>	0.7393	<b>0.7403 (+0.0010)</b>
	×4	LPIPS↓	0.2637	<b>0.2490 (-0.0147)</b>	0.3746	<b>0.2870 (-0.0876)</b>	0.2579	<b>0.2538 (-0.0041)</b>	0.2054	<b>0.2005 (-0.0049)</b>
Noise	×4	PSNR↑	22.82	<b>23.84 (+1.02)</b>	19.95	<b>21.48 (+1.53)</b>	24.34	<b>25.04 (+0.70)</b>	26.11	<b>26.34 (+0.23)</b>
	×4	SSIM↑	0.6519	<b>0.6957 (+0.0438)</b>	0.4569	<b>0.5599 (+0.1030)</b>	0.7130	<b>0.7198 (+0.0068)</b>	0.7382	<b>0.7411 (+0.0029)</b>
	×4	LPIPS↓	0.2788	<b>0.2624 (-0.0164)</b>	0.4279	<b>0.3040 (-0.1239)</b>	0.2676	<b>0.2659 (-0.0017)</b>	0.2279	<b>0.2219 (-0.0060)</b>
Blur	×4	PSNR↑	24.12	<b>24.42 (+0.30)</b>	19.98	<b>21.50 (+1.52)</b>	24.03	<b>24.67 (+0.64)</b>	24.99	<b>25.33 (+0.34)</b>
	×4	SSIM↑	0.6639	<b>0.6787 (+0.0148)</b>	0.4373	<b>0.5437 (+0.1064)</b>	0.6764	<b>0.6833 (+0.0069)</b>	0.6892	<b>0.6942 (+0.0050)</b>
	×4	LPIPS↓	<b>0.3168</b>	0.3199 <i>(+0.0031)</i>	0.5112	<b>0.4763 (-0.0349)</b>	0.3197	<b>0.3168 (-0.0029)</b>	0.2768	<b>0.2751 (-0.0017)</b>
JPEG	×4	PSNR↑	22.92	<b>23.87 (+0.95)</b>	20.17	<b>21.91 (+1.74)</b>	24.55	<b>25.27 (+0.72)</b>	26.36	<b>26.59 (+0.23)</b>
	×4	SSIM↑	0.6696	<b>0.7068 (+0.0372)</b>	0.5141	<b>0.5943 (+0.0802)</b>	0.7301	<b>0.7372 (+0.0071)</b>	0.7497	<b>0.7538 (+0.0041)</b>
	×4	LPIPS↓	0.2633	<b>0.2508 (-0.0125)</b>	0.3682	<b>0.2767 (-0.0915)</b>	0.2535	<b>0.2506 (-0.0029)</b>	0.2113	<b>0.2063 (-0.0050)</b>
Hybrid	×4	PSNR↑	23.40	<b>23.72 (+0.32)</b>	19.27	<b>21.43 (+2.16)</b>	23.52	<b>24.35 (+0.83)</b>	24.35	<b>24.71 (+0.36)</b>
	×4	SSIM↑	0.6211	<b>0.6392 (+0.0181)</b>	0.3656	<b>0.5197 (+0.1541)</b>	0.6458	<b>0.6492 (+0.0034)</b>	0.6587	<b>0.6636 (+0.0049)</b>
	×4	LPIPS↓	<b>0.3453</b>	0.3516 <i>(+0.0063)</i>	0.5727	<b>0.4461 (-0.1266)</b>	0.3634	<b>0.3571 (-0.0063)</b>	0.3244	<b>0.3210 (-0.0034)</b>

400 Figure 4: Qualitative results on synthetic datasets with  $\times 4$  scale factor. **(Zoom in for details)**  
401402 

### 4.3 IMPROVING EXISTING SR MODELS VIA FINE-TUNING WITH LDP

403 We evaluate LDP on Blind SR models, including the GAN-based FeMaSR Chen et al. (2022),  
404 Diffusion-based StableSR Wang et al. (2024), Transformer-based SwinIR Liang et al. (2021), and  
405 Mamba-based MambaIR Guo et al. (2024). In these experiments, LDP is applied only during the  
406 fine-tuning stage and is not used at inference.

407 **Improving SR Models on Synthetic Benchmarks.** Quantitative and qualitative results are pre-  
408 sented in Tab. 3 and Fig. 4 (Fig. 7 in **Appendix**). As listed in Tab. 3, incorporating LDP consistently  
409 improves all baseline models across all degradation types. Among them, MambaIR+LDP achieves  
410 the best overall performance. SwinIR and StableSR also benefit significantly from LDP. StableSR,  
411 in particular, shows substantial relative gains under challenging conditions such as Blur and Hybrid.  
412 These results highlight LDP’s effectiveness in narrowing the solution space via cycle consistency,  
413 enabling stronger generalization to unknown degradations. Although FeMaSR+LDP outperforms  
414 the original model in most metrics, its LPIPS values in Blur and Hybrid remain higher. As shown in  
415 Fig. 4, LDP effectively reduces GAN artifacts and corrects texture distortions, significantly impro-  
416 ving perceptual quality. The low LPIPS scores of the original FeMaSR are likely due to severe GAN  
417 artifacts misinterpreted as texture.

418 **Improving SR Models on Real-World Benchmarks.** Quantitative and qualitative results are pre-  
419 sented in Tab. 4 and Fig. 5 (Fig. 8 in **Appendix**). Table 4 shows that incorporating LDP consis-  
420 tently improves the performance of existing blind SR models across almost all datasets and metrics,  
421 demonstrating its enhanced generalization to unseen degradations. For FeMaSR, LDP suppresses  
422 GAN-induced artifacts, producing more stable, natural outputs. This can lower no-reference met-  
423 rics, e.g., the CLIPQA score drops on RealSR, as such metrics may favor visually striking but  
424 structurally inaccurate results. As shown in Fig. 5, the visual results explain the numerical improve-  
425 ments, with LDP mitigating ringing and GAN-induced artifacts, thereby enhancing visual fidelity  
426 and contributing to the better no-reference metrics scores.

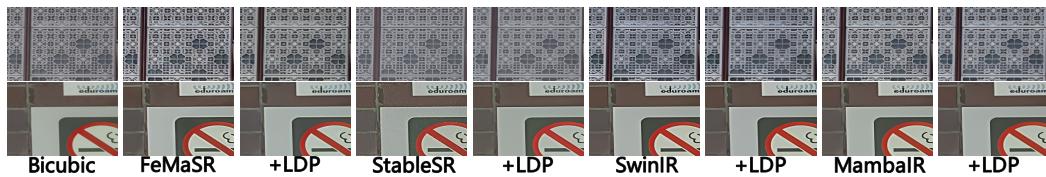
427 

### 4.4 LDP FOR POSTERIOR SAMPLING OF PRETRAINED DIFFUSION MODELS

428 We evaluated how LDP enhances pre-trained diffusion models through posterior sampling, in-  
429 cluding LDM Rombach et al. (2022), StableSR Wang et al. (2024), ResShift Yue et al. (2025),  
430 and UPSR Zhang et al. (2025). Quantitative and qualitative results are presented in Tab. 5 and

432 Table 4: Performance improvements of blind SR models across diverse architectures using our  
 433 proposed LDP on real-world benchmarks.  
 434

