Improving Diffusion Posterior Samplers with Lagged Temporal Corrections for Image Restoration

Davide Evangelista
Dept. of Computer Science and Engineering
University of Bologna
davide.evangelista5@unibo.it

Elena Morotti
Dept. of Political and Social Sciences
University of Bologna
elena.morotti4@unibo.it

Francesco Pivi
Dept. of Computer Science and Engineering
University of Bologna
francesco.pivi2@unibo.it

Maurizio Gabbrielli
Dept. of Computer Science and Engineering
University of Bologna
maurizio.gabbrielli@unibo.it

Abstract

Diffusion-based posterior sampling (PS) is a leading framework for imaging inverse problems, combining learned priors with measurement constraints. Yet, its standard formulations rely on instantaneous data-consistent estimates, which induce temporal variability in the reverse dynamics. We reinterpret PS from a dynamical perspective, showing that the standard PS update corresponds to a first-order discretization of the diffusion dynamics plus a residual correction capturing the mismatch between the denoised prediction and the data-consistent estimate. A second-order discretization, however, naturally introduces a temporal correction based on the variation of consecutive estimates.
Building on this, we propose LAMP, a second-order-inspired lagged posterior-sampling scheme that combines a multistep temporal correction with the residual correction characterizing a PS technique. LAMP thus inherits a lagged temporal correction, and it can be implemented as a modular plug-in over the PS backbone. We show that LAMP preserves the structure of a posterior sampler, and we perform a one-step risk analysis to characterize when LAMP improves the reverse transition via a bias-variance trade-off. Experiments across multiple imaging tasks show consistent improvement, with the most perceptual gains obtained for DDRM, without increasing the number of denoising evaluations.

1 Introduction

Inverse problems are prevalent in imaging, with applications in restoration, computational photography, and medical reconstruction. The objective is to recover an unknown signal $\boldsymbol{x}_0$ from measurements $\boldsymbol{y} = \boldsymbol{K}\boldsymbol{x}_0 + \boldsymbol{e}$, where $\boldsymbol{K}$ is a possibly ill-conditioned forward operator and $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \sigma_{\boldsymbol{y}}^2 \boldsymbol{I})$ represents noise corruption. Classical approaches rely on variational methods, which cast the task as an optimization problem regularized by hand-crafted priors. While effective in many settings, these models often struggle to capture complex image statistics, especially in highly ill-posed regimes.

Deep learning has emerged as a powerful alternative, enabling significant improvements in reconstruction quality by learning data-driven priors from large datasets. However, learning-based reconstruction methods remain susceptible to instabilities and hallucinated features, particularly under measurement perturbations or distribution shifts [4]. Among these approaches, diffusion-based generative models have demonstrated remarkable performance in modeling complex high-dimensional data distributions [5, 13]. Building on this success, recent works have proposed diffusion models as powerful priors for

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

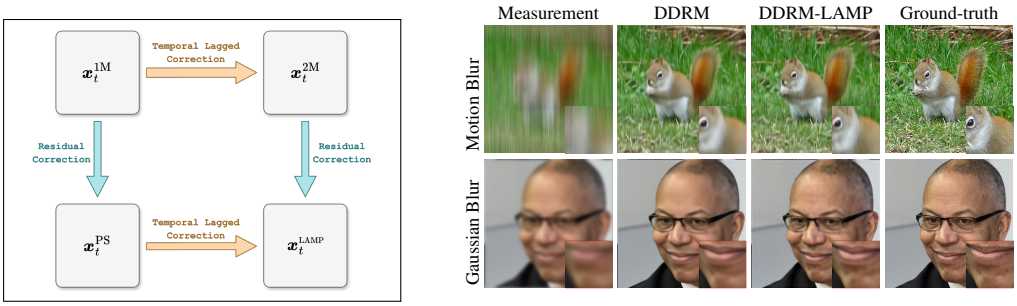

Figure 1: Overview of the LAMP scheme. Left: relation among the updates of the considered diffusive samplers for a fixed time. Right: examples showing LAMP onto the DDRM method.

solving inverse problems, achieving state-of-the-art results across a range of imaging tasks. These approaches reinterpret image processing as a *posterior sampling* (PS) task: given a forward model and noisy measurements, the goal is to sample from the posterior distribution of the unknown image conditioned on the observations. In this framework, a pre-trained diffusion model acts as a powerful prior, while data consistency is enforced during the sampling process. More precisely, let $\boldsymbol{x}_t^{\mathrm{PS}}$ denote the current iterate at diffusion time t. Each standard posterior sampling update combines a denoising step provided by the diffusion model with a correction enforcing consistency with the observed data. This leads to updates of the general form:

$$\boldsymbol{x}_{t-\Delta t}^{\mathrm{PS}} = \alpha_{t-\Delta t}\boldsymbol{D}_{t-\Delta t} + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t^{\mathrm{PS}}, t), \tag{1}$$

where $\{\alpha_t\}_{t=0}^{T-1}$ and $\{\sigma_t\}_{t=0}^{T-1}$ denote the time-dependent signal and noise coefficients satisfying $\alpha_t^2 + \sigma_t^2 = 1$ for any t, $\Delta t > 0$ is the time discretization term, $\boldsymbol{\epsilon}_\Theta(\cdot,\cdot)$ is the learned noise predictor, and $\boldsymbol{D}_t$ is a data-consistent estimate.

Variants of this approach underlie several recent methods, which differ in how the data-consistency step is implemented and integrated within the diffusion dynamics. For instance, $\boldsymbol{D}_t$ is obtained via a proximal step in the DiffPIR method [18], via a projection onto the measurement constraints in the DDNM method [15], or in closed form in a transformed domain when the forward operator admits a suitable decomposition in the DDRM method [6]. More details are provided in Appendix A. In all cases, existing PS methods rely on instantaneous, step-wise data-consistent estimates $\boldsymbol{D}_t$ that are recomputed and injected at each reverse iteration. This design introduces instability in the correction term, as small changes in the iterate $\boldsymbol{x}_t$ can lead to inconsistent updates due to denoising error, discretization, and model mismatch. As a result, the reverse dynamics are effectively driven by a time-varying and potentially noisy forcing signal.

From a dynamical perspective, this leads to two key limitations. First, the resulting trajectory is not temporally smooth, deviating from the behavior expected from a well-defined continuous-time process. Second, the method lacks temporal memory, as each correction $\boldsymbol{D}_t$ is computed independently despite the strong correlation between consecutive estimates. These effects are largely overlooked in current approaches, yet they play a crucial role in the stability and consistency of diffusion-based inverse solvers.

This lack of temporal coherence also reveals a fundamental mismatch with recent advances in diffusion PS for unconditional generation based on probability-flow ODEs [13]. In this context, in fact, high-order solvers such as DPM-Solver [9] achieve remarkable acceleration by assuming that the reverse process follows a smooth and well-defined continuous-time dynamics. However, in the presence of measurement constraints, this assumption no longer holds. The data-consistency corrections introduced by PS act as a time-dependent forcing term that perturbs the reverse dynamics at every step. As a result, the trajectory deviates from that of a standard probability-flow ODE, both in terms of smoothness and consistency, limiting the direct applicability of classical high-order solvers to inverse problems.

Motivated by this perspective, our contributions are as follows:

- We identify a structural mismatch between ODE-based diffusion solvers and diffusive PS methods for inverse problems, showing that the latter are implicitly driven by a residual-dependent forcing term that is not captured by standard probability-flow ODE formulations.

- We reinterpret posterior samplers as discrete dynamical systems and show that standard posterior updates correspond to a first-order discretization of a diffusion dynamics corrected by a residual-dependent force. Building on this perspective, we propose **LAMP** (LAgged Multistep Posterior) as a second-order-inspired scheme, explicitly accounting for both the temporal evolution and a residual-dependent term, as sketched in the first panel of Figure 1.
- From a theoretical perspective, we show that LAMP preserves the standard posterior sampling structure in Eq. (1), as its update can be expressed in the same form with a modified data-consistent estimate that incorporates temporal information. This establishes LAMP as a modular extension of PS methods, rather than a separate reconstruction scheme.
- We show that LAMP can be instantiated on posterior sampling (PS) backbones whose updates admit the form in Eq. (1), by replacing their measurement-aware estimate $\boldsymbol{D}_t$ with a temporally filtered estimate. The resulting update preserves this posterior-sampling structure without modifying the denoiser or the backbone correction operator.
- We provide a one-step risk analysis showing that, under a mild condition, the error induced by LAMP is smaller than that of the base PS method on which it is built. We further show that this condition is typically satisfied in the later stages of the diffusion process, which motivates using the base sampler for the first two reverse updates and applying LAMP thereafter.

2 Related Works

Diffusion models for image generation. Diffusion probabilistic models have become a standard paradigm for generative modeling, starting from DDPM [5] and score-based formulations [13, 10]. These models admit both stochastic (SDE) and deterministic (ODE) interpretations, the latter leading to probability-flow dynamics that can be discretized for sampling [13, 11]. This connection has motivated a large body of work on accelerating sampling through improved numerical solvers, including DDIM [11], PNDM [8], DPM-Solver [9], and subsequent higher-order and accelerated methods [9, 17]. These approaches are derived under the assumption that the reverse process follows an underlying ODE, and therefore aim to reduce the discretization error of that continuous dynamics. In contrast, our work highlights that such ODE-based interpretations do not fully capture the behavior of posterior sampling methods for inverse problems, where additional residual-dependent terms arise.

