Optical Tolerance-Compensated Diffusion Model for Image Restoration

Supplementary Material

A. Universality of Tolerance Priors Across Architectures

In the main paper, we compare models trained with and without our tolerance prior to verify its effectiveness within our primary framework. Here, we further conduct an extended ablation to answer a broader question: does the proposed tolerance prior consistently help across different diffusion-based restoration architectures?

To validate this, we apply the same tolerance-prior injection mechanism to three representative diffusion-based restoration models with distinct designs: StableSR (time-aware encoder without ControlNet), DiffBIR (ControlNet-based), and SeeSR (multi-branch ControlNet). For all models, the tolerance prior is encoded into N_{tol} tokens and incorporated via layer-wise cross-attention modulation in their UNet backbones, following the procedure described in Algorithm 1. The PSF is flattened, encoded into a 768-D latent, expanded into N_{tol} tokens, and injected into UNet attention as key/value embeddings.

As summarized in Table 5, integrating the tolerance prior consistently improves restoration quality across all architectures despite their structural differences. This indicates that our approach is not tied to a particular network design, but rather provides a general and architecture-agnostic way to leverage physical degradation information.

Algorithm 1 Kernel-Prompt Injection in UNet Attention Blocks

Require: Input feature map x, PSF kernel k
Ensure: Updated feature map x'
  $k_{\text{flat}} \leftarrow \text{flatten}(k)$
  $z_{768} \leftarrow \text{TolPriorNet}(k_{\text{flat}})$ ▷ project to 1×768
  $\mathbf{t}_k \leftarrow \text{Expand}(z_{768}, N_{\text{tol}})$ ▷ $N_{\text{tol}} \times 768$ tokens
  for each attention block in UNet **do**
    $\mathbf{Q} \leftarrow \text{Proj}_q(x)$
    $\mathbf{K}, \mathbf{V} \leftarrow \text{Proj}_{kv}(\mathbf{t}_k)$
    $\mathbf{a} \leftarrow \text{MultiHeadAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$
    $x \leftarrow x + \text{Proj}_o(\mathbf{a})$ ▷ residual update
  end for
  return x

B. Degradation Consistency Loss Analysis

In this section, we provide a rigorous mathematical analysis of the **Degradation Consistency(DC) Loss** and explain why it remains effective for non-ideal, tolerance-induced PSF kernels.

Definition. Let $\hat{x}_{\text{mid}}$ denotes the the output of the intermediate step, and $D(\cdot; k)$ is re-degraded by the same degradation kernel K from OTPS in Sec. 3.1:

$$\hat{x}_{\text{LQ}} = \text{D}(\hat{x}_{\text{mid}}, K), \tag{10}$$

and compared with the original low-quality observation x_{LQ}:

$$\mathcal{L}_{\text{DC}} = \|\hat{x}_{\text{LQ}} - x_{\text{LQ}}\|_1. \tag{11}$$

Gradient Analysis. The gradient of DC Loss with respect to $\hat{x}$ is:

$$\frac{\partial \mathcal{L}_{\text{DCL}}}{\partial \hat{x}_{\text{mid}}} = k^\top * \text{sign}(\text{D}(\hat{x}_{\text{mid}}, K) - x_{\text{LQ}}), \tag{12}$$

where $k^\top$ denotes the 180-degree rotated kernel. Crucially, the gradient of $\mathcal{L}_{\text{DC}}$ with respect to the intermediate reconstruction $\hat{x}_{\text{mid}}$ is well-defined for any known PSF kernel K, including Gaussian, asymmetric, or spatially varying kernels, since convolution is a linear operation and the L1 loss is subdifferentiable everywhere.

Linearity and Solution Space. If k is (approximately) full-rank, the linear system

$$k * \hat{x}_{\text{mid}} = x_{\text{LQ}} \tag{13}$$

admits a unique solution $\hat{x}_{\text{mid}} \approx k^{-1} * x_{\text{LQ}}$, and DC Loss constrains the restored image to lie in this physically plausible solution space, reducing ambiguity.

For spatially varying PSFs, denoted $k_{i,j}$ at location (i, j):

$$D(\hat{x}, k)_{i,j} = \sum_{(u,v)} k_{i,j}(u, v) \cdot \hat{x}_{i+u,j+v}, \tag{14}$$

the gradient similarly applies locally:

$$\frac{\partial \mathcal{L}_{\text{DCL}}}{\partial \hat{x}_{m,n}} = \sum_{i,j} k_{i,j}^\top(m, n) \cdot \text{sign}\big(D(\hat{x}_{\text{mid}}, K)_{i,j} - x_{\text{LQ},i,j}\big), \tag{15}$$

which enforces local consistency according to the PSF at each spatial position.