Datasets	Scale	Metrics	FeMaSR	+LDP	StableSR	+LDP	SwinIR	+LDP	MambaIR	+LDP
RealSR	×4	NIQE↓	<b>4.708</b>	5.533 (+0.825)	7.446	<b>6.331 (-1.115)</b>	<b>4.773</b>	4.838 (+0.065)	<b>5.330</b>	5.350 (+0.020)
	×4	MANIQA↑	0.3430	<b>0.3654 (+0.0224)</b>	0.3303	<b>0.3548 (+0.0245)</b>	0.3510	<b>0.3742 (+0.0232)</b>	0.2882	<b>0.3374 (+0.0492)</b>
	×4	CLIPQA↑	<b>0.5645</b>	0.4482 (-0.1163)	0.4886	<b>0.5213 (+0.0237)</b>	0.4739	<b>0.5478 (+0.0739)</b>	0.3989	<b>0.4642 (+0.0653)</b>
	×4	MUSIQ↑	58.94	<b>60.70 (+1.76)</b>	52.99	<b>59.26 (+6.27)</b>	59.67	<b>61.91 (+2.24)</b>	51.87	<b>57.85 (+5.98)</b>
	×4	QAlign↑	3.695	<b>3.860 (+0.165)</b>	2.347	<b>2.646 (+0.299)</b>	3.820	<b>3.877 (+0.057)</b>	3.631	<b>3.766 (+0.135)</b>
DPED	×4	NIQE↓	<b>5.045</b>	5.704 (+0.659)	7.616	<b>7.228 (-0.388)</b>	4.982	<b>4.821 (-0.161)</b>	5.983	<b>5.430 (-0.553)</b>
	×4	MANIQA↑	<b>0.3102</b>	0.2719 (-0.0383)	<b>0.3056</b>	0.2970 (-0.0086)	0.2637	<b>0.2832 (+0.0195)</b>	0.2334	<b>0.2767 (+0.0433)</b>
	×4	CLIPQA↑	<b>0.5570</b>	0.3610 (-0.1960)	<b>0.3968</b>	0.3843 (-0.0125)	0.3402	<b>0.4538 (+0.1136)</b>	0.3083	<b>0.3850 (+0.0767)</b>
	×4	MUSIQ↑	<b>49.14</b>	44.07 (-5.07)	42.97	<b>45.08 (+2.11)</b>	42.10	<b>45.91 (+3.81)</b>	35.25	<b>44.64 (+9.39)</b>
	×4	QAlign↑	<b>3.429</b>	3.262 (-0.167)	2.033	<b>2.311 (+0.278)</b>	2.988	<b>3.090 (+0.102)</b>	3.192	<b>3.248 (+0.056)</b>
RealSRSet	×4	NIQE↓	<b>5.236</b>	5.952 (+0.716)	6.090	<b>5.586 (-0.504)</b>	<b>5.424</b>	5.441 (+0.017)	<b>5.726</b>	5.893 (+0.167)
	×4	MANIQA↑	<b>0.4006</b>	0.4002 (-0.0004)	0.3904	<b>0.4012 (+0.0108)</b>	0.3740	<b>0.3938 (+0.0198)</b>	0.2978	<b>0.3555 (+0.0577)</b>
	×4	CLIPQA↑	<b>0.6874</b>	0.5683 (-0.1191)	0.6057	<b>0.6214 (+0.0157)</b>	0.5843	<b>0.6376 (+0.0533)</b>	0.4793	<b>0.5428 (+0.0635)</b>
	×4	MUSIQ↑	<b>64.65</b>	64.07 (-0.58)	60.15	<b>62.84 (+2.69)</b>	63.60	<b>65.33 (+1.73)</b>	55.96	<b>61.28 (+5.32)</b>
	×4	QAlign↑	3.776	<b>3.870 (+0.094)</b>	2.916	<b>3.247 (+0.331)</b>	2.749	<b>3.322 (+0.573)</b>	3.434	<b>3.632 (+0.198)</b>



446 Figure 5: Qualitative results on real-world benchmarks with  $\times 4$  scale factor. (Zoom in for details)  
 447  
 448  
 449  
 450  
 451

452 Fig. 6 (Fig. 9 in **Appendix**). As listed in Tab. 5, after applying LDP, the baselines show improvements  
 453 across nearly all metrics on most datasets. For instance, StableSR demonstrates notable gains  
 454 in MANIQA, CLIPQA, and MUSIQ scores after applying LDP, while ResShift and UPSR also  
 455 achieve higher metric values in most cases. For StableSR, we applied the noise-subtraction tech-  
 456 nique (Appendix E), which accounts for the differences from Tab. 4. As showed in Fig. 6, our LDP  
 457 effectively reduces texture artifacts while preserving structural consistency.  
 458

## 5 ABLATION STUDY

463 In ablation study, we examine the loss components, patch size, frequency band selection, scale factor  
 464 for high-frequency acquisition, performance of LDP under severe degradations, and computational  
 465 burden of LDP. Further details are provided in Appendix F.  
 466

467 **Ablation of Loss Terms in the Fine-Tuning Stage.** Table 6 presents the impact of different loss  
 468 components in  $\mathcal{L}_{sym}^{FT}$  (Equ. 16) and  $\mathcal{L}_{fre}$  (Equ. 14) during fine-tuning of pretrained SwinIR models,  
 469 evaluated on the synthetic Hybrid dataset. In all experiments, we set  $\tau = 100$  and the weight of  
 470 each loss term is set to 1. All variants using any combination of the proposed losses outperform  
 471 the baseline. Incorporating both symmetric and frequency losses (LDPV5–LDPV7) consistently  
 472 improves perceptual quality (lower LPIPS) and reconstruction accuracy (higher PSNR and SSIM),  
 473 with LDPV7 achieving the best overall performance, highlighting the complementary nature of these  
 474 loss components. The LDP parameters can be universally configured as  $\tau = 100$  and  $\lambda_1 = \lambda_2 =$   
 475  $\lambda_3 = 1$  for any super-resolution model, leading to improved generalization performance.  
 476

477 **Ablation of the weight of  $\tau_{au}$ .** Table 7 presents the impact of different weight of  $\tau_{au}$  when fine-  
 478 tuning SwinIR. All values of  $\tau_{au}$  outperform the baseline, with  $\tau_{au} = 100$  achieving the best overall  
 479 performance.  
 480

## 6 LIMITATIONS AND CONCLUSION

482 We propose LDP, a lightweight denoising autoencoder plug-in. By integrating HR images and the  
 483 high-frequency component of LR, the model achieves realistic degradation modeling while main-  
 484 taining efficiency. Experiments show LDP significantly improves the generalization of existing SR  
 485 models on unseen degradations after fine-tuning, and enables test-time artifact correction. However,  
 486 LDP has two main limitations: (1) in posterior sampling, it lacks generative ability and only per-

Table 5: Improving Diffusion models via posterior sampling with LDP on real-world benchmarks.