Diffusion models for inverse problems. Recently, diffusion models have been adapted to inverse problems by integrating measurement constraints directly into the reverse dynamics. Early approaches combine pretrained diffusion models with projection or conditioning strategies [7, 16]. A large class of methods can be interpreted as posterior sampling (PS) procedures, where data consistency is enforced at each step through an explicit update. MCG introduces a manifold-constrained gradient to enforce measurement consistency while maintaining proximity to the data manifold [2]. DDRM formulates posterior sampling for linear inverse problems via a variational approximation, achieving strong performance and efficiency [6]. DPS guides the reverse process using an approximate measurement-likelihood gradient, extending diffusion posterior sampling to noisy and nonlinear inverse problems [1]. ΠGDM approximates the conditional score using the measurement model and a pseudoinverse-guided correction [12]. DDNM instead exploits a range/null-space decomposition to enforce measurement consistency in linear settings, enabling zero-shot restoration without retraining [15]. Related approaches also include plug-and-play diffusion priors, where a diffusion denoiser is combined with iterative data-consistency updates, as in DiffPIR [18]. While these methods differ in how the posterior update is constructed, they all share a common iterative structure, which we reinterpret in this work as a discretization of an underlying diffusion dynamics with an additional residual-dependent forcing term.

Positioning of our method. Rather than introducing a new posterior update, prior, or conditioning strategy, our contribution focuses on the temporal structure of existing PS methods. We identify a previously underexplored source of instability, namely the temporal variability of measurement-aware estimates across iterations, and show that standard PS updates correspond to a first-order discretization of a forced diffusion dynamics. This perspective naturally leads to a multistep correction scheme for PS backbones admitting the update structure in Eq. (1), without modifying the underlying model or training procedure.

3 Methodology

In this section, we derive LAMP by combining a temporal correction inspired by a second-order multistep approximation with the residual term typical of posterior sampling. More details, including proofs of the theoretical results, are provided in Appendix A and B. We first recall the minimal DDIM and probability-flow identities needed to fix notation.

3.1 Background

Consider a fixed timestep t. Using the variance-preserving parameterization $\alpha_t^2 + \sigma_t^2 = 1$, and the standard marginal noising relation $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}_t$, with $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, the clean-image Tweedie estimation [14, 5] given the noise predictor $\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t)$ reads

$$\hat{\boldsymbol{x}}_{0|t} := \mathbb{E}\left[\boldsymbol{x}_0 | \boldsymbol{x}_t\right] = \frac{\boldsymbol{x}_t - \sigma_t \boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t)}{\alpha_t}. \tag{2}$$

DDIM [11] then defines the deterministic updates at time t as

$$\boldsymbol{x}_{t-\Delta t} = \alpha_{t-\Delta t} \hat{\boldsymbol{x}}_{0|t} + \sigma_{t-\Delta t} \boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t). \tag{3}$$

A complementary line of work interprets the generation process as the discretization of a continuous-time probability-flow ordinary differential equation (PF-ODE) [13]. In the variance-preserving setting, this ODE can be written directly in terms of the signal and noise rates α_t and σ_t as

$$\frac{d\boldsymbol{x}_t}{dt} = \frac{d \log \sigma_t}{dt} \boldsymbol{x}_t + \frac{d}{dt}\left(\frac{\alpha_t}{\sigma_t}\right) \sigma_t^2 \hat{\boldsymbol{x}}_{0|t}, \tag{4}$$

where $\hat{\boldsymbol{x}}_{0|t}$ is defined in (2). It is often convenient to reparameterize time using the log-SNR variable $\lambda_t := \log(\alpha_t/\sigma_t)$, which provides a more uniform parametrization of the diffusion trajectory. Using this change of variables, Eq. (4) can be rewritten in integrating-factor form as

$$\frac{d}{d\lambda}\left(\frac{\boldsymbol{x}(\lambda)}{\sigma(\lambda)}\right) = \frac{\alpha(\lambda)}{\sigma(\lambda)} \hat{\boldsymbol{x}}_0(\boldsymbol{x}(\lambda), \lambda), \tag{5}$$

which highlights a linear structure in the transformed variable $\boldsymbol{x}(\lambda)/\sigma(\lambda)$. Integrating from λ_t to $\lambda_{t-\Delta t}$ and setting $h = \lambda_{t-\Delta t} - \lambda_t > 0$, yields

$$\boldsymbol{x}_{t-\Delta t} = \frac{\sigma_{t-\Delta t}}{\sigma_t} \boldsymbol{x}_t + \sigma_{t-\Delta t} \int_{\lambda_t}^{\lambda_{t-\Delta t}} \frac{\alpha(\lambda)}{\sigma(\lambda)} \hat{\boldsymbol{x}}_0(\boldsymbol{x}(\lambda), \lambda)\, d\lambda. \tag{6}$$

Approximating $\hat{\boldsymbol{x}}_0(\boldsymbol{x}(\lambda), \lambda)$ by the constant value $\hat{\boldsymbol{x}}_{0|t}$ over the interval $[\lambda_t, \lambda_{t-\Delta t}]$, (6) becomes:

$$\boldsymbol{x}_{t-\Delta t} = \frac{\sigma_{t-\Delta t}}{\sigma_t} \boldsymbol{x}_t + \alpha_{t-\Delta t} A_0(h) \hat{\boldsymbol{x}}_{0|t}, \quad \text{with } A_0(h) = 1 - e^{-h} \approx h. \tag{7}$$

As proved in Appendix B.2, this expression is algebraically equivalent to (3), implying that standard DDIM step can be interpreted as the first-order exponential discretization of the probability-flow ODE. The idea of higher-order solvers such as DPM-Solver++ [9], is to achieve improved accuracy by approximating the integral in (6) using multistep polynomial expansions of the model prediction. In particular, employing 2-step approximation of the integrating factor $\hat{\boldsymbol{x}}_0(\boldsymbol{x}(\lambda), \lambda)$ leads to:

$$\boldsymbol{x}_{t-\Delta t} = \frac{\sigma_{t-\Delta t}}{\sigma_t} \boldsymbol{x}_t + \alpha_{t-\Delta t} A_0(h) \hat{\boldsymbol{x}}_{0|t} + \alpha_{t-\Delta t} A_1(h) \gamma \frac{\hat{\boldsymbol{x}}_{0|t+\Delta t} - \hat{\boldsymbol{x}}_{0|t}}{h}, \tag{8}$$

where $A_1(h)$ is the standard second-order exponential coefficient, given by $A_1(h) = 1 - \frac{1-e^{-h}}{h} \approx \frac{h^2}{2}$.

3.2 LAMP: Lagged Multistep Posterior Sampler

When diffusive PS models are employed to solve inverse problems as $\boldsymbol{y} = \boldsymbol{K}\boldsymbol{x}_0 + \boldsymbol{e}$, at each reverse step, the raw denoised estimate $\hat{\boldsymbol{x}}_{0|t}$ in (3) is replaced with the measurement-aware estimate $\boldsymbol{D}_t$, and the reverse step takes the common form reported in (1). Here $\boldsymbol{D}_t = \mathcal{C}_t(\hat{\boldsymbol{x}}_{0|t}, \boldsymbol{y}, \boldsymbol{K})$ where $\mathcal{C}_t$ represents the image operator induced by the chosen posterior sampler, as already discussed in Section 1.

A continuous analogue of (1) can be obtained by replacing $\hat{\boldsymbol{x}}_0(\boldsymbol{x}(\lambda), \lambda)$ with a measurement-aware target $\boldsymbol{D}_\lambda := \boldsymbol{D}_{\lambda^{-1}(\lambda_t)}$ in (5):

$$\frac{d}{d\lambda}\left(\frac{\boldsymbol{x}(\lambda)}{\sigma(\lambda)}\right) = \frac{\alpha(\lambda)}{\sigma(\lambda)}\boldsymbol{D}_\lambda, \tag{9}$$

which, approximating $\boldsymbol{D}_\lambda$ by the constant value $\boldsymbol{D}_t$ on $[\lambda_t, \lambda_{t-\Delta t}]$, yield to the corresponding first-order exponential step:

$$\boldsymbol{x}^{\mathrm{1M}}_{t-\Delta t} = \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \alpha_{t-\Delta t}A_0(h)\boldsymbol{D}_t. \tag{10}$$

However, unlike in the unconditional setting where DDIM coincides with a first-order exponential discretization, the posterior update rule in (1) is not equal to (10). Indeed, substituting $\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_t - \alpha_t \hat{\boldsymbol{x}}_{0|t}}{\sigma_t}$ into (1) gives the exact decomposition

$$\boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t} = \underbrace{\frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \alpha_{t-\Delta t}A_0(h)\boldsymbol{D}_t}_{x^{\mathrm{1M}}_{t-\Delta t}} + \alpha_{t-\Delta t}e^{-h}\left(\boldsymbol{D}_t - \hat{\boldsymbol{x}}_{0|t}\right). \tag{11}$$

Here, the first two terms correspond to the first-order exponential step associated with (9), i.e. to $\boldsymbol{x}^{\mathrm{1M}}_{t-\Delta t}$, while the last term is a *residual* correction term, as it explicitly depends on the mismatch $\boldsymbol{D}_t - \hat{\boldsymbol{x}}_{0|t}$ between the data-consistent estimate and the diffusion prior. This term therefore captures the discrepancy between the two sources of information and acts as a corrective forcing that steers the reverse dynamics toward data consistency. In particular, it vanishes only when $\boldsymbol{D}_t = \hat{\boldsymbol{x}}_{0|t}$, i.e., when the prior prediction is already consistent with the measurements. As we will further illustrate in Section 4, this residual correction plays a beneficial role, leading to improved performance of $\boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t}$ over its baseline first-order step $\boldsymbol{x}^{\mathrm{1M}}_{t-\Delta t}$.

In parallel, a second-order approximation of (9) involves the temporal variation of the data-consistent estimates across two consecutive reverse steps. In particular, introducing $h_{\mathrm{prev}} := \lambda_t - \lambda_{t+\Delta t}$, a standard 2-step DPM-Solver-like discretization includes a *temporal* term proportional to the discrete derivative $(\boldsymbol{D}_t - \boldsymbol{D}_{t+\Delta t})/h_{\mathrm{prev}}$, yielding

$$\boldsymbol{x}^{\mathrm{2M}}_{t-\Delta t} = \boldsymbol{x}^{\mathrm{1M}}_{t-\Delta t} + \alpha_{t-\Delta t}A_1(h)\gamma\frac{\boldsymbol{D}_t - \boldsymbol{D}_{t+\Delta t}}{h_{\mathrm{prev}}}, \tag{12}$$

which corresponds to the posterior sampling in (8).