If the degradation operator is nonlinear, e.g., $D(\hat{x}; k) = f(k * \hat{x})$ including sensor noise, quantization, or optical aberrations, the gradient can still be computed:

$$\frac{\partial \mathcal{L}_{\text{DC}}}{\partial \hat{x}_{\text{mid}}} = \frac{\partial D(\hat{x}_{\text{mid}}, K)}{\partial \hat{x}_{\text{mid}}}^\top \cdot \text{sign}(D(\hat{x}_{\text{mid}}, K) - x_{\text{LQ}}), \tag{16}$$

allowing DC Loss to guide the restoration process toward physically consistent solutions.

Table 5. Ablation study: effect of injecting the known tolerance prior into three baselines on two datasets (DIV2K-val and RealSR). We also report parameter increment and inference overhead to ensure fairness. The best results are stressed in RED.

Dataset	Method	StableSR					DiffBIR					SEESR				
		PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	FID ↓
DIV2K-val	Baseline	19.85	0.5294	0.2701	0.1787	65.81	21.32	0.5127	0.3392	0.2047	47.24	21.07	0.5551	0.3576	0.2164	62.42
	Baseline + Tol.	20.70	0.5524	0.2715	0.1567	49.52	22.58	0.6067	0.1654	0.1132	36.51	22.09	0.5798	0.2168	0.1660	45.19
RealSR	Baseline	21.21	0.5598	0.2930	0.2147	74.27	22.83	0.5887	0.3176	0.2049	65.24	22.23	0.5864	0.3716	0.2390	74.47
	Baseline + Tol.	21.48	0.5792	0.2917	0.1998	66.28	23.92	0.6241	0.2441	0.1877	50.50	23.11	0.5908	0.2136	0.2043	63.92

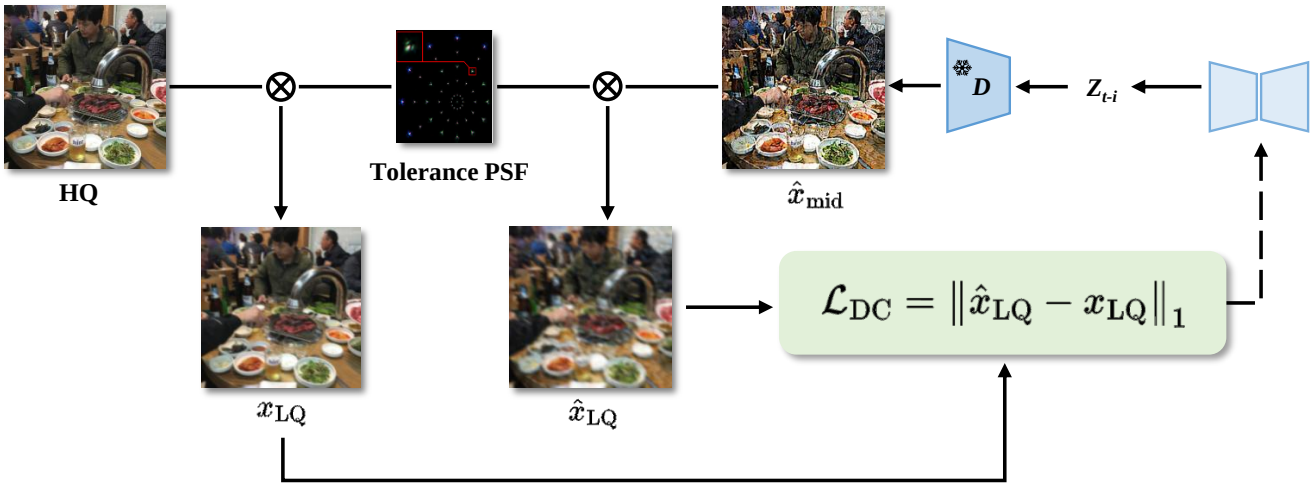

Figure 5. Architecture of the Degradation Consistency Loss (DCL).