Datasets	Scale	Metrics	LDM	+LDP	StableSR	+LDP	ResShift	+LDP	UPSR	+LDP
RealSR	×4	NIQE↓	<b>6.651</b>	6.830 ( <b>+0.179</b> )	5.948	<b>5.636 (-0.312)</b>	<b>8.021</b>	8.027 ( <b>+0.006</b> )	4.854	<b>4.834 (-0.020)</b>
	×4	MANIQA↑	<b>0.2904</b>	0.2810 ( <b>-0.0094</b> )	0.3552	<b>0.3644 (+0.0092)</b>	<b>0.3487</b>	0.3486 ( <b>-0.0001</b> )	0.3901	<b>0.3908 (+0.0007)</b>
	×4	CLIPQA↑	<b>0.4564</b>	0.4319 ( <b>-0.0245</b> )	0.4840	<b>0.5031 (+0.0191)</b>	0.5353	<b>0.5354 (+0.0001)</b>	0.5278	<b>0.5361 (+0.0083)</b>
	×4	MUSIQ↑	<b>52.09</b>	50.37 ( <b>-1.72</b> )	55.11	<b>56.56 (+1.45)</b>	56.85	56.85	<b>64.82</b>	64.70 ( <b>-0.12</b> )
	×4	QAlign↑	<b>2.685</b>	2.610 ( <b>-0.075</b> )	2.607	<b>2.716 (+0.109)</b>	3.036	3.036	3.218	<b>3.231 (+0.013)</b>
DPED	×4	NIQE↓	<b>8.724</b>	8.770 ( <b>+0.046</b> )	6.456	<b>6.267 (-0.189)</b>	9.429	<b>9.415 (-0.014)</b>	<b>6.266</b>	6.281 ( <b>+0.015</b> )
	×4	MANIQA↑	0.2381	<b>0.2418 (+0.0037)</b>	0.3255	<b>0.3341 (+0.0086)</b>	<b>0.3107</b>	0.3104 ( <b>-0.0003</b> )	0.3151	<b>0.3163 (+0.0012)</b>
	×4	CLIPQA↑	<b>0.3718</b>	0.3681 ( <b>-0.0037</b> )	0.4041	<b>0.4053 (+0.0012)</b>	0.4875	<b>0.4879 (+0.0004)</b>	<b>0.4094</b>	0.4026 ( <b>-0.0068</b> )
	×4	MUSIQ↑	<b>32.92</b>	32.55 ( <b>-0.37</b> )	45.55	<b>49.25 (+3.70)</b>	<b>44.63</b>	44.59 ( <b>-0.04</b> )	46.47	<b>46.52 (+0.05)</b>
	×4	QAlign↑	1.901	<b>1.917 (+0.016)</b>	2.302	<b>2.343 (+0.041)</b>	2.422	<b>2.423 (+0.001)</b>	<b>2.271</b>	2.257 ( <b>-0.014</b> )
RealSRSet	×4	NIQE↓	6.349	<b>6.258 (-0.091)</b>	4.898	<b>4.687 (-0.211)</b>	<b>6.979</b>	7.011 ( <b>+0.032</b> )	<b>4.864</b>	4.878 ( <b>+0.014</b> )
	×4	MANIQA↑	0.3407	<b>0.3470 (+0.0063)</b>	0.4411	<b>0.4573 (+0.0162)</b>	0.4004	0.4004	0.4647	<b>0.4720 (+0.0073)</b>
	×4	CLIPQA↑	<b>0.5439</b>	0.5311 ( <b>-0.0128</b> )	0.6384	<b>0.6584 (+0.0200)</b>	0.6656	<b>0.6658 (+0.0002)</b>	0.6709	<b>0.6753 (+0.0044)</b>
	×4	MUSIQ↑	58.54	<b>59.52 (+0.98)</b>	62.73	<b>62.96 (+0.23)</b>	66.05	<b>66.06 (+0.01)</b>	69.68	<b>69.74 (+0.06)</b>
	×4	QAlign↑	3.046	<b>3.089 (+0.043)</b>	<b>3.193</b>	3.192 ( <b>-0.001</b> )	<b>3.561</b>	3.560 ( <b>-0.001</b> )	<b>3.705</b>	3.656 ( <b>-0.049</b> )

Figure 6: Qualitative results of LDP enhances diffusion models through posterior sampling at  $\times 4$  scale SR. (Zoom in for details)

Table 6: Ablation study of the loss terms used in the fine-tuning stage of pretrained SwinIR models.

Methods	$\mathcal{L}_1^{Sym}$	$\mathcal{L}_{LPIPS}^{Sym}$	$\mathcal{L}_{fre}^{Sym}$	$\mathcal{L}_{fre}^{SR}$	PSNR↑	SSIM↑	LPIPS↓
<b>baseline</b>	×	×	×	×	23.52	0.6458	0.3634
<b>LDPV1</b>	×	×	×	✓	23.99	0.6481	0.3591
<b>LDPV2</b>	✓	✓	×	×	24.08	0.6406	0.3585
<b>LDPV3</b>	×	×	✓	×	24.01	0.6404	0.3582
<b>LDPV4</b>	✓	✓	✓	×	24.13	0.6406	0.3609
<b>LDPV5</b>	✓	✓	×	✓	24.33	0.6499	0.3578
<b>LDPV6</b>	×	×	✓	✓	24.28	<b>0.6500</b>	0.3580
<b>LDPV7</b>	✓	✓	✓	✓	<b>24.35</b>	0.6492	<b>0.3571</b>

forms texture rectification; (2) It does not support unpaired degradation modeling, as the generated LR image inevitably retains information from the input LR high-frequency components.

## REFERENCES

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1122–1131, 2017.

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. In *Advances in Neural Information Processing Systems*, 2023.

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 *IEEE International Conference on Computer Vision*, pp. 3086–3095, 2019.

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 *ACM International Conference on Multimedia*, pp. 1329–1338, 2022.

Table 7: Ablation study of the  $\tau$  weight.

tau	PSNR↑	SSIM↑	LPIPS↓
-	23.52	0.6458	0.3634
0.1	24.15	<b>0.6547</b>	0.3601
1	24.27	<b>0.6547</b>	0.3595
10	24.30	0.6500	0.3596
100	<b>24.35</b>	0.6492	<b>0.3571</b>

540 Haoyu Chen, Wenbo Li, Jinjin Gu, Jingjing Ren, Haoze Sun, Xueyi Zou, Zhensong Zhang, Youliang Yan, and  
 541 Lei Zhu. Low-res leads the way: Improving generalization for super-resolution by self-supervised learning.  
 542 In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 25857–25867, 2024.

543 Honggang Chen, Xiaohai He, Hong Yang, Yuanyuan Wu, Linbo Qing, and Ray E. Sheriff. Self-supervised  
 544 cycle-consistent learning for scale-arbitrary real-world single image super-resolution. *Expert Systems with*  
 545 *Applications*, 212:118657, 2023a.

546 Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-  
 547 resolution transformer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 22367–22377,  
 548 2023b.

549 Yuying Chen, Mingde Yao, Wenbo Li, Renjing Pei, Jingjing Zhao, and Wenqi Ren. Unsupervised diffusion-  
 550 based degradation modeling for real-world super-resolution. In *Proceedings of the AAAI Conference on*  
 551 *Artificial Intelligence*, pp. 2348–2356, 2025.

552 Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: conditioning  
 553 method for denoising diffusion probabilistic models. In *IEEE International Conference on Computer Vision*,  
 554 pp. 14347–14356, 2021.

555 Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse  
 556 problems using manifold constraints. In *Advances in Neural Information Processing Systems*, 2022.

557 Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. Dif-  
 558 fusion posterior sampling for general noisy inverse problems. In *International Conference on Learning*  
 559 *Representations*, 2023.

560 Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for  
 561 image super-resolution. In *European conference on computer vision*, pp. 184–199, 2014.

562 Runmin Dong, Shuai Yuan, Bin Luo, Mengxuan Chen, Jinxiao Zhang, Lixian Zhang, Weijia Li, Juepeng  
 563 Zheng, and Haohuan Fu. Building bridges across spatial and temporal resolutions: Reference-based super-  
 564 resolution via change priors and conditional diffusion model. In *IEEE Conference on Computer Vision and*  
 565 *Pattern Recognition*, pp. 27674–27684, 2024.

566 Mohammad Emad, Maurice Peemen, and Henk Corporaal. DualSR: Zero-shot dual learning for real-world  
 567 super-resolution. In *IEEE Winter Conference on Applications of Computer Vision*, pp. 1629–1638, 2021.

568 Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. MambaIR: A simple baseline  
 569 for image restoration with state-space model. In *European conference on computer vision*, pp. 222–241,  
 570 2024.

571 Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mam-  
 572 bairv2: Attentive state space restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*,  
 573 2025.

574 Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezheng Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan.  
 575 Closed-loop matters: Dual regression networks for single image super-resolution. In *IEEE Conference on*  
 576 *Computer Vision and Pattern Recognition*, pp. 5406–5415, 2020.

577 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural*  
 578 *Information Processing Systems*, 2020.