Motivated by the favorable role of the residual correction term linking $\boldsymbol{x}^{\mathrm{1M}}_{t-\Delta t}$ and $\boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t}$, we introduce **LAMP** (LAgged Multistep Posterior), a second-order-inspired multistep posterior sampler that combines the temporal correction motivated by the diffusion discretization with the residual correction term. Building on (11), the LAMP update is defined as

$$\boldsymbol{x}^{\mathrm{LAMP}}_{t-\Delta t} := \boldsymbol{x}^{\mathrm{2M}}_{t-\Delta t} + \alpha_{t-\Delta t}e^{-h}\left(\boldsymbol{D}_t - \hat{\boldsymbol{x}}_{0|t}\right), \tag{13}$$

i.e., as a two-step posterior scheme motivated by the multistep discretization and augmented by the residual correction term. This idea is depicted in Figure 1.

By combining (11) and (12) into (13), we obtain the following update:

$$\boldsymbol{x}^{\mathrm{LAMP}}_{t-\Delta t} = \boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t} + \alpha_{t-\Delta t}A_1(h)\gamma\frac{\boldsymbol{D}_t - \boldsymbol{D}_{t+\Delta t}}{h_{\mathrm{prev}}}, \tag{14}$$

which highlights the presence of the temporal correction term modulated by the coefficient γ. Precisely, the term $(\boldsymbol{D}_t - \boldsymbol{D}_{t+\Delta t})/h_{\mathrm{prev}}$ captures the variation of the data-consistent estimates across consecutive reverse steps, and can be interpreted as a discrete temporal derivative. The sign of γ determines whether this correction extrapolates the sequence of estimates forward or introduces a stabilizing effect. In particular, choosing $\gamma < 0$ yields a *lagged* temporal correction, where the current estimate is slightly pulled toward previous ones rather than extrapolated. This mechanism counteracts the variability induced by the instantaneous posterior correction, leading to smoother and more stable reverse dynamics. This motivates the name LAMP, which emphasizes the role of lagged multistep information in the posterior update.

The following identity, whose proof is given in Appendix B.4, shows that the LAMP scheme preserves the posterior sampling structure.

Proposition 1 (Lagged posterior form)**.** *Let* $\beta_t := -\gamma \frac{A_1(h)}{h_{\text{prev}}} > 0$*. Then, the LAMP update* (14) *can be written as*

$$\boldsymbol{x}_{t-\Delta t}^{\text{LAMP}} = \alpha_{t-\Delta t}\widetilde{\boldsymbol{D}}_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t), \tag{15}$$

where

$$\widetilde{\boldsymbol{D}}_t = (1-\beta_t)\boldsymbol{D}_t + \beta_t \boldsymbol{D}_{t+\Delta t}. \tag{16}$$

Proposition 1 is the key structural property of LAMP: the method is not an external post-processing step and it is not a classical higher-order solver for the unconditional probability-flow ODE. It is again a posterior sampler of the form (1), but with the instantaneous estimate $\boldsymbol{D}_t$ replaced by the temporally filtered estimate $\widetilde{\boldsymbol{D}}_t$. Therefore, LAMP can be applied on top of different correction operators $\mathcal{C}_t$ without changing the denoiser, the forward model, or the reverse schedule characterising state-of-the-art PS models.

We note that, while the filtered estimate in Eq. (16) resembles an exponential moving average (EMA), the two updates use past information differently. In particular, LAMP combines the current estimate $\boldsymbol{D}_t$ with the previous unfiltered estimate $\boldsymbol{D}_{t+\Delta t}$, whereas EMA uses the previously filtered estimate. In addition, the LAMP weight β_t varies with the reverse schedule, while the EMA weight is typically constant.

We now give a local error argument explaining when the lagged estimate improves the reverse transition. First of all, note that although the LAMP trajectory is deterministic after initialization, the corrected estimate $\boldsymbol{D}_t$ is not exact, since it inherits denoising error, discretization error, and model mismatch. In the following, we rely on the assumption that $\boldsymbol{D}_t = \mu_t + \eta_t$, $\boldsymbol{D}_{t+\Delta t} = \mu_{t+\Delta t} + \eta_{t+\Delta t}$, where μ_t and $\mu_{t+\Delta t}$ are local posterior targets and $\eta_t, \eta_{t+\Delta t}$ are zero-mean estimation errors, such that $\mathbb{E}[\eta_t] = \mathbb{E}[\eta_{t+\Delta t}] = 0$, $\text{Var}(\eta_t) = \text{Var}(\eta_{t+\Delta t}) = \Sigma_t$, and $\text{Cov}(\eta_t, \eta_{t+\Delta t}) = \rho_t \Sigma_t$, with $0 \le \rho_t < 1$. In Appendix B.7 we show a more general result that does not rely on this assumption.

Proposition 2 (One-step risk comparison)**.** *Let* $\boldsymbol{x}_{t-\Delta t}^{\mu} := \alpha_{t-\Delta t}\mu_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t)$ *be the ideal one-step state obtained if the corrected estimate were equal to the local target* μ_t*, and define* $r_t := \mu_{t+\Delta t} - \mu_t$*. It holds:*

$$\mathbb{E}\|\boldsymbol{x}_{t-\Delta t}^{\text{PS}} - \boldsymbol{x}_{t-\Delta t}^{\mu}\|^2 = \alpha_{t-\Delta t}^2 \text{tr}(\Sigma_t), \tag{17}$$

whereas

$$\mathbb{E}\|\boldsymbol{x}_{t-\Delta t}^{\text{LAMP}} - \boldsymbol{x}_{t-\Delta t}^{\mu}\|^2 = \alpha_{t-\Delta t}^2 \left[(1 - 2\beta_t(1-\beta_t)(1-\rho_t))\,\text{tr}(\Sigma_t) + \beta_t^2 \|r_t\|^2\right]. \tag{18}$$

Consequently, LAMP improves the one-step posterior transition whenever

$$\beta_t \|r_t\|^2 < 2(1-\beta_t)(1-\rho_t)\text{tr}(\Sigma_t). \tag{19}$$

The condition in (19) captures the trade-off introduced by temporal filtering. The term $(1-\rho_t)\text{tr}(\Sigma_t)$ measures the non-shared part of the local estimation error between adjacent corrected estimates, which is the component reduced by averaging. The term $\|r_t\|^2$ measures the drift of the underlying posterior target, which is the bias introduced by the lag. Thus, LAMP is beneficial when adjacent posterior targets evolve smoothly while the estimation errors are not perfectly correlated. This motivates a fixed warm-up: in our experiments, LAMP is activated from the third reverse update onward and remains active for the rest of the trajectory. The implementation only requires storing the previous corrected estimate and replacing $\boldsymbol{D}_t$ by the convex combination (16), which makes the method modular and computationally negligible.

4 Numerical Experiments

We evaluate LAMP on four posterior samplers: DiffPIR, DDRM, DPS, and ΠGDM. We refer to their LAMP variants as DiffPIR-LAMP, DDRM-LAMP, DPS-LAMP, and ΠGDM-LAMP, respectively. These backbones cover plug-and-play, spectral, and gradient-based approaches to enforcing data consistency, allowing us to test the effect of the same temporal correction across different posterior-sampling formulations. The goal of the experiments is to test whether the proposed method improves existing diffusion posterior samplers without changing the pretrained diffusion model, the forward operator, or the number of denoising evaluations.

4.1 Experimental setup

We compare each of the four LAMP variants directly with its unmodified backbone. DPS, DiffPIR, DDRM, and ΠGDM use 1000, 100, 20, and 100 denoising evaluations, respectively. Each LAMP variant retains the denoising schedule and number of evaluations of its backbone. We also evaluate the 1M and 2M variants of DiffPIR and DDRM without the residual correction term; these results are reported in Appendix D.

We follow the experimental protocols of each compared method, using their test sets and pretrained backbones. We consider three image datasets: FFHQ, ImageNet, and CelebA. For each dataset, we evaluate three linear inverse problems: Gaussian deblurring, motion deblurring, and super-resolution with scale factor $\times 4$. The measurements are generated according to $\boldsymbol{y} = \boldsymbol{K}\boldsymbol{x}_0 + \boldsymbol{e}$, $\boldsymbol{e} \sim \mathcal{N}(0, \sigma_{\boldsymbol{y}}^2 \boldsymbol{I})$ with $\sigma_{\boldsymbol{y}} = 0.05$. The exact degradation parameters, preprocessing details, and hyperparameters are reported in Appendix C. Additional quantitative and qualitative results are provided in Appendices D and E, respectively. Precisely, Appendix D.5 reports results in the noiseless setting ($\sigma_{\boldsymbol{y}} = 0$), while Appendix D.6 presents performance across different noise levels.

We mainly evaluate reconstruction quality using average PSNR, SSIM and LPIPS values, computed on the standard test sets. Higher PSNR and SSIM values, and lower LPIPS values, indicate better reconstruction quality. The variability of the results over five independent runs is reported in Appendix D.7. In all quantitative tables, the best value in each column is shown in bold, and the second-best value is underlined.

All the experiments have been executed on an NVIDIA RTX 5080 GPU with 16 GB of VRAM, running PyTorch 2.11 with CUDA 13.0. The code to reproduce the experiments is available at: `https://github.com/devangelista2/LAMP-Diff`.

