Supplementary Material for *Self-Diffusion Driven Blind Imaging*

A. Theoretical Motivation

The SDI framework operates as an unrolled iterative optimization process that treats the self-deblurring task as a sequence of refined Bayesian updates, where each step's output serves as the informed prior for the subsequent stage. Unlike the static Deep Image Prior (DIP), SDI leverages a closed-loop feedback mechanism and stochastic noise injection to progressively narrow the solution space, theoretically allowing the network to converge onto high-frequency image details that are typically lost in single-pass architectures. By integrating the previous step's estimate back into the current optimization cycle, SDI aims to balance the robustness of structural priors with a dynamic data consistency constraint, effectively navigating the complex landscape of non-blind deconvolution through a trajectory of guided refinements.

B. Extending Analysis

B.1. More Layers *vs*. Noise Schedule?

We perform a full ablation study in Figure 1 to address the concern about whether the performance gain is solely due to increased depth of networks. If the performance gain were solely attributed to increased depth, we would expect comparable results across different scheduling strategies using the same architecture. However, as shown in the PSNR/SSIM curves and visual results, the DDPM schedule (steeper decay) significantly outperforms all EDM schedules ($\rho = 3$ to 7) despite using identical network layers. The more gradual EDM schedules lead to over-smoothed reconstructions, demonstrating that the specific dynamics of noise reduction, rather than simple model capacity, are the primary driver of optimization success. This evidence confirms that increasing the number of layers in a "standard mode" without the appropriate scheduling fails to achieve high-fidelity restoration, as the schedule is what dictates the balance between structural recovery and detail refinement.

B.2. Analysis on λ_k:

Table 1 demonstrates that our model remains robust across a wide range of regularization weights (λ_k), with stable PSNR/SSIM metrics indicating that the optimization is not overly sensitive to sparsity constraints. This robustness is primarily attributed to the softmax output layer of the kernel network, which inherently constrains the PSF to a valid probability simplex, allowing the model to adaptively recover naturally dense optical aberrations without being hindered by external regularization weights.

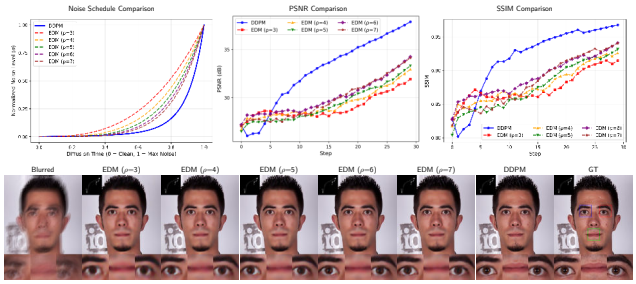

Figure 1. Noise schedule comparison between DDPM and EDM (Multiple ρ values). Up right: Noise schedule curves. Up middle/right: PSNR/SSIM comparison. Bottom: DeblurSDI results at different edm ρ values.

Table 1. Ablation study on λ_k in Equation (12).

λ_k	1e-4	1e-3	1e-2	0.1	1	10
PSNR/SSIM	37.83/0.9843	37.72/0.9856	38.18/0.9865	37.33/0.9845	38.04/0.9861	37.99/0.9859

B.3. Computational Complexity Analysis

Table 2 provides a quantitative comparison of average runtime per iteration, showing that while DeblurSDI offers competitive speed on standard benchmarks, the back-propagation in iterations result in longer runtimes for high-resolution images compared to feed-forward computation. Future implementations could employ mixed-precision computation or early-stopping criteria to minimize redundant iterations.

Table 2. Comparison of computational efficiency (s/iter)

Dataset (Size)	Phase-Only	FFT-ReLU	SelfDeblur	FastDiffusionEM	**DeblurSDI**
Levin (398×398)	2.7947	0.00278	0.01402	0.03857	**0.0406**
Cho (463×622)	1.2415	0.00224	0.05568	0.03805	**0.2149**
Kohler (800×800)	2.9619	0.00304	0.12479	0.04026	**0.4006**
FFHQ (256×256)	0.2708	0.00260	0.01392	0.03826	**0.0386**

C. Real-world Experiments

As demonstrated in Figure 2, the proposed DeblurSDI significantly outperforms other baseline methods in real-world microscopy settings. It achieves a clean reconstruction.

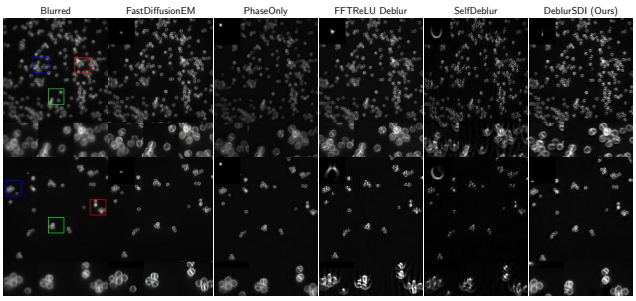

Figure 2. Real-world microscope image deblurring results.

D. Explanations

Metrics Discrepancy: Our evaluation strictly follows the authors' original code and boundary handling protocols; however, the significant metric gaps stem from the variety of psf dataset, where it is both impractical and counterproductive to manually tune hyperparameters for every image-kernel combination. We found that some methods used simple psf to achieve good metrics, e.g., FastDiffusionEM. As illustrated by Figure 6 in main paper, these performance discrepancies highlight the sensitivity of existing methods to varying kernel sizes and emphasize the need for more resilient deblurring solutions.

Input discrepancy of G_ϕ standard mode and diffusion mode: The standard mode of G_ϕ uses a fixed latent vector z sampled from a normal distribution, while the diffusion mode treats the latent vector z_t as a dynamic input that evolves through the self-diffusion process.

"ReLU6" in G_ϕ and Non-local Blocks in D_θ: $ReLU6 = \min(\max(x, 0), 6)$ [1], while non-local blocks are used to capture long-range dependencies referring from [2].

References

[1] PyTorch Developers. *ReLU6 — PyTorch 2.1 documentation*, 2023. Accessed: 2024-05-20. 2

[2] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2