579 Shady Abu Hussein, Tom Tirer, and Raja Giryes. Correction filter for single image super-resolution: Robusti-  
 580 fying off-the-shelf deep super-resolvers. In *IEEE Conference on Computer Vision and Pattern Recognition*,  
 581 pp. 1425–1434, 2020.

582 Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. DSLR-Quality photos  
 583 on mobile devices with deep convolutional networks. In *IEEE International Conference on Computer Vision*,  
 584 pp. 3297–3305, 2017.

585 Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: multi-scale image quality  
 586 transformer. In *IEEE/CVF International Conference on Computer Vision*, pp. 5128–5137, 2021.

587 Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference*  
 588 *on Learning Represent*, 2015.

589 Guangyuan Li, Chen Rao, Juncheng Mo, Zhanjie Zhang, Wei Xing, and Lei Zhao. Rethinking diffusion model  
 590 for multi-contrast MRI super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*,  
 591 pp. 11365–11374, 2024a.

594 Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation  
 595 without vector quantization. In *Advances in Neural Information Processing Systems*, 2024b.

596

597 Xiaoming Li, Chaofeng Chen, Xianhui Lin, Wangmeng Zuo, and Lei Zhang. From face to natural image:  
 598 Learning real degradation for blind image super-resolution. In *European conference on computer vision*, pp.  
 599 376–392, 2022.

600 Yawei Li, Kai Zhang, Jingyun Liang, Jie Zhang, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu,  
 601 Denis Demanolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. LSDIR: A large scale dataset for  
 602 image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1775–  
 603 1787, 2023.

604 Jingyun Liang, Jie Zhang, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image  
 605 restoration using swin transformer. In *IEEE International Conference on Computer Vision*, pp. 1833–1844,  
 606 2021.

607 Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual net-  
 608 works for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition  
 609 Workshops*, pp. 136–144, 2017.

610 Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a “completely blind” image quality analyzer.  
 611 *IEEE Signal Processing Letters*, 20(3):209–212, 2012.

612 Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning  
 613 with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp.  
 614 3942–3951, 2018.

615 Vaishnav Potlapalli, Syed Waqas Zamir, Salman H. Khan, and Fahad Shahbaz Khan. Promptir: Prompting for  
 616 all-in-one image restoration. In *Advances in Neural Information Processing Systems*, 2023.

617

618 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution im-  
 619 age synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*,  
 620 pp. 10674–10685, 2022.

621

622 Assaf Shocher, Nadav Cohen, and Michal Irani. ”Zero-Shot” super-resolution using deep internal learning. In  
 623 *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3118–3126, 2018.

624

625 Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International  
 626 Conference on Learning Representations*, 2021.

627

628 Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. In *IEEE Conference on Computer  
 629 Vision and Pattern Recognition*, pp. 9446–9454, 2018.

630

631 Jianyi Wang, Kelvin C. K. Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of  
 632 images. In *AAAI Conference on Artificial Intelligence*, pp. 2555–2563, 2023a.

633

634 Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin C. K. Chan, and Chen Change Loy. Exploiting diffusion  
 635 prior for real-world image super-resolution. *International Journal of Computer vision*, 2024.

636

637 Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-  
 638 resolution with pure synthetic data. In *International Conference on Computer Vision Workshops*, pp. 1905–  
 639 1914, 2021.

640

641 Zhixin Wang, Ziyi Zhang, Xiaoyun Zhang, Huangjie Zheng, Mingyuan Zhou, Ya Zhang, and Yanfeng Wang.  
 642 DR2: diffusion-based robust degradation remover for blind face restoration. In *IEEE Conference on Com-  
 643 puter Vision and Pattern Recognition*, pp. 1704–1713, 2023b.

644

645 Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error  
 646 visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

647

648 Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Chunyi Li, Liang Liao, Annan Wang, Erli Zhang,  
 649 Wenxiu Sun, Qiong Yan, Xiongkuo Min, Guangtai Zhai, and Weisi Lin. Q-align: Teaching Lmms for visual  
 650 scoring via discrete text-defined levels. In *International Conference on Machine Learning*, 2024.

651

652 Jiahao Xie, Wei Li, Xiaohang Zhan, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Masked frequency  
 653 modeling for self-supervised visual pre-training. In *International Conference on Learning Representations*,  
 654 2023.

648 Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang.  
 649 Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1191–1200, 2022.  
 650

651 Zhicun Yin, Ming Liu, Xiaoming Li, Hui Yang, Longan Xiao, and Wangmeng Zuo. MetaF2N: Blind image  
 652 super-resolution by learning efficient model adaptation from faces. In *IEEE International Conference on  
 653 Computer Vision*, pp. 12987–12998, 2023.  
 654

655 Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image  
 656 super-resolution using cycle-in-cycle generative adversarial networks. In *IEEE International Conference on  
 657 Computer Vision Workshops*, pp. 701–710, 2018.  
 658

659 Zongsheng Yue, Qian Zhao, Jianwen Xie, Lei Zhang, Deyu Meng, and Kwan-Yee K. Wong. Blind image super-  
 660 resolution with elaborate degradation modeling on noise and kernel. In *IEEE Conference on Computer Vision  
 661 and Pattern Recognition*, pp. 2118–2128, 2022.  
 662

663 Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Efficient diffusion model for image restoration by residual  
 664 shifting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(1):116–130, 2025.  
 665

666 Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for  
 667 deep blind image super-resolution. In *IEEE International Conference on Computer Vision*, pp. 4791–4800,  
 668 2021a.  
 669

670 Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for  
 671 deep blind image super-resolution. In *IEEE International Conference on Computer Vision*, pp. 4791–4800,  
 672 2021b.  
 673

674 Leheng Zhang, Weiyi You, Kexuan Shi, and Shuhang Gu. Uncertainty-guided perturbation for image super-  
 675 resolution diffusion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.  
 676 17980–17989, 2025.  
 677

678 Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness  
 679 of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*,  
 680 pp. 586–595, 2018.  
 681

682 Ruofan Zhang, Jinjin Gu, Haoyu Chen, Chao Dong, Yulun Zhang, and Wenming Yang. Crafting training degra-  
 683 dation distribution for the accuracy-generalization trade-off in real-world super-resolution. In *International  
 684 Conference on Machine Learning*, pp. 41078–41091, 2023.  
 685

686 Hongyang Zhou, Xiaobin Zhu, Jianqing Zhu, Zheng Han, Shi-Xue Zhang, Jingyan Qin, and Xu-Cheng Yin.  
 687 Learning correction filter via degradation-adaptive regression for blind single image super-resolution. In  
 688 *IEEE International Conference on Computer Vision*, pp. 12331–12341, 2023.  
 689

## 690 A THE USE OF LARGE LANGUAGE MODELS

691 We used Large Language Models to assist or polish the writing, without involving our experiments,  
 692 figures, or other core contributions.

## 693 B ANONYMIZED LINK TO OUR CODE

694 Our code is available on an anonymous link for open-source access  
<https://anonymous.4open.science/r/LDP-3CAC/>.

## 695 C CREATION OF SYNTHETIC TESTING DATASETS

696 We adopt the `bsrgan_plus` degradation model Zhang et al. (2021a); Wang et al. (2021) to con-  
 697 struct synthetic multi-degradation datasets from the DIV2K validation set. Specifically, the full  
 698 `bsrgan_plus` pipeline is used to generate the hybrid degradation dataset, while four individual  
 699 datasets (Downsample, Blur, Noise, and JPEG) are created by applying only the corresponding  
 700 components of `bsrgan_plus`.