4.2 Main results and discussion

Table 1 reports the noisy-setting comparison across all four backbones. DPS-LAMP and ΠGDM-LAMP improve their respective backbones in PSNR, SSIM, and LPIPS for every dataset and task considered. DDRM-LAMP also improves DDRM in all of these paired comparisons, with particularly clear gains in LPIPS. The effect on DiffPIR is smaller and more metric-dependent. Although LAMP improves ΠGDM in motion deblurring, the absolute performance of this backbone and its LAMP variant remains low in that task. DiffPIR-LAMP improves PSNR in all reported settings, but LPIPS worsens in several cases, indicating a perception-distortion trade-off rather than a uniform improvement. For example, on FFHQ motion deblurring, PSNR increases from 23.66 to 23.94 dB while LPIPS increases from 0.170 to 0.202. These paired improvements are obtained without increasing the number of denoising evaluations. Appendix D.2 reports the corresponding FID comparison: LAMP reduces FID in 35 of the 36 backbone-task-dataset pairs, with the exception of DPS on CelebA $\times 4$ super-resolution. For readability, the main table focuses on paired backbone-LAMP comparisons; the 1M and 2M variants are deferred to the appendix. Those comparisons help distinguish the proposed residual correction from simply increasing the order of an ODE discretization.

Figure 2 shows some qualitative results for noisy motion deblurring, a particularly challenging task where temporal fluctuations in the correction sequence can significantly affect reconstruction quality. The main figure retains a compact comparison of DiffPIR, DDRM against their LAMP version. Additional qualitative examples for DPS and ΠGDM are reported in Appendix E. The restored images illustrate how the effect of LAMP varies across image structures and backbones. In the first row, DDRM-LAMP better preserves hair and produces a more natural, uniform skin texture than the smoother DDRM output, whereas DiffPIR-LAMP reduces the graininess visible in DiffPIR. In the second row, the faces have reduced blurring, and the caps exhibit sharper boundaries and more coherent shading with LAMP. In the third row, the iguana's texture is more clearly recovered: DiffPIR-LAMP enhances the contrast and definition of its scales, while DDRM-LAMP yields sharper dark details along the back of the neck.

Overall, both quantitative and qualitative results support the main claim of the paper: temporal filtering of the measurement-aware estimate improves posterior sampling. DDRM-LAMP, DPS-LAMP, and ΠGDM-LAMP improve all three metrics in the paired comparisons, while DiffPIR-LAMP yields smaller changes and can trade improved distortion metrics for worse LPIPS. This suggests that the

CelebA		**Deblur (Gaussian)**			**Deblur (Motion)**			**SR** ($\times 4$)		
Method	NFE	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DPS [1]	1000	26.45	0.720	0.095	23.74	0.659	0.134	25.88	0.710	0.118
DiffPIR [18]	100	24.84	0.705	0.107	23.20	0.567	0.254	26.90	0.690	0.202
DDRM [6]	20	26.11	0.789	0.169	25.52	0.770	0.188	28.97	0.818	0.131
ΠGDM [12]	100	27.86	0.773	0.080	10.66	0.260	0.737	28.29	0.793	0.077
DPS-LAMP (Ours)	1000	26.65	0.725	0.090	23.91	0.664	**0.127**	26.09	0.715	0.113
DiffPIR-LAMP (Ours)	100	25.04	0.716	0.106	23.26	0.566	0.230	27.11	0.699	0.191
DDRM-LAMP (Ours)	20	26.14	**0.791**	0.118	**25.55**	**0.772**	0.129	**29.12**	**0.823**	0.105
ΠGDM-LAMP (Ours)	100	**28.36**	0.787	**0.073**	11.53	0.334	0.663	28.74	0.805	**0.070**
FFHQ		**Deblur (Gaussian)**			**Deblur (Motion)**			**SR** ($\times 4$)		
Method	NFE	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DPS [1]	1000	25.35	0.706	0.142	21.58	0.595	0.204	24.27	0.676	0.186
DiffPIR [18]	100	24.84	0.741	0.132	23.66	0.696	**0.170**	26.54	0.676	0.239
DDRM [6]	20	25.51	0.776	0.238	24.48	0.732	0.249	27.94	0.804	0.189
ΠGDM [12]	100	25.45	0.729	0.136	8.30	0.155	0.852	27.27	0.772	0.112
DPS-LAMP (Ours)	1000	25.98	0.732	0.138	22.06	0.608	0.193	24.70	0.694	0.173
DiffPIR-LAMP (Ours)	100	25.18	0.761	0.160	23.94	0.713	0.202	27.05	0.704	0.242
DDRM-LAMP (Ours)	20	25.55	**0.779**	0.164	**24.52**	**0.737**	0.181	**28.07**	**0.808**	0.146
ΠGDM-LAMP (Ours)	100	**26.42**	0.741	**0.129**	8.68	0.179	0.837	27.39	0.781	**0.110**
ImageNet		**Deblur (Gaussian)**			**Deblur (Motion)**			**SR** ($\times 4$)		
Method	NFE	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DPS [1]	1000	21.02	0.495	0.146	16.97	0.367	0.473	20.76	0.477	0.461
DiffPIR [18]	100	21.77	0.581	0.337	20.96	0.528	0.460	23.17	0.569	0.481
DDRM [6]	20	22.38	0.611	0.500	21.36	0.544	0.480	23.73	0.642	0.405
ΠGDM [12]	100	22.02	0.574	0.256	6.11	0.027	0.954	22.92	0.628	0.240
DPS-LAMP (Ours)	1000	21.19	0.515	**0.134**	18.07	0.390	0.422	21.59	0.495	0.436
DiffPIR-LAMP (Ours)	100	22.01	0.596	0.397	21.01	0.529	0.485	23.43	0.588	0.502
DDRM-LAMP (Ours)	20	22.41	**0.613**	0.416	**21.38**	**0.547**	**0.410**	**23.80**	**0.647**	0.372
ΠGDM-LAMP (Ours)	100	**22.61**	0.581	0.250	6.58	0.045	0.940	23.08	0.633	**0.236**

Table 1: Quantitative comparison in the noisy setting. Best values are in bold and second-best values are underlined.

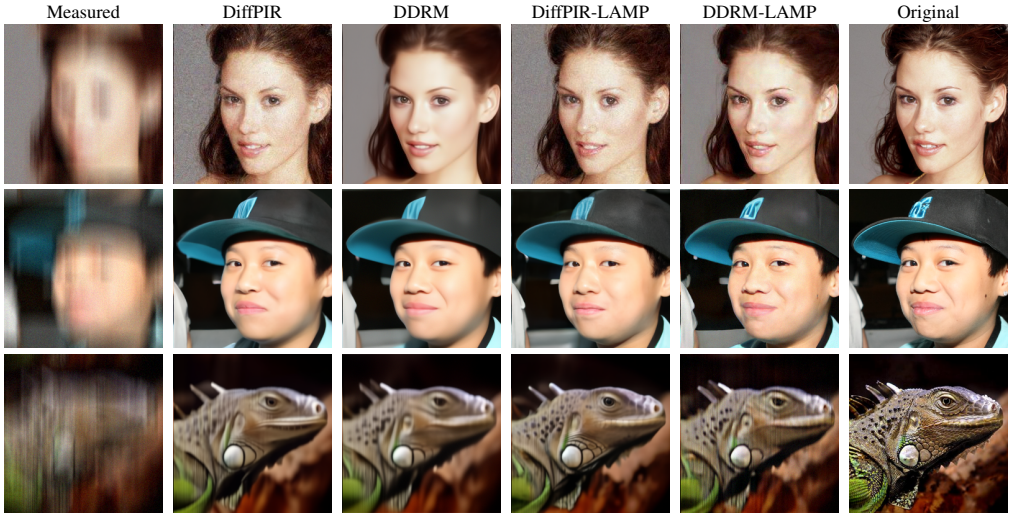

Figure 2: Qualitative comparison for noisy motion deblurring across different datasets.

benefit of LAMP depends on the temporal behavior of the correction sequence of the underlying sampler. Additional results in the appendices further support these observations.

4.3 Ablation on the lagged correction

While LAMP is designed around a *lagged* correction (i.e., $\gamma < 0$ in (14)), the formulation itself does not restrict γ to negative values. We therefore study its effect through an ablation analysis. Figure 3

(left) reports PSNR and SSIM as functions of the correction strength, with γ shown on the top axis, for DDRM-LAMP in the case of motion deblur with $\sigma_{\boldsymbol{y}} = 0.05$, averaged over FFHQ dataset. Recalling the effective coefficient $\beta_t := -\gamma \frac{A_1(h)}{h_{\text{prev}}}$ introduced in Proposition 1, which makes explicit the relation between $\widetilde{\boldsymbol{D}}_t$, $\boldsymbol{D}_t$, and $\boldsymbol{D}_{t+\Delta t}$, we additionally report on the bottom axis its average value $\bar{\beta}$ across iterations. In Figure 3 (right), the role of β_t upon each update is also illustrated schematically.

In this ablation, the highest PSNR is obtained with a mild lag, achieved by choosing $\gamma \approx -0.15$, matching γ of order $\mathcal{O}(1/h)$. This yields $\bar{\beta} \approx 0.03$, which corresponds to non-trivial yet controlled values of β_t at each time step, within the range $(0, 1)$ of moderate lagged corrections.

Specifically, if $\bar{\beta} \approx 1$, the updates exhibit a near-stalling behavior (i.e. $\tilde{\boldsymbol{D}}_t \approx \boldsymbol{D}_{t+\Delta t} \approx \tilde{\boldsymbol{D}}_{t+\Delta t}$), with the trajectory lingering around the last posterior iterate $\boldsymbol{x}_t^{\text{PS}}$ and the final solutions degrade.

For $\bar{\beta} > 1$ (i.e., large negative γ), the method introduces excessive lag, further ruining performance. Conversely, when $\bar{\beta} = 0$ (i.e. $\gamma = 0$), the LAMP rule reduces to the baseline PS update and the solutions are those previously discussed. At last, for $\bar{\beta} < 0$ (i.e., $\gamma > 0$), the update becomes extrapolating rather than lagged, which also degrades performance. These distortion results support a moderate lag in the tested setting.