Exploration: Adaptive Weighting for Degradation Consistency Loss. In addition to the main training strategy, we also experimented with an adaptive weighting scheme for DC Loss. The idea is that early in training the model's noise predictions are inaccurate, making the intermediate reconstruction $\hat{x}_{\text{mid}}^{(t)}$ unreliable and a strong DC Loss penalty potentially harmful. As training improves the denoising accuracy, a stronger DC Loss constraint becomes more meaningful. To capture this training-progress-dependent behavior, we used the model's denoising error as a proxy for its current capability. Specifically, we compute an exponential moving average (EMA) of the noise-prediction loss:

$$\bar{e}_t = \beta\, \bar{e}_{t-1} + (1-\beta)\, \|\hat{\epsilon}_\theta(x_t, t) - \epsilon\|_2^2, \tag{17}$$

with $\beta \in [0.95, 0.995]$. A normalized confidence score is then derived:

$$q_t = 1 - \frac{\bar{e}_t - e_{\min}}{(e_{\max} - e_{\min}) + \varepsilon}, \tag{18}$$

where $e_{\min}$ and $e_{\max}$ track the minimum and maximum historical EMA errors.

The total training objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Tol}} + \lambda_{\text{DC}} q_t \mathcal{L}_{\text{DC}}. \tag{19}$$

Although intuitive, the adaptive weighting did not bring consistent gains in our experiments. The noise-prediction error fluctuated considerably during training, making the adaptive weights unstable and sometimes suppressing DC Loss when it should be emphasized. Therefore, we did not use this strategy in the final model and instead adopted a fixed, manually tuned DC Loss weight for more stable training.

C. More qualitative comparison results.

We provide more visual comparisons in Fig. 6, Fig. 7, respectively.

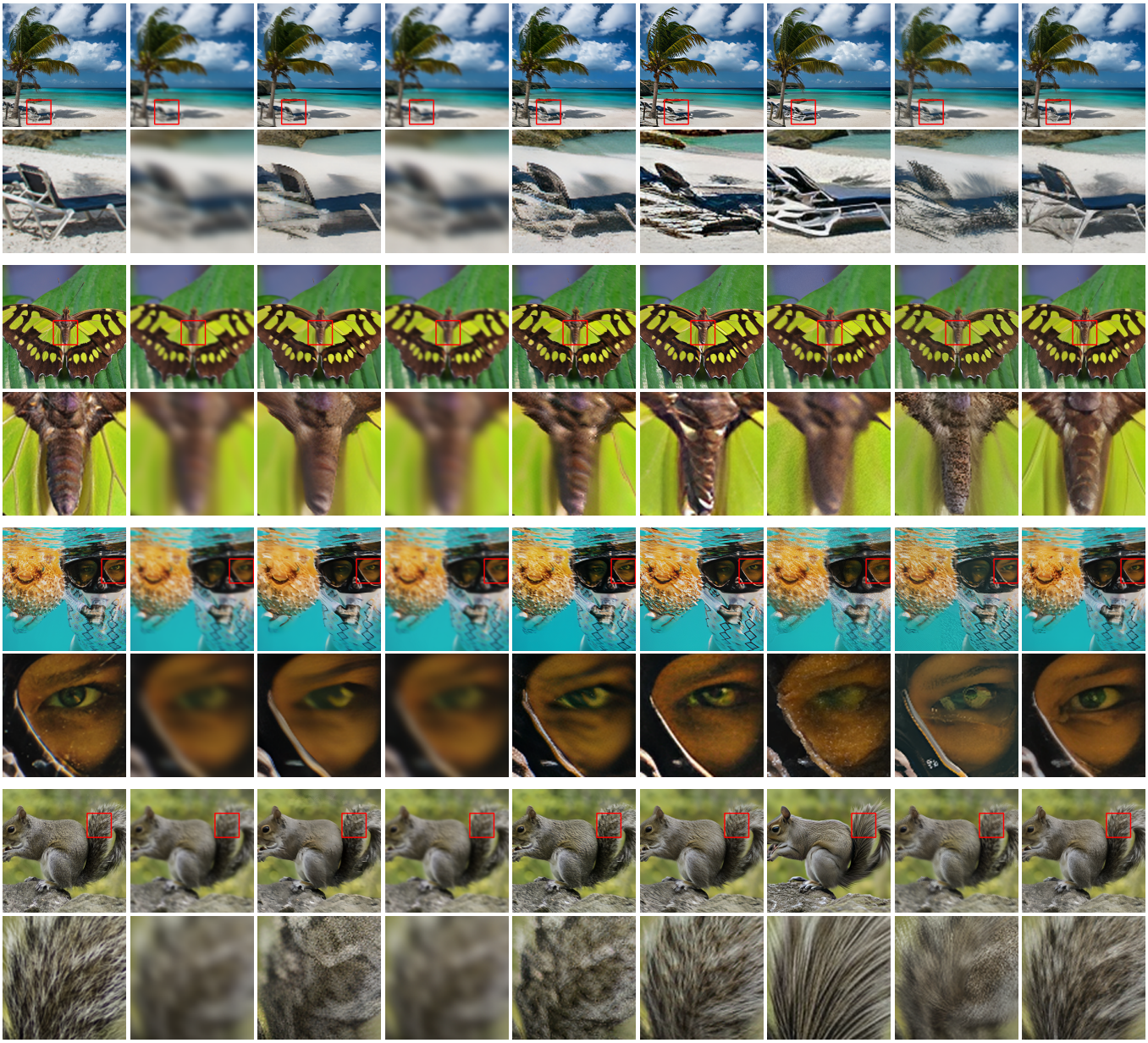

Figure 6. More qualitative comparison results of test images. For each we also show an enlarged patch in which finer details are more apparent.