701 **Downsample.** For the downsample mode, four types of interpolation methods are employed:  
 $D_{nearest}^s$ ,  $D_{bilinear}^s$ ,  $D_{bicubic}^s$  and  $D_{down-up}^s$ , where  $s$  is the scale factor. For the  $D_{nearest}^s$  method,

702 there is a probability that a centered  $21 \times 21$  isotropic Gaussian kernel is shifted by  $0.5 \times (s-1)$  pixels  
 703 using a 2D linear grid interpolation technique. This step is taken to correct a potential misalignment  
 704 of  $0.5 \times (s-1)$  pixels towards the upper-left corner that may occur during the downsampling pro-  
 705 cess. In the  $D_{down-up}^s = D_{down}^{s/a} D_{up}^a$ , the HR image is first downsampled by a scale factor of  $s/a$   
 706 and then upsampled by a scale factor of  $a$ . The interpolation methods for both downsampling and  
 707 upsampling are randomly selected from nearest neighbor, bilinear, or bicubic interpolation. Addi-  
 708 tionally, with a probability of 0.25, the HR image is initially resized to half of its original dimensions  
 709 using a randomly selected interpolation technique. Following this resizing,  $s$  is set to  $s/2$  for the  
 710 subsequent downsampling operation.

711 **Noise.** For the noise mode, a shuffle order of 5 operations is generated. These operations include:  
 712 (1) Gaussian noise with a standard deviation in [2, 25], including grayscale, multivariate, and color  
 713 variants with probabilities of 0.4, 0.2, and 0.4, respectively; (2) Speckle noise, applied multiplica-  
 714 tively with the same probability setting as Gaussian noise; (3) Poisson noise, added either globally  
 715 or in grayscale with equal probability after scaling and rounding; (4) JPEG compression with a ran-  
 716 dom quality factor in [30, 95]; and (5) downsampling by a factor of  $s$  using a randomly selected  
 717 interpolation method (nearest, bilinear, or bicubic).

718 **Blur.** For the blur mode, a random sequence of two operations is applied: (1) blurring the image  
 719 twice using randomly generated kernels with scale factor  $s$ , with a 50% chance of selecting an  
 720 anisotropic Gaussian kernel, and otherwise using an isotropic Gaussian kernel, with kernel size and  
 721 width also randomized; and (2) downsampling by a factor of  $s$  using a randomly chosen interpolation  
 722 method (nearest-neighbor, bilinear, or bicubic).

723 **JPEG.** For the JPEG mode, a random sequence of two operations is applied: (1) simulating JPEG  
 724 compression artifacts by converting the image to `uint8` format, compressing it using a randomly  
 725 sampled quality factor between 30 and 95, and then decompressing it; and (2) downsampling the  
 726 image by a factor of  $s$  using a randomly selected interpolation method (nearest-neighbor, bilinear,  
 727 or bicubic).

## 729 730 D DETAILS OF FINE-TUNING PRETRAINED SUPER-RESOLUTION MODELS

731 All pretrained SR models were obtained from their respective official GitHub repositories. Fine-  
 732 tuning was performed using the DF2K dataset, which combines DIV2K Agustsson & Timofte (2017)  
 733 and Flickr2K Lim et al. (2017), with BSRGAN Zhang et al. (2021a) employed as the degradation  
 734 model. In this setting, LDP is applied only during the fine-tuning stage and is not used at inference.

735 **FeMaSR.** We directly fine-tuned the second-stage model of FeMaSR using its original loss func-  
 736 tions: L1 loss, LPIPS loss, GAN loss, and a codebook-specific loss. In addition, we incorporated  
 737 the frequency loss  $\mathcal{L}_{fre}$  (Equ. 14) and the fine-tuning symmetry loss  $\mathcal{L}_{sym}^{FT}$  (Equ. 16). The hy-  
 738 perparameters were set as follows:  $\lambda_{fre} = 1$ ,  $\lambda_1 = \lambda_2 = \lambda_3 = 0.1$ , and  $\tau = 1$ . The model  
 739 was fine-tuned for 100,000 iterations. Notably, even a brief fine-tuning of 1,000 iterations signifi-  
 740 cantly reduces GAN-induced artifacts. Longer training durations allow the discriminator to better  
 741 converge, thereby enhancing the generation of realistic and detailed textures. Experimental results  
 742 demonstrate that when employing GAN loss, extended fine-tuning is typically necessary to ensure  
 743 stable convergence of the discriminator.

744 **StableSR.** To fine-tune StableSR, we follow the original loss settings with two additional loss terms:  
 745 the frequency loss  $\mathcal{L}_{fre}$  (Equ. 14) and the fine-tuning symmetry loss  $\mathcal{L}_{sym}^{FT}$  (Equ. 16). Since Sta-  
 746 bleSR is a latent diffusion model, it is necessary to use the decoder to transform the latent features  
 747 back into the RGB space. Specifically, at each diffusion step, based on DDPM Ho et al. (2020) or  
 748 DDIM Song et al. (2021), the model predicts the clean image  $\hat{x}_0$  from the noisy input. We first apply  
 749 the decoder to convert  $\hat{x}_0$  into a RGB image  $X'$ , which is then used to compute the frequency loss  
 750  $\mathcal{L}_{fre}(X', x)$  for frequency modulation. Subsequently,  $X'$  along with the high-frequency component  
 751 of LR  $y_{hf}$  is fed into our LDP module to generate a predicted LR image  $y'$ . We then apply the  
 752 symmetry loss  $\mathcal{L}_{sym}^{FT}(y', y)$  to further guide the super-resolution process. The hyper-parameters are  
 753 set  $\lambda_{fre} = 0.1$ ,  $\lambda_1 = \lambda_2 = \lambda_3 = 0.1$  and  $\tau = 1$ . The model was fine-tuned for 2,000 iterations.  
 754 The inference code is the same as the original StableSR with the DDPM step set as 200.

756 **SwinIR and MambaIR.** To fine-tune SwinIR and MambaIR, we use  $\mathcal{L}_1$ ,  $\mathcal{L}_{LPIPS}$  and  
 757  $\mathcal{L}_{fre}$  (Equ. 14) to constrain HR and SR result, while use  $\mathcal{L}_{sym}^{FT}$  to constrain LR and the predicted  
 758 LR from our LDP. The hyper-parameters are set  $\lambda_{fre} = 10$ ,  $\lambda_1 = \lambda_2 = \lambda_3 = 1$  and  $\tau = 100$ . The  
 759 models were fine-tuned for 1,000 iterations.  
 760

## 761 E DIFFUSION POSTERIOR SAMPLING WITH LDP 762

763 We evaluated how LDP enhances pre-trained diffusion models through posterior sampling, including  
 764 LDM Rombach et al. (2022), StableSR Wang et al. (2024), ResShift Yue et al. (2025), and  
 765 UPSR Zhang et al. (2025). Posterior sampling, as formulated in Eq. 17, is carried out without any  
 766 fine-tuning. In this setting, quantitative metrics may show limited improvement. However, visual re-  
 767 sults demonstrate a notable reduction in artifacts and enhanced fidelity in the outputs of the diffusion  
 768 models. For all four baseline models, the LDP parameters are set to  $\tau = 100$  and  $\lambda_1 = \lambda_2 = \lambda_3 = 1$ .  
 769 As all selected models are latent diffusion model, we should first use the Decoder to transfor the la-  
 770 tent feature back to the color space. Specifically, for every diffusion step, according DDPM Ho  
 771 et al. (2020) or DDIM Song et al. (2021), the model will get the predicted clean image  $\hat{x}_0$  from  
 772 the model output. We decode  $\hat{x}_0$  into the RGB image  $X'$ , which is then combined with the high-  
 773 frequency component of LR images  $y_{hf}$  and passed into our LDP module to generate a predicted  
 774 LR image  $y'$ . The fine-tuning symmetry loss  $\mathcal{L}_{sym}^{FT}(y', y)$  is subsequently applied to further guide  
 775 the super-resolution model.  
 776