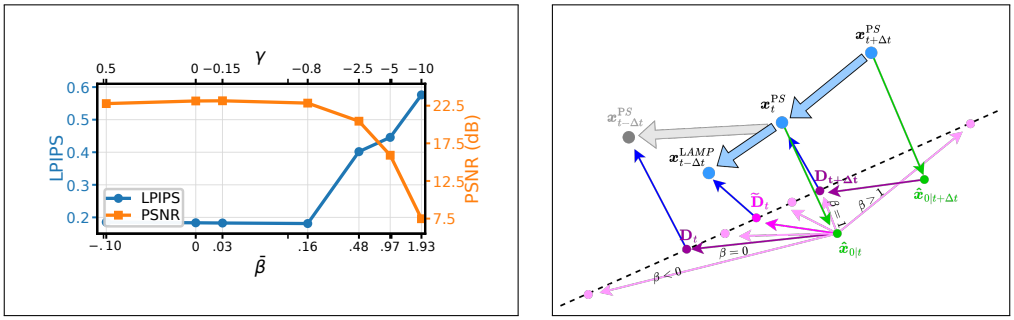

Figure 3: Ablation of the lagged correction strength. Left: PSNR and LPIPS as functions of the average coefficient $\bar{\beta}$ (with corresponding γ values shown on top). Right: geometric interpretation of the temporal correction, highlighting the role of the coefficient β as in (16) at each temporal update.

5 Limitations

LAMP is a plug-in extension of posterior sampling methods and therefore inherits their assumptions and limitations. In particular, it relies on the same modeling assumptions on the forward operator, noise distribution, and diffusion prior. As a result, it cannot compensate for failures of the underlying sampler, such as lack of convergence or model mismatch.

Our empirical evaluation is restricted to linear inverse problems. The algebraic filtering rule does not require an explicit linear decomposition of the forward operator, but it requires a compatible measurement-aware update and does not establish empirical effectiveness for nonlinear forward models.

The effectiveness of LAMP also depends on the behavior of the backbone method. Its benefits depend on the backbone and evaluation metric, and it does not fundamentally alter the reconstruction pattern. For example, DiffPIR-LAMP tends to preserve some of the characteristic noise patterns of DiffPIR, indicating that LAMP refines rather than replaces the underlying dynamics. Similarly, although ΠGDM-LAMP improves ΠGDM on motion deblurring, both retain low absolute reconstruction quality in that task. Consequently, LAMP is most beneficial when the baseline sampler already produces reasonably stable estimates.

Finally, the lag strength may require adjustment for a new backbone. In the reported experiments, it is fixed per backbone across datasets and tasks, and the warm-up is fixed rather than tuned separately. Automatically adapting these choices to the sampling trajectory remains a direction for future work.

6 Conclusion

We introduced LAMP, a second-order-inspired lagged multistep correction for compatible diffusion posterior samplers in image reconstruction problems. The method is motivated by the observation that standard posterior updates combine a first-order discretization with a residual forcing term, which can lead to unstable corrections when extended naively to multistep schemes. LAMP addresses this issue, and the resulting update can still be interpreted as a standard posterior sampler, applied to a filtered estimate.

Theoretical analysis characterizes a bias-variance trade-off, showing that LAMP improves the reverse transition when temporal smoothing reduces variability more than it introduces bias. Across DiffPIR, DDRM, DPS, and ΠGDM, a mild lag generally improves reconstruction quality without additional denoising evaluations. The paired PSNR, SSIM, and LPIPS gains are consistent for DDRM, DPS, and ΠGDM, whereas DiffPIR shows smaller, metric-dependent effects.

Future work includes adaptive lag selection, extensions to nonlinear settings, and a deeper analysis of temporally regularized diffusion dynamics.

Acknowledgments and Disclosure of Funding

This work has been partially supported by GNCS - Gruppo Nazionale per il Calcolo Scientifico on the project "Oltre il gradiente deterministico: analisi di dinamiche stocastiche per funzioni non convesse", INdAM GNCS Project, grant code CUP E53C25002010001, and by GNCS - Gruppo Nazionale per il Calcolo Scientifico on the project "Metodi data-driven convergenti per l'imaging biomedicale", INdAM GNCS Project, grant code CUP E53C25002010001. Finally, we thank Gianluca D'Alessandro and Salvatore Emanuel from Ferrari for their support to our co-author Francesco Pivi during this project.

References

[1] Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

[2] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022.

[3] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

[4] Davide Evangelista, Elena Loli Piccolomini, Elena Morotti, and James G Nagy. To be or not to be stable, that is the question: understanding neural networks for inverse problems. *SIAM Journal on Scientific Computing*, 47(1):C77–C99, 2025.

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

[6] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[7] Bahjat Kawar et al. Snips: Solving noisy inverse problems stochastically. In *NeurIPS*, 2021.

[8] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *ICLR*, 2022.

[9] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, 2025.

[10] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ICML*, 2015.

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

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

[13] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[14] Maurice CK Tweedie. Statistical properties of inverse gaussian distributions. i. *The Annals of Mathematical Statistics*, 28(2):362–377, 1957.

[15] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *International Conference on Learning Representations*, 2023.

[16] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G. Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[17] Qinsheng Zhang et al. Fast sampling of diffusion models with exponential integrator. *arXiv preprint*, 2023.

[18] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

A Definition of the Measurement-Aware Estimate for DiffPIR and DDRM, DPS, and ΠGDM

This section describes the measurement-aware estimate $\boldsymbol{D}_t$ for DiffPIR and DDRM and explains how an effective estimate can be identified for DPS and ΠGDM. In the main paper, we use the abstract notation

$$\boldsymbol{D}_t = \mathcal{C}_t(\hat{\boldsymbol{x}}_{0|t}, \boldsymbol{y}, \boldsymbol{K}), \tag{20}$$

where $\hat{\boldsymbol{x}}_{0|t}$ is the denoised estimate produced by the diffusion model at reverse time t, and $\mathcal{C}_t$ is the data-consistency correction associated with the chosen sampler. The purpose of this section is to make this definition explicit.

A.1 DiffPIR correction

DiffPIR can be interpreted as a plug-and-play diffusion method based on half-quadratic splitting. At each reverse step, the diffusion model produces the clean-image estimate

$$\hat{\boldsymbol{x}}_{0|t} = \frac{\boldsymbol{x}_t - \sigma_t \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\alpha_t}. \tag{21}$$

DiffPIR then computes a data-consistent reconstruction by solving a quadratic proximal problem that balances proximity to $\hat{\boldsymbol{x}}_{0|t}$ and consistency with the measurement model

$$\boldsymbol{y} = \boldsymbol{K}\boldsymbol{x}_0 + \boldsymbol{e}. \tag{22}$$

In our notation, the DiffPIR measurement-aware estimate is

$$\boldsymbol{D}_t^{\mathrm{DiffPIR}} = \arg\min_{\boldsymbol{x}} \frac{1}{2\sigma_{\boldsymbol{y}}^2}\|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{x}\|_2^2 + \frac{1}{2\tau_t^2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_{0|t}\|_2^2, \tag{23}$$

where $\tau_t > 0$ is the step-dependent regularization parameter induced by the DiffPIR schedule. Equivalently, defining $\mu_t := \frac{\sigma_{\boldsymbol{y}}^2}{\tau_t^2}$, the solution of (23) can be written as

$$\boldsymbol{D}_t^{\mathrm{DiffPIR}} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{x}\|_2^2 + \frac{\mu_t}{2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_{0|t}\|_2^2 \tag{24}$$

$$= \left(\boldsymbol{K}^T\boldsymbol{K} + \mu_t\boldsymbol{I}\right)^{-1}\left(\boldsymbol{K}^T\boldsymbol{y} + \mu_t\hat{\boldsymbol{x}}_{0|t}\right). \tag{25}$$

Thus, in DiffPIR, $\boldsymbol{D}_t$ is the proximal reconstruction obtained by pulling the denoised estimate toward the affine measurement constraint.

The exact implementation of (25) depends on the forward operator. For convolutional blur, the linear system is solved efficiently in the Fourier domain; for super-resolution, one can exploit the operator structure to solve it exactly.

A.2 DDRM spectral correction and operator decompositions

DDRM [6] is a posterior sampling method for linear inverse problems of the form

$$\boldsymbol{y} = \boldsymbol{K}\boldsymbol{x}_0 + \boldsymbol{e}, \qquad \boldsymbol{e} \sim \mathcal{N}(0, n_0^2\boldsymbol{I}), \tag{26}$$

where $n_0 = \sigma_{\boldsymbol{y}}$ denotes the observation noise level. Its key idea is to exploit a spectral decomposition of the linear degradation operator, so that the inverse problem decouples into one-dimensional problems. Let

$$\boldsymbol{K} = U\boldsymbol{\Sigma}V^T \tag{27}$$

be a singular value decomposition of $\boldsymbol{K}$, where $\boldsymbol{\Sigma}$ contains the singular values $\{a_i\}_{i=1}^r$. In the corresponding coordinates,

$$\bar{\boldsymbol{y}} := U^T\boldsymbol{y}, \qquad \bar{\boldsymbol{x}}_{0|t} := V^T\hat{\boldsymbol{x}}_{0|t}, \tag{28}$$

the measurement model becomes componentwise

$$\bar{y}_i = a_i\bar{x}_{0,i} + \bar{e}_i. \tag{29}$$

Thus, each spectral component can be corrected independently according to the singular value a_i and the relative strength of the diffusion and measurement noise.

In the variance-preserving parameterization used in this paper, the effective diffusion noise level at time t is

$$n_t := \frac{\sigma_t}{\alpha_t}. \tag{30}$$

DDRM combines the denoiser prediction $\hat{\boldsymbol{x}}_{0|t}$ with the measurement in the V-coordinates. Writing $\bar{\boldsymbol{D}}_{t,i}^{\mathrm{DDRM}} := \left(V^T \boldsymbol{D}_t^{\mathrm{DDRM}}\right)_i$, the deterministic measurement-aware estimate used by the posterior update is defined componentwise as follows:

- For components that are not observed by the forward operator, namely when $a_i = 0$, the measurement provides no information and DDRM keeps the prior estimate:

$$\bar{\boldsymbol{D}}_{t,i}^{\mathrm{DDRM}} = \bar{x}_{0|t,i}. \tag{31}$$

- For observed components satisfying

$$a_i n_t > n_0, \tag{32}$$

the current diffusion uncertainty is large enough compared with the observation noise that DDRM can safely move the estimate toward the pseudo-inverse measurement. In this regime,

$$\bar{\boldsymbol{D}}_{t,i}^{\mathrm{DDRM}} = \eta_B \frac{\bar{y}_i}{a_i} + (1 - \eta_B)\bar{x}_{0|t,i}, \tag{33}$$

where $\eta_B \in [0, 1]$ controls the strength of the data-consistency replacement. The choice $\eta_B = 1$ gives the fully data-consistent value $\bar{y}_i / a_i$.