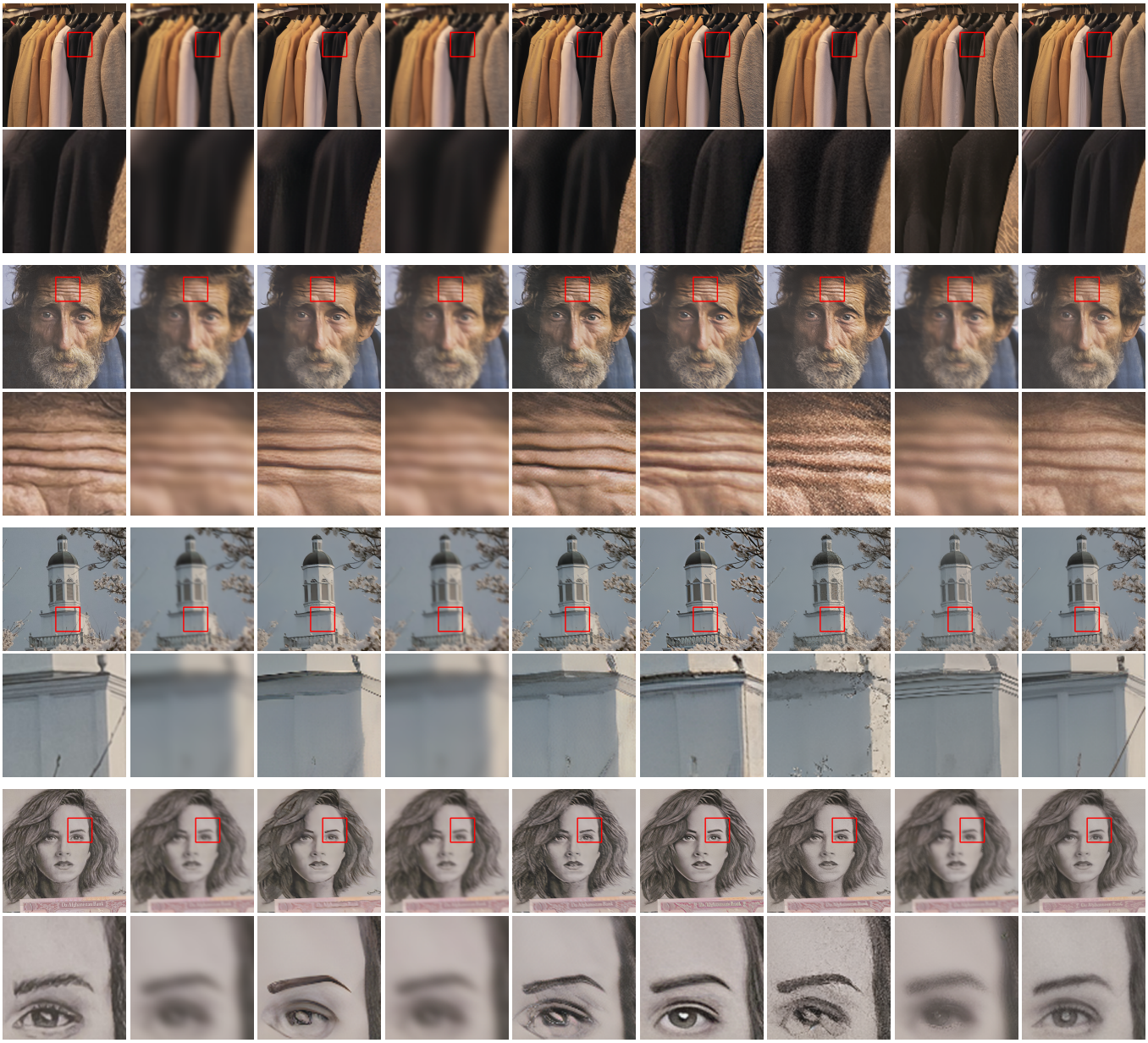

Figure 7. More qualitative comparison results of test images. For each we also show an enlarged patch in which finer details are more apparent.