776 **LDM.** We use the SR version of LDM with 50 DDIM steps, we apply LDP only every 5 steps  
 777 during the last 25 steps of the sampling process. This is because LDM has already undergone  
 778 super-resolution training, so the predicted clean image  $\hat{x}_0$  in the early stages of the DDPM process  
 779 are sufficiently close to the LR input. However, as the diffusion process progresses, the generated  
 780 SR images may gradually diverge from the LR features, thereby necessitating additional guidance.  
 781 Moreover, applying the DPS operation increases inference time. While applying it at every step  
 782 could further improve the fidelity of the generated results, the computational overhead becomes  
 783 prohibitive.  
 784

784 **StableSR.** We found that the SR result of StableSR exhibits a noticeable repeat-spot artifact, as il-  
 785 lustrated in Fig. 4 and Fig. 7. We note that the artifact can be removed by subtracting noise during  
 786 inference Bansal et al. (2023), a technique compatible with the inference process of StableSR. How-  
 787 ever, in our experiments, this artifact removal method was applied only in the posterior sampling  
 788 setting and not during inference with fine-tuned models. Specifically, we set  $P(x, t)$  as the noise  
 789 diffusion process at time  $t$ . In each denoising step, the update can be formulated as:  
 790

$$x_{t-1} = x_{t-1} - \lambda * (\mathbf{P}(\hat{x}_0, t) + \mathbf{P}(\hat{x}_0, t-1)), \quad (18)$$

791 where we set  $\lambda = 0.01$ . We adopt 200 DDPM steps, but our LDP are applied in the last 100 steps,  
 792 and only every 10 steps. For the same reasons as in LDM. We observe that applying LDP directly  
 793 to StableSR without this technique tends to exacerbate the repeat-spot artifact. In contrast, applying  
 794 the artifact removal prior to LDP further enhances StableSR’s performance. We hypothesize that  
 795 this is because StableSR possesses strong generative capability, producing super-resolved images  
 796 that deviate from the LR input. Consequently, when LDP is used to enforce consistency between  
 797 the SR and LR images, it may inadvertently suppress the model’s generative ability.  
 798

798 **ResShift.** We adopt the journal version of ResShift, requiring only four steps to generate SR results,  
 799 with LDP applied at each step.  
 800

800 **UPSR.** UPSR generates SR results in only five steps, with LDP applied at each step.  
 801

## 802 F EXTENDED ABLATION STUDY 803

804 **Ablation of the Patch Sise in Noise Addition Module.** Table 8 presents an ablation study investi-  
 805 gating the effect of patch size in the patch-wise noise addition process of diffusion. We systemati-  
 806 cally vary the patch size in 1, 4, 8, 16 and evaluate each configuration on the fine-tuning pretrained  
 807 SwinIR model (baseline) using the synthetic Hybrid dataset. Experimental results demonstrate that  
 808 any patch configuration surpasses the baseline. When the patch size equals one, it implies that uni-  
 809 form noise is added across the entire image. Since a patch size of 16 attains the highest PSNR and  
 the lowest LPIPS, we set  $P = 16$  in our LDP.  
 810

810  
811 Table 8: Ablation study of the Patch Size of  
812 LDP.

Methods	<i>patch</i>	PSNR↑	SSIM↑	LPIPS↓
<b>baseline</b>	-	23.52	0.6458	0.3634
<b>LDPp2</b>	1	24.43	0.6505	0.3567
<b>LDPp4</b>	4	24.45	0.6519	0.3567
<b>LDPp8</b>	8	24.34	<b>0.6520</b>	0.3572
<b>LDPp16</b>	16	<b>24.46</b>	0.6513	<b>0.3566</b>

813  
814 Table 9: Ablation study of the frequency band  
815 used in  $\mathcal{L}_{sym}^{FT}$ .

Methods	<b>DWT<sub>fre</sub></b>	PSNR↑	SSIM↑	LPIPS↓
<b>baseline</b>	baseline	23.52	0.6458	0.3634
<b>LDP<sub>LF</sub></b>	LL	<b>24.35</b>	0.6472	0.3573
<b>LDP<sub>HF</sub></b>	LH+HL+HH	<b>24.35</b>	<b>0.6492</b>	<b>0.3571</b>
<b>LDP<sub>ALL</sub></b>	ALL	24.33	0.6430	0.3574

820 **Ablation of the Frequency Band in  $\mathcal{L}_{sym}^{FT}$ .** Table 9 presents an ablation study on DWT frequency-  
821 band supervision. In this experiment, the pretrained SwinIR model (baseline) is fine-tuned and  
822 evaluated on the synthetic Hybrid dataset. The variants  $LDP_{LF}$ ,  $LDP_{HF}$ , and  $LDP_{ALL}$  apply  
823 supervision to the LL (low-frequency), LH/HL/HH (high-frequency), and all DWT sub-bands. Both  
824  $LDP_{LF}$  and  $LDP_{HF}$  improve PSNR from 23.52 to 24.35, with  $LDP_{HF}$  achieving slightly higher  
825 SSIM and the lowest LPIPS. In contrast,  $LDP_{ALL}$  yields comparable PSNR and LPIPS but slightly  
826 lower SSIM, suggesting that focused supervision on specific frequency bands is more effective than  
827 supervising all sub-bands indiscriminately.

828 Table 10: Ablation study of the scale factor in LR residual acquisition phase.

Methods	<i>s'</i>	PSNR↑	SSIM↑	LPIPS↓
<b>baseline</b>	-	23.52	0.6458	0.3634
<b>LDPsf2</b>	2	<b>24.35</b>	0.6492	<b>0.3571</b>
<b>LDPsf4</b>	4	24.31	0.6490	0.3576
<b>LDPsf8</b>	8	24.24	0.6495	0.3582
<b>LDPsf16</b>	16	24.21	<b>0.6496</b>	0.3585

830 **Ablation of the Scale Factor in the LR Residual Acquisition Phase.** To investigate how the scale  
831 factor  $s'$  affects performance, we conduct an ablation study by varying  $s'$  during the fine-tuning of  
832 a pretrained SwinIR model (baseline) using the synthetic Hybrid dataset. This factor determines  
833 the high-frequency components extracted from LR images. As listed in Tab. 10, all LDP variants  
834 outperform the baseline, with the best performance achieved at  $s' = 2$ . As  $s'$  increases, PSNR and  
835 LPIPS consistently decline, while SSIM steadily improves. This is because larger  $s'$  values introduce  
836 stronger but less reliable high-frequency components into the LDP input. These components may  
837 amplify edge-like patterns that enhance SSIM but do not faithfully reflect true HR details, thereby  
838 increasing prediction errors and perceptual inconsistencies. As a result, the quality of the supervision  
839 signal deteriorates, weakening the fine-tuning effectiveness and degrading overall SR performance.  
840 These findings highlight the importance of selecting an appropriate  $s'$  to balance structural sharpness  
841 and reconstruction fidelity.

842 **LDP contributions to existing SR models evaluated on severely degraded test dataset.** To  
843 evaluate our method on severely degraded LR images, we regard pretrained SwinIR as baseline  
844 and test SwinIR+LDP in our main text. We still use the bsrgan\_plus Zhang et al. (2021a); Wang  
845 et al. (2021) Zhang et al. (2021a); Wang et al. (2021) degradation setting, while changing the  
846 maximum length (wd2) of the Gaussian blur kernel (please refer to the bsrgan\_plus code), we set  
847  $wd2 \in [8, 16, 32, 64, 484]$ . By default,  $wd2$  is set to 8 and applied in the synthetic Hybrid dataset.  
848 As reported in Tab. 11, the results demonstrate that our method retains strong generalization  
849 under server Gaussian blur. Because the training signal is the residual obtained by subtracting the  
850 downsample-upsample from the original LR, the residual is never zero even when a heavy blur re-  
851 moves substantial high-frequency content, thereby providing a distinctive cue for identifying the  
852 LR corresponding to the same HR image. Tab. 12 further reports the LR-prediction evaluation of  
853 LDP. Even under severe blur ( $wd2 = 484$ ), the predicted LR remains highly consistent with its initial  
854 counterpart, achieving 26.87 dB PSNR, 0.89 SSIM, and 0.1618 LPIPS. These results underscore the  
855 robustness of our LDP.

856 **The computational burden of LDP when it is employed for posterior sampling.** We assess  
857 the integration of LDP into the diffusion posterior sampling Chung et al. (2023) framework built  
858 upon StableSR under the synthetic Hybrid dataset, where Eq. 18 is applied to suppress artifacts.

864  
865  
866  
867  
868  
869 Table 11: Ablation study on severely Gaussian  
870 blur degraded LR images.  $wd2$  is the maximum  
871 length of Gaussian blur in bsrgan\_plus degrada-  
872 tion process.  
873  
874  
875  
876  
877  
878  
879  
880

Methods	$wd2$	PSNR↑	SSIM↑	LPIPS↓
<b>baseline</b>	8	23.52	0.6458	0.3634
<b>+LDP</b>	8	<b>24.21</b>	<b>0.6496</b>	<b>0.3585</b>
<b>baseline</b>	16	22.99	0.6296	0.3974
<b>+LDP</b>	16	<b>23.78</b>	<b>0.6319</b>	<b>0.3932</b>
<b>baseline</b>	32	22.82	0.6188	0.3967
<b>+LDP</b>	32	<b>23.76</b>	<b>0.6238</b>	<b>0.3935</b>
<b>baseline</b>	64	22.27	0.5971	0.4363
<b>+LDP</b>	64	<b>23.15</b>	<b>0.6023</b>	<b>0.4341</b>
<b>baseline</b>	484	21.24	0.5740	0.4810
<b>+LDP</b>	484	<b>22.09</b>	<b>0.5812</b>	<b>0.4759</b>

881  
882 Table 13: Inference time of posterior sampling with LDP in Diffusion models and its impact on  
883 performance.  
884  
885  
886  
887  
888

per image (s)	baseline	LDPPtV1	LDPPtV2	LDPPtV3
Times	19	178	99	28
PSNR↑	19.71	<b>19.90</b>	19.72	19.72
SSIM↑	0.3756	<b>0.3848</b>	0.3718	0.3705
LPIPS↓	0.5118	<b>0.5020</b>	0.5115	0.5057

889 Four configurations are compared: (1) baseline: the original StableSR baseline with 200 DDPM  
890 denoising iterations; (2) LDPPtV1: LDP applied at every step across all 200 iterations; (3) LDPPtV2:  
891 LDP applied only during the last 100 iterations; and (4) LDPPtV3: LDP applied once every ten  
892 steps within the last 100 iterations. The quantitative results are reported in Tab. 13. Applying DPS  
893 at every step significantly improves the performance of diffusion models, but incurs prohibitive  
894 inference overhead. In contrast, applying LDP once every ten steps during the final 100 iterations  
895 introduces only a modest runtime increase, while still yielding performance gains over the baseline.  
896 We emphasize that no acceleration techniques such as half-precision were used during testing. All  
897 models were run in full precision on the GPU, and additional speed-ups may be achieved with  
898 alternative strategies.  
899

900  
901 Table 14: Comparison of training cost and efficiency between the proposed LDP and other plug-in  
902 methods.  
903  
904

Methods	GPU memory (MiB)	Time per Iteration (s)	PSNR↑	SSIM↑	LPIPS↓
<b>SwinIR</b>	15575	1.413	23.64	<b>0.6098</b>	0.4541
<b>SwinIR+LDP</b>	22405	2.094	<b>23.96</b>	0.6050	<b>0.4468</b>
<b>SwinIR+Lway</b>	200768	22.55	21.11	0.6024	0.5126

### 905 906 907 908 Evaluating training cost and efficiency of LDP against other plug-in methods.

909 We report the training cost and efficiency of incorporating LDP as a loss component of SwinIR  
910 under the synthetic Hybrid dataset, in comparison with Lway Chen et al. (2024). Since the official  
911 Lway code is not publicly available, we re-implemented it following their GitHub guidelines. Using  
912 Lway as a loss component is equivalent to the original Lway paper, where the pre-trained model  
913 is 100% fine-tuned. Three configurations are compared: (1) **SwinIR**: SwinIR trained from scratch  
914 with  $L_1 + L_{fre}$ ; (2) **S+LDP**: SwinIR trained with  $L_1 + L_{fre} + \mathcal{L}_{sym}^{FT}$ . Predicted LR comes from  
915 LDP. (2) **S+Lway**: SwinIR trained with  $L_1 + L_{fre} + \mathcal{L}_{sym}^{FT}$ . Predicted LR comes from Lway. As  
916 reported in Tab 14, LDP increases SwinIR’s training GPU memory from 15,575 MiB to 22,405  
917 MiB, extends the per-iteration runtime from 1.413 s to 2.094 s, and consequently raises the compute  
918 cost for 50,000 iterations from 21.23 h to 31.26 h. In exchange, PSNR and LPIPS improve, and  
919

920  
921  
922 Table 12: Performance of LDP in LR prediction  
923 on severely Gaussian-degraded LR images.  
924  
925  
926

$wd2$	PSNR↑	SSIM↑	LPIPS↓
8	29.81	0.9169	0.1009
16	27.61	0.9123	0.1231
32	27.51	0.9087	0.1215
64	27.32	0.9053	0.1325
484	26.87	0.8900	0.1618

918 SSIM changes marginally. In contrast, Lway does not improve model performance within the same  
919 training time and consumes even more GPU memory.  
920

## 921 G EXTENDED QUALITATIVE RESULTS 922

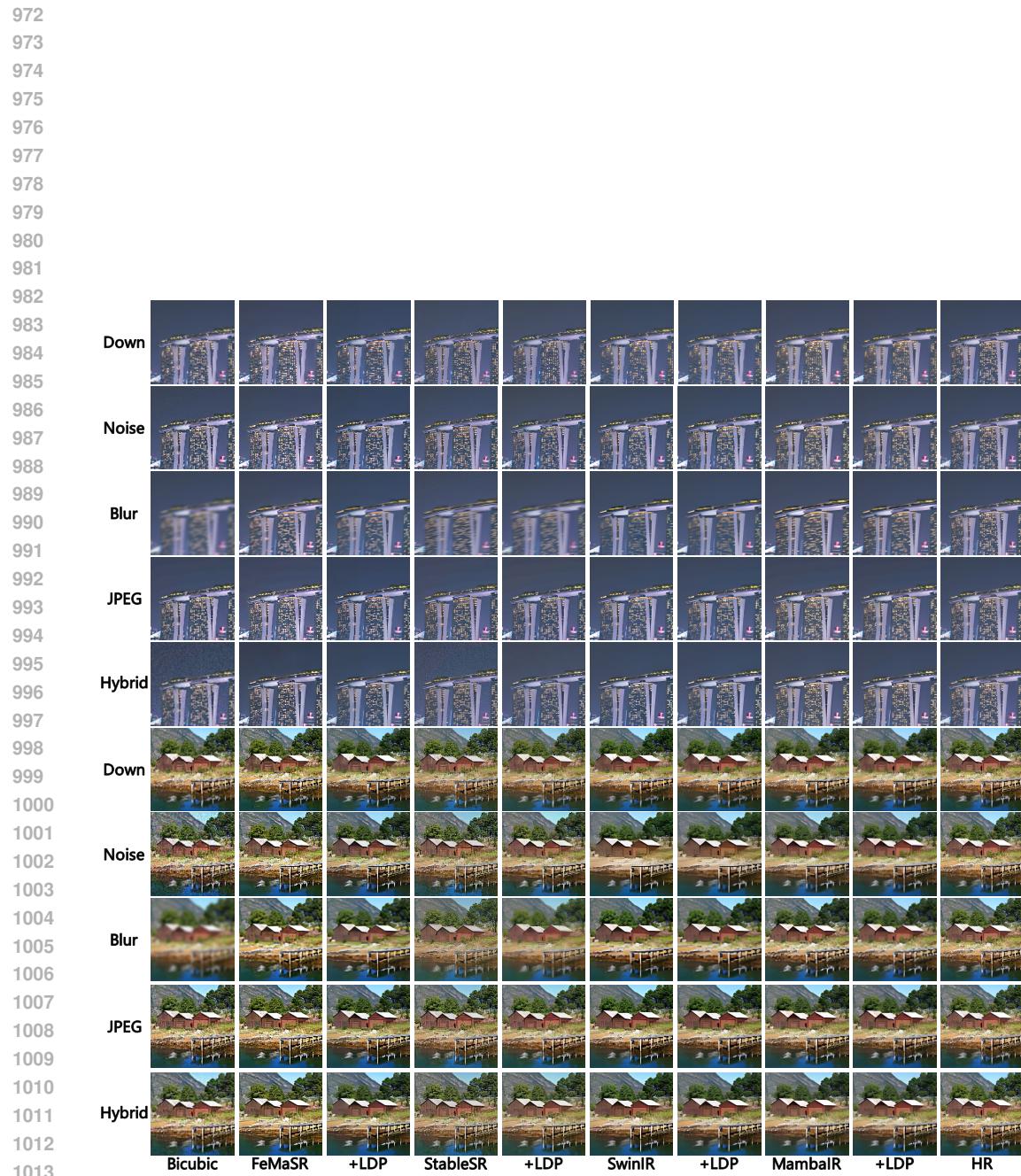
923 More visual results of the blind SR models on both synthetic and real-world benchmarks are pro-  
924 vided in Fig. 7 and Fig. 8, respectively. Additional qualitative results of diffusion posterior sampling  
925 are presented in Fig. 9. With the assistance of LDP, existing SR models demonstrate a clear abil-  
926 ity to suppress artifacts, preserve LR features, and generalize better to unseen degradation types.  
927 However, this approach also reveals a limitation: for models such as FeMaSR, which treat certain  
928 artifacts as part of the texture, LDP struggles to preserve the model’s original ability to generate  
929 detailed textures while removing artifacts. This highlights a trade-off between artifact suppression  
930 and texture fidelity in models that implicitly rely on artifact patterns for texture synthesis.  
931

## 932 H ETHICS STATEMENT 933

934 Our work focuses on single-image super-resolution and synthetic degradation modeling using pub-  
935 licly available or properly licensed images. No human subjects or sensitive personal data are in-  
936 volved. The LDP model is intended for research and image enhancement, and we acknowledge that  
937 generative image processing can be misused. We encourage responsible use and compliance with  
938 relevant legal and ethical guidelines.  
939

## 940 I REPRODUCIBILITY STATEMENT 941

942 Our code is provided in Appendix B. The training details of our proposed LDP are described in  
943 Section 4.1 of the main text. The generation process of the synthetic multi-degradation datasets  
944 is presented in Appendix C. Experimental details of fine-tuning existing SR models with LDP are  
945 given in Appendix D, while Appendix E provides the details of applying LDP for posterior sampling  
946 with pre-trained diffusion models.  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971



1014 Figure 7: Qualitative results on synthetic multi-degradation datasets with  $\times 4$  scale factor. (Zoom in  
1015 for details)

1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025

Figure 8: Qualitative results on real-world datasets with  $\times 4$  scale factor. (Zoom in for details)

1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079

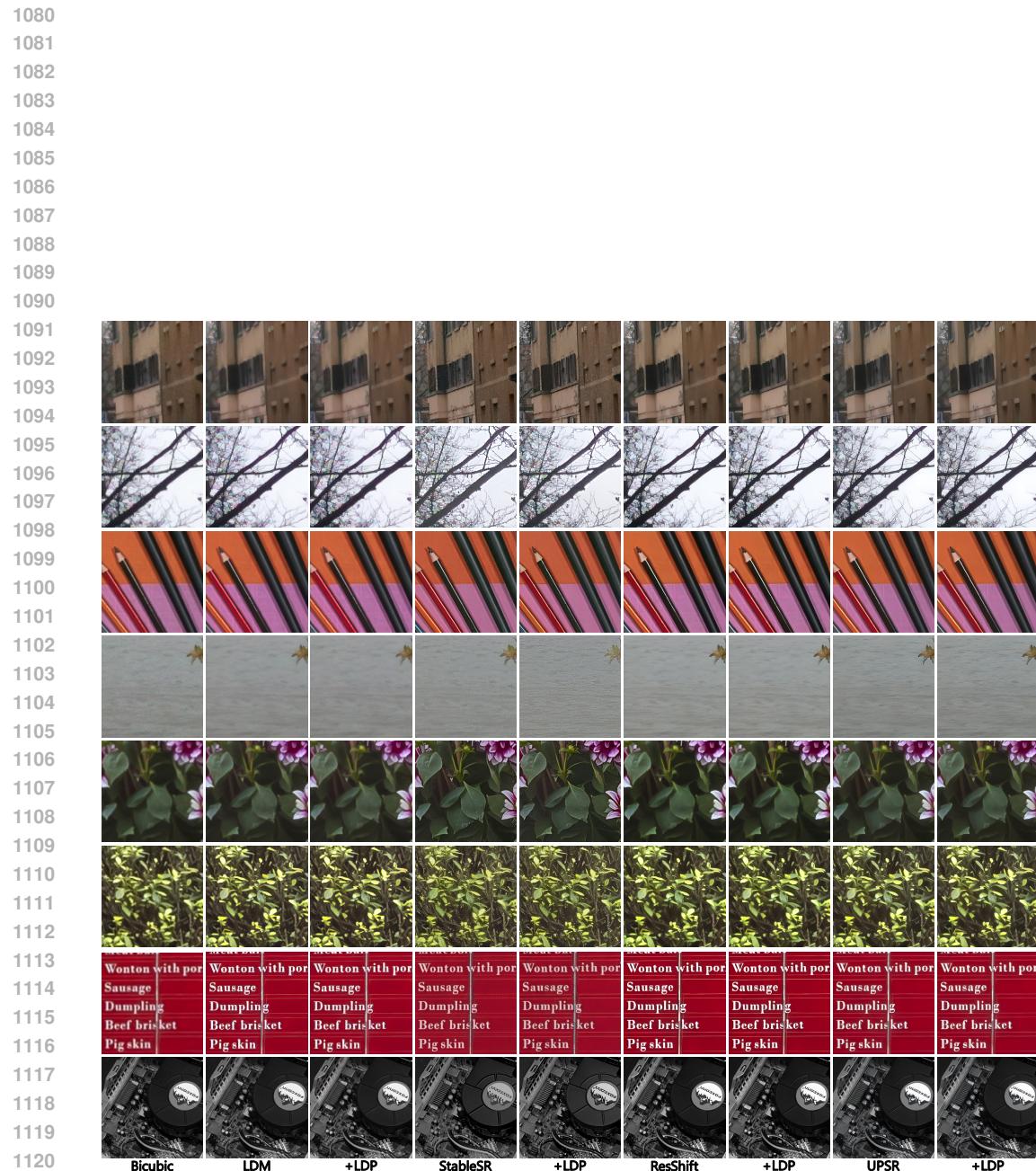


Figure 9: Qualitative results of LDP enhances diffusion models through posterior sampling at  $\times 4$  scale SR. (Zoom in for details)