- For observed components satisfying

$$a_i n_t \leq n_0, \tag{34}$$

the measurement noise is large relative to the current diffusion level along that spectral direction. DDRM therefore uses a residual correction rather than a direct pseudo-inverse replacement:

$$\bar{\boldsymbol{D}}_{t,i}^{\mathrm{DDRM}} = \bar{x}_{0|t,i} + \widetilde{n}_t \frac{\bar{y}_i - a_i \bar{x}_{0|t,i}}{n_0}, \tag{35}$$

with

$$\widetilde{n}_t = n_t \sqrt{1 - \eta^2}. \tag{36}$$

Here $\eta \in [0, 1]$ is the stochasticity parameter of DDRM. The complete DDRM sampler may additionally inject stochastic terms in the spectral coordinates, while the quantity $\boldsymbol{D}_t^{\mathrm{DDRM}}$ above is the deterministic measurement-aware target that enters the posterior update.

After applying the componentwise correction, the corrected estimate is mapped back to the image domain through

$$\boldsymbol{D}_t^{\mathrm{DDRM}} = V \bar{\boldsymbol{D}}_t^{\mathrm{DDRM}}. \tag{37}$$

This construction shows that DDRM balances prior information and data consistency in a spectral way: directions that are well constrained by the measurements are corrected toward the pseudo-inverse solution, while directions that are weakly observed, noisy, or in the null space remain closer to the diffusion prior.

One crucial limitation of DDRM is that it requires access to the SVD decomposition of the operator $\boldsymbol{K}$. For the inverse problems considered in this paper, the required spectral structure is available in closed form or through standard structured decompositions. In particular, for Gaussian and motion deblurring, the degradation operator is a convolution. Under circular boundary conditions, convolution is diagonalized by the two-dimensional discrete Fourier transform. If k is the blur kernel padded to the image size and

$$\widehat{k} = \mathcal{F}(k), \tag{38}$$

then

$$\mathcal{F}(\boldsymbol{K}\boldsymbol{x})(\omega) = \widehat{k}(\omega)\mathcal{F}(\boldsymbol{x})(\omega). \tag{39}$$

Writing

$$\widehat{k}(\omega) = a(\omega)e^{\mathrm{i}\phi(\omega)}, \qquad a(\omega) = |\widehat{k}(\omega)|, \tag{40}$$

the singular values are given by the Fourier magnitudes $a(\omega)$, while the phase factors are absorbed into the unitary matrices of the SVD. Equivalently, after rotating the Fourier-domain measurements by the conjugate phase, each frequency obeys a scalar degradation model with singular value $a(\omega)$. DDRM then applies the same componentwise correction rules in (31), (33), and (35) frequency by frequency, and the corrected estimate is transformed back to the image domain by the inverse Fourier transform. This Fourier representation is exact for circular convolution. Therefore, the DDRM formulation for deblurring should be understood under the circular-convolution model associated with the padded blur kernel. With other boundary conditions, such as zero or reflection padding, the Fourier diagonalization is no longer exact, and the same construction becomes an approximation unless the corresponding operator is explicitly diagonalized or otherwise adapted.

For super-resolution by a factor r, the forward operator averages each non-overlapping $r \times r$ block and then downsamples. The local degradation acting on one block is the rank-one operator

$$H_{\mathrm{SR}} = \begin{bmatrix} \frac{1}{r^2} & \frac{1}{r^2} & \cdots & \frac{1}{r^2} \end{bmatrix} \in \mathbb{R}^{1\times r^2}. \tag{41}$$

This operator admits a simple blockwise SVD: one direction, corresponding to the average intensity in the block, has a nonzero singular value, while the remaining $r^2 - 1$ directions span the null space. Applying this decomposition independently across blocks and channels yields a structured spectral representation of the full super-resolution operator. DDRM therefore corrects the block-average component using the measurement, while the high-frequency within-block components are sampled from the diffusion prior.

In our framework, LAMP does not modify the DDRM correction itself. Instead, it takes the DDRM measurement-aware target $\boldsymbol{D}_t^{\mathrm{DDRM}}$ and replaces it by the lagged estimate

$$\widetilde{\boldsymbol{D}}_t^{\mathrm{DDRM}} = (1-\beta_t)\boldsymbol{D}_t^{\mathrm{DDRM}} + \beta_t \boldsymbol{D}_{t+\Delta t}^{\mathrm{DDRM}}, \tag{42}$$

which is then inserted into the same posterior sampling update:

$$\boldsymbol{x}_{t-\Delta t}^{\mathrm{DDRM-LAMP}} = \alpha_{t-\Delta t}\widetilde{\boldsymbol{D}}_t^{\mathrm{DDRM}} + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t). \tag{43}$$

Thus, DDRM-LAMP remains a DDRM-type posterior sampler, but with the instantaneous spectral correction replaced by a temporally filtered one.

A.3 DPS correction

DPS [1] enforces measurement consistency through the gradient of a discrepancy evaluated at the denoised prediction. In particular, we consider:

$$g_t^{\mathrm{DPS}} := \nabla_{\boldsymbol{x}_t}\left\|\boldsymbol{y} - \boldsymbol{K}\hat{\boldsymbol{x}}_{0|t}\right\|_2^2 = \left(\frac{\partial \hat{\boldsymbol{x}}_{0|t}}{\partial \boldsymbol{x}_t}\right)^T \boldsymbol{K}^T(\boldsymbol{K}\hat{\boldsymbol{x}}_{0|t} - \boldsymbol{y}). \tag{44}$$

The denoised prediction is clipped before this gradient is evaluated. On the ancestral DDPM path, let $c_{1,t}$ and $c_{2,t}$ be the coefficients of $\hat{\boldsymbol{x}}_{0|t}$ and $\boldsymbol{x}_t$ in the posterior mean, and let ξ_t denote the sampled innovation. With guidance strength ζ, the DPS step is

$$\boldsymbol{x}_{t-\Delta t}^{\mathrm{DPS}} = c_{1,t}\hat{\boldsymbol{x}}_{0|t} + c_{2,t}\boldsymbol{x}_t + \xi_t - \zeta g_t^{\mathrm{DPS}} = c_{1,t}\boldsymbol{D}_t^{\mathrm{DPS}} + c_{2,t}\boldsymbol{x}_t + \xi_t, \tag{45}$$

$$\boldsymbol{D}_t^{\mathrm{DPS}} := \hat{\boldsymbol{x}}_{0|t} - \frac{\zeta}{c_{1,t}} g_t^{\mathrm{DPS}}. \tag{46}$$

Thus, the correction is absorbed into the coefficient of the clean-image estimate. DPS-LAMP filters this estimate with its previous *unfiltered* value, $\widetilde{\boldsymbol{D}}_t^{\mathrm{DPS}} = (1-\beta_t)\boldsymbol{D}_t^{\mathrm{DPS}} + \beta_t \boldsymbol{D}_{t+\Delta t}^{\mathrm{DPS}}$, which adds $c_{1,t}\beta_t(\boldsymbol{D}_{t+\Delta t}^{\mathrm{DPS}} - \boldsymbol{D}_t^{\mathrm{DPS}})$ to the ancestral step. The supplied solver also has a deterministic DDIM path: there $c_{1,t}$ is replaced by $\alpha_{t-\Delta t}$, and the reverse update has the form of Eq. (1).

A.4 ΠGDM correction

ΠGDM [12] guides the reverse step through an approximation to the conditional score. For the linear forward operator used in this paper, we first form the measurement residual $\boldsymbol{y} - \boldsymbol{K}\hat{\boldsymbol{x}}_{0|t}$, and then we map it back to image space as

$$m_t := r_t^2 \boldsymbol{K}^T\left(r_t^2 \boldsymbol{K}\boldsymbol{K}^T + \sigma_{\boldsymbol{y},\mathrm{eff}}^2 \boldsymbol{I}\right)^{-1}(\boldsymbol{y} - \boldsymbol{K}\hat{\boldsymbol{x}}_{0|t}), \qquad r_t^2 = 1 - \alpha_t^2. \tag{47}$$

Here $\sigma_{\boldsymbol{y},\mathrm{eff}}$ is the measurement-noise level used by the sampler after conversion to model coordinates. In the noiseless case, this correction reduces to the pseudoinverse of the residual for the well-conditioned operators; the blur implementation retains a regularization floor. The conditional guidance is the vector-Jacobian product $J_t^T m_t$, where $J_t = \partial \hat{\boldsymbol{x}}_{0|t}/\partial \boldsymbol{x}_t$ and m_t is held fixed during differentiation. Given a guidance weight parameter ζ, the measurement-aware estimate and reverse update are

$$\boldsymbol{D}_t^{\Pi\mathrm{GDM}} := \hat{\boldsymbol{x}}_{0|t} + \alpha_t \zeta J_t^T m_t, \qquad \boldsymbol{x}_{t-\Delta t}^{\Pi\mathrm{GDM}} = \alpha_{t-\Delta t}\boldsymbol{D}_t^{\Pi\mathrm{GDM}} + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t). \tag{48}$$

ΠGDM-LAMP replaces $\boldsymbol{D}_t^{\Pi\mathrm{GDM}}$ by $(1-\beta_t)\boldsymbol{D}_t^{\Pi\mathrm{GDM}} + \beta_t \boldsymbol{D}_{t+\Delta t}^{\Pi\mathrm{GDM}}$ in this update, again storing the previous unfiltered estimate. This leaves the pseudoinverse guidance and denoiser unchanged.

B Proofs

This section provides the full derivations of the main structural identities used in the methodological section 3. In the following, we use the same notation as in the main paper. In particular, t and $t-\Delta t$ denote two consecutive reverse diffusion times, $h = \lambda_{t-\Delta t} - \lambda_t > 0$, and

$$\lambda_t = \log\left(\frac{\alpha_t}{\sigma_t}\right).$$

B.1 Proof of Eq. (6)

While this result is not new, we report for completeness the derivation of the integrating-factor formulation. For a variance-preserving diffusion, the probability-flow ODE can be written in the clean-image parameterization as

$$\frac{d\boldsymbol{x}}{dt} = -\frac{1}{2}\beta(t)\boldsymbol{x} + \frac{1}{2}\beta(t)\frac{\boldsymbol{x} - \alpha_t \hat{\boldsymbol{x}}_0(\boldsymbol{x}, t)}{\sigma_t^2}. \tag{49}$$

Using the variance-preserving identities

$$\frac{d\alpha_t}{dt} = -\frac{1}{2}\beta(t)\alpha_t, \qquad \frac{d\sigma_t^2}{dt} = \beta(t)\alpha_t^2, \tag{50}$$

one obtains

$$\frac{d\lambda_t}{dt} = -\frac{1}{2}\beta(t)\left(1 + \frac{\alpha_t^2}{\sigma_t^2}\right) = -\frac{\beta(t)}{2\sigma_t^2}, \tag{51}$$

where the last equality uses $\alpha_t^2 + \sigma_t^2 = 1$. Reparametrizing (49) by λ gives

$$\frac{d\boldsymbol{x}}{d\lambda} = \frac{d\boldsymbol{x}/dt}{d\lambda/dt} = \boldsymbol{x} - \alpha_\lambda \hat{\boldsymbol{x}}_0(\boldsymbol{x}, \lambda) - \sigma_\lambda^2 \boldsymbol{x}. \tag{52}$$

Since $1 - \sigma_\lambda^2 = \alpha_\lambda^2$, this becomes

$$\frac{d\boldsymbol{x}}{d\lambda} = \alpha_\lambda^2 \boldsymbol{x} - \alpha_\lambda \hat{\boldsymbol{x}}_0(\boldsymbol{x}, \lambda). \tag{53}$$

Depending on the convention used to orient the reverse-time integration, the same equation is equivalently written with the integration limits ordered from the noisier time t to the cleaner time $t - \Delta t$. Under the convention adopted in the main paper, this yields the integrating-factor identity

$$\frac{d}{d\lambda}\left(\frac{\boldsymbol{x}(\lambda)}{\sigma(\lambda)}\right) = \frac{\alpha(\lambda)}{\sigma(\lambda)}\hat{\boldsymbol{x}}_0(\boldsymbol{x}(\lambda), \lambda). \tag{54}$$

The mild formulation follows by integrating (54):

$$\boldsymbol{x}_{t-\Delta t} = \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \sigma_{t-\Delta t}\int_{\lambda_t}^{\lambda_{t-\Delta t}} \frac{\alpha(\lambda)}{\sigma(\lambda)}\hat{\boldsymbol{x}}_0(\boldsymbol{x}(\lambda), \lambda)\, d\lambda. \tag{55}$$

Approximating $\hat{\boldsymbol{x}}_0(\boldsymbol{x}(\lambda), \lambda)$ by $\hat{\boldsymbol{x}}_{0|t}$ on the integration interval gives exactly (7).

B.2 Proof of Eq. 10

The deterministic DDIM step is

$$\boldsymbol{x}_{t-\Delta t} = \alpha_{t-\Delta t}\hat{\boldsymbol{x}}_{0|t} + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t), \tag{56}$$

where

$$\hat{\boldsymbol{x}}_{0|t} = \frac{\boldsymbol{x}_t - \sigma_t\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t)}{\alpha_t}. \tag{57}$$

Rearranging (57) gives

$$\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_t - \alpha_t\hat{\boldsymbol{x}}_{0|t}}{\sigma_t}. \tag{58}$$

Substituting (58) into (56), we obtain

$$\begin{aligned}\boldsymbol{x}_{t-\Delta t} &= \alpha_{t-\Delta t}\hat{\boldsymbol{x}}_{0|t} + \sigma_{t-\Delta t}\frac{\boldsymbol{x}_t - \alpha_t\hat{\boldsymbol{x}}_{0|t}}{\sigma_t} \\ &= \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \left(\alpha_{t-\Delta t} - \frac{\sigma_{t-\Delta t}\alpha_t}{\sigma_t}\right)\hat{\boldsymbol{x}}_{0|t}.\end{aligned} \tag{59}$$

Using the definition of the log-SNR,

$$e^{-h} = e^{-(\lambda_{t-\Delta t}-\lambda_t)} = \frac{\alpha_t\sigma_{t-\Delta t}}{\alpha_{t-\Delta t}\sigma_t}, \tag{60}$$

we have

$$\frac{\sigma_{t-\Delta t}\alpha_t}{\sigma_t} = \alpha_{t-\Delta t}e^{-h}. \tag{61}$$

Therefore,

$$\alpha_{t-\Delta t} - \frac{\sigma_{t-\Delta t}\alpha_t}{\sigma_t} = \alpha_{t-\Delta t}(1-e^{-h}) = \alpha_{t-\Delta t}A_0(h), \qquad A_0(h) = 1-e^{-h}. \tag{62}$$

Substituting this into (59) gives

$$\boldsymbol{x}_{t-\Delta t} = \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \alpha_{t-\Delta t}A_0(h)\hat{\boldsymbol{x}}_{0|t}. \tag{63}$$

This is the first-order exponential-integrator form of DDIM.

B.3 Proof of Eq. (11)

We now prove the structural identity used to motivate LAMP. The generic posterior sampler step is

$$\boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t} = \alpha_{t-\Delta t}\boldsymbol{D}_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t). \tag{64}$$

Using again

$$\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_t - \alpha_t\hat{\boldsymbol{x}}_{0|t}}{\sigma_t}, \tag{65}$$

we obtain

$$\begin{aligned}\boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t} &= \alpha_{t-\Delta t}\boldsymbol{D}_t + \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t - \frac{\sigma_{t-\Delta t}\alpha_t}{\sigma_t}\hat{\boldsymbol{x}}_{0|t} \\ &= \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \alpha_{t-\Delta t}\boldsymbol{D}_t - \alpha_{t-\Delta t}e^{-h}\hat{\boldsymbol{x}}_{0|t},\end{aligned} \tag{66}$$

where we used (61). We now add and subtract $\alpha_{t-\Delta t}e^{-h}\boldsymbol{D}_t$:

$$\begin{aligned}\boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t} &= \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \alpha_{t-\Delta t}\boldsymbol{D}_t - \alpha_{t-\Delta t}e^{-h}\boldsymbol{D}_t + \alpha_{t-\Delta t}e^{-h}\boldsymbol{D}_t - \alpha_{t-\Delta t}e^{-h}\hat{\boldsymbol{x}}_{0|t} \\ &= \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \alpha_{t-\Delta t}(1-e^{-h})\boldsymbol{D}_t + \alpha_{t-\Delta t}e^{-h}\left(\boldsymbol{D}_t - \hat{\boldsymbol{x}}_{0|t}\right) \\ &= \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \alpha_{t-\Delta t}A_0(h)\boldsymbol{D}_t + \alpha_{t-\Delta t}e^{-h}\left(\boldsymbol{D}_t - \hat{\boldsymbol{x}}_{0|t}\right).\end{aligned} \tag{67}$$

This proves the decomposition. The first two terms coincide with the first-order exponential step associated with the modified continuous equation driven by $\boldsymbol{D}_\lambda$, while the last term is the additional posterior forcing term.

B.4 Proof of Proposition 1

The LAMP update is defined as

$$\boldsymbol{x}_{t-\Delta t}^{\mathrm{LAMP}} = \boldsymbol{x}_{t-\Delta t}^{\mathrm{PS}} + \alpha_{t-\Delta t} A_1(h)\gamma \frac{\boldsymbol{D}_t - \boldsymbol{D}_{t+\Delta t}}{h_{\mathrm{prev}}}, \qquad \gamma < 0. \tag{68}$$

Using

$$\boldsymbol{x}_{t-\Delta t}^{\mathrm{PS}} = \alpha_{t-\Delta t}\boldsymbol{D}_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t), \tag{69}$$

we get

$$\begin{aligned}\boldsymbol{x}_{t-\Delta t}^{\mathrm{LAMP}} &= \alpha_{t-\Delta t}\boldsymbol{D}_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t) + \alpha_{t-\Delta t}A_1(h)\gamma\frac{\boldsymbol{D}_t - \boldsymbol{D}_{t+\Delta t}}{h_{\mathrm{prev}}} \\ &= \alpha_{t-\Delta t}\left[\boldsymbol{D}_t + \gamma\frac{A_1(h)}{h_{\mathrm{prev}}}(\boldsymbol{D}_t - \boldsymbol{D}_{t+\Delta t})\right] + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t).\end{aligned} \tag{70}$$

Define

$$\beta_t := -\gamma\frac{A_1(h)}{h_{\mathrm{prev}}}. \tag{71}$$

Since $\gamma < 0$ and $A_1(h)/h_{\mathrm{prev}} > 0$ under the reverse schedule used in the main paper, we have $\beta_t > 0$. The term in brackets in (70) becomes

$$\boldsymbol{D}_t - \beta_t(\boldsymbol{D}_t - \boldsymbol{D}_{t+\Delta t}) = (1-\beta_t)\boldsymbol{D}_t + \beta_t\boldsymbol{D}_{t+\Delta t}. \tag{72}$$

Defining

$$\widetilde{\boldsymbol{D}}_t := (1-\beta_t)\boldsymbol{D}_t + \beta_t\boldsymbol{D}_{t+\Delta t}, \tag{73}$$

we obtain

$$\boldsymbol{x}_{t-\Delta t}^{\mathrm{LAMP}} = \alpha_{t-\Delta t}\widetilde{\boldsymbol{D}}_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t). \tag{74}$$

Applying the posterior forcing identity (67) with $\boldsymbol{D}_t$ replaced by $\widetilde{\boldsymbol{D}}_t$ also gives

$$\boldsymbol{x}_{t-\Delta t}^{\mathrm{LAMP}} = \frac{\sigma_{t-\Delta t}}{\sigma_t}\boldsymbol{x}_t + \alpha_{t-\Delta t}A_0(h)\widetilde{\boldsymbol{D}}_t + \alpha_{t-\Delta t}e^{-h}\left(\widetilde{\boldsymbol{D}}_t - \hat{\boldsymbol{x}}_{0|t}\right). \tag{75}$$

This proves that LAMP preserves the posterior sampling structure.

B.5 Variance reduction under the local stationarity model

In this section we prove that, under the same assumptions of Proposition 2, the LAMP-modified correction term $\widetilde{\boldsymbol{D}}_t = (1-\beta)\boldsymbol{D}_t + \beta\boldsymbol{D}_{t+\Delta t}$ has less variance then $\boldsymbol{D}_t$. We consider the local error model

$$\boldsymbol{D}_t = \mu_t + \eta_t, \qquad \boldsymbol{D}_{t+\Delta t} = \mu_{t+\Delta t} + \eta_{t+\Delta t}, \tag{76}$$

with

$$\mathbb{E}[\eta_t] = \mathbb{E}[\eta_{t+\Delta t}] = 0, \qquad \mathrm{Var}(\eta_t) = \mathrm{Var}(\eta_{t+\Delta t}) = \Sigma_t, \qquad \mathrm{Cov}(\eta_t, \eta_{t+\Delta t}) = \rho_t\Sigma_t. \tag{77}$$

The lagged estimate is

$$\widetilde{\boldsymbol{D}}_t = (1-\beta_t)\boldsymbol{D}_t + \beta_t\boldsymbol{D}_{t+\Delta t}. \tag{78}$$

The fluctuation part of $\widetilde{\boldsymbol{D}}_t$ is

$$\widetilde{\eta}_t = (1-\beta_t)\eta_t + \beta_t\eta_{t+\Delta t}. \tag{79}$$

Therefore,

$$\begin{aligned}\mathrm{Var}(\widetilde{\eta}_t) &= (1-\beta_t)^2\Sigma_t + \beta_t^2\Sigma_t + 2\beta_t(1-\beta_t)\rho_t\Sigma_t \\ &= \left[(1-\beta_t)^2 + \beta_t^2 + 2\beta_t(1-\beta_t)\rho_t\right]\Sigma_t \\ &= \left[1 - 2\beta_t(1-\beta_t)(1-\rho_t)\right]\Sigma_t.\end{aligned} \tag{80}$$

If $0 < \beta_t < 1$ and $\rho_t < 1$, then

$$1 - 2\beta_t(1-\beta_t)(1-\rho_t) < 1, \tag{81}$$

so the lagged estimate has smaller local error variance than the instantaneous estimate.

B.6 Proof of Proposition 2

Let

$$\boldsymbol{x}^{\mu}_{t-\Delta t} := \alpha_{t-\Delta t}\mu_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t) \tag{82}$$

be the ideal one-step target associated with the local posterior target μ_t. The base posterior step is

$$\boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t} = \alpha_{t-\Delta t}\boldsymbol{D}_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t). \tag{83}$$

Thus,

$$\boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t} - \boldsymbol{x}^{\mu}_{t-\Delta t} = \alpha_{t-\Delta t}(\boldsymbol{D}_t - \mu_t) = \alpha_{t-\Delta t}\eta_t. \tag{84}$$

Taking expectations gives

$$\mathbb{E}\|\boldsymbol{x}^{\mathrm{PS}}_{t-\Delta t} - \boldsymbol{x}^{\mu}_{t-\Delta t}\|^2 = \alpha^2_{t-\Delta t}\mathbb{E}\|\eta_t\|^2 = \alpha^2_{t-\Delta t}\mathrm{tr}(\Sigma_t). \tag{85}$$

For LAMP, we have

$$\boldsymbol{x}^{\mathrm{LAMP}}_{t-\Delta t} = \alpha_{t-\Delta t}\widetilde{\boldsymbol{D}}_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t), \tag{86}$$

where

$$\widetilde{\boldsymbol{D}}_t = (1-\beta_t)\boldsymbol{D}_t + \beta_t\boldsymbol{D}_{t+\Delta t}. \tag{87}$$

Let

$$r_t := \mu_{t+\Delta t} - \mu_t. \tag{88}$$

Then

$$\begin{aligned}\widetilde{\boldsymbol{D}}_t - \mu_t &= (1-\beta_t)(\mu_t + \eta_t) + \beta_t(\mu_{t+\Delta t} + \eta_{t+\Delta t}) - \mu_t \\ &= (1-\beta_t)\eta_t + \beta_t\eta_{t+\Delta t} + \beta_t(\mu_{t+\Delta t} - \mu_t) \\ &= (1-\beta_t)\eta_t + \beta_t\eta_{t+\Delta t} + \beta_t r_t. \end{aligned} \tag{89}$$

Since η_t and $\eta_{t+\Delta t}$ are zero-mean, the cross terms between the stochastic error and the deterministic drift vanish in expectation. Therefore,

$$\begin{aligned}\mathbb{E}\|\widetilde{\boldsymbol{D}}_t - \mu_t\|^2 &= \mathbb{E}\left\|(1-\beta_t)\eta_t + \beta_t\eta_{t+\Delta t}\right\|^2 + \beta_t^2\|r_t\|^2 \\ &= \left[1 - 2\beta_t(1-\beta_t)(1-\rho_t)\right]\mathrm{tr}(\Sigma_t) + \beta_t^2\|r_t\|^2, \end{aligned} \tag{90}$$

where the last equality follows from (80). Multiplying by $\alpha^2_{t-\Delta t}$ gives

$$\mathbb{E}\|\boldsymbol{x}^{\mathrm{LAMP}}_{t-\Delta t} - \boldsymbol{x}^{\mu}_{t-\Delta t}\|^2 = \alpha^2_{t-\Delta t}\left[(1 - 2\beta_t(1-\beta_t)(1-\rho_t))\,\mathrm{tr}(\Sigma_t) + \beta_t^2\|r_t\|^2\right]. \tag{91}$$

Comparing (91) with (85), LAMP improves the one-step risk whenever

$$\left[1 - 2\beta_t(1-\beta_t)(1-\rho_t)\right]\mathrm{tr}(\Sigma_t) + \beta_t^2\|r_t\|^2 < \mathrm{tr}(\Sigma_t). \tag{92}$$

After rearranging and dividing by $\beta_t > 0$, this becomes

$$\beta_t\|r_t\|^2 < 2(1-\beta_t)(1-\rho_t)\mathrm{tr}(\Sigma_t). \tag{93}$$

This proves the proposition.

B.7 Proof of generalized version of Proposition 2

We now remove the local equal-variance assumption. Assume

$$\boldsymbol{D}_t = \mu_t + \eta_t, \qquad \boldsymbol{D}_{t+\Delta t} = \mu_{t+\Delta t} + \eta_{t+\Delta t}, \qquad r_t := \mu_{t+\Delta t} - \mu_t, \tag{94}$$

with

$$\mathbb{E}[\eta_t] = \mathbb{E}[\eta_{t+\Delta t}] = 0, \qquad \Sigma_t := \mathrm{Var}(\eta_t), \qquad \Sigma_{t+\Delta t} := \mathrm{Var}(\eta_{t+\Delta t}), \tag{95}$$

and

$$C_t := \mathrm{Cov}(\eta_t, \eta_{t+\Delta t}). \tag{96}$$

The lagged estimate satisfies

$$\widetilde{\boldsymbol{D}}_t - \mu_t = (1-\beta_t)\eta_t + \beta_t\eta_{t+\Delta t} + \beta_t r_t. \tag{97}$$

Therefore,

$$\mathbb{E}\|\widetilde{\boldsymbol{D}}_t - \mu_t\|^2 = \beta_t^2\|r_t\|^2 + (1-\beta_t)^2\mathrm{tr}(\Sigma_t) + \beta_t^2\mathrm{tr}(\Sigma_{t+\Delta t}) + 2\beta_t(1-\beta_t)\mathrm{tr}(C_t). \tag{98}$$

The base estimate has risk

$$\mathbb{E}\|\boldsymbol{D}_t - \mu_t\|^2 = \mathrm{tr}(\Sigma_t). \tag{99}$$

Thus, the lagged estimate improves over the instantaneous estimate whenever

$$\beta_t^2\|r_t\|^2 + (1-\beta_t)^2\mathrm{tr}(\Sigma_t) + \beta_t^2\mathrm{tr}(\Sigma_{t+\Delta t}) + 2\beta_t(1-\beta_t)\mathrm{tr}(C_t) < \mathrm{tr}(\Sigma_t). \tag{100}$$

Equivalently,

$$2\beta_t\mathrm{tr}(\Sigma_t) > \beta_t^2\|r_t\|^2 + \beta_t^2\mathrm{tr}(\Sigma_t) + \beta_t^2\mathrm{tr}(\Sigma_{t+\Delta t}) + 2\beta_t(1-\beta_t)\mathrm{tr}(C_t). \tag{101}$$

For $\beta_t > 0$, this can be written as

$$\beta_t\left(\|r_t\|^2 + \mathrm{tr}(\Sigma_t) + \mathrm{tr}(\Sigma_{t+\Delta t})\right) < 2\left(\mathrm{tr}(\Sigma_t) - (1-\beta_t)\mathrm{tr}(C_t)\right). \tag{102}$$

The corresponding state-level comparison is obtained by multiplying both sides of the estimator risks by $\alpha_{t-\Delta t}^2$.

C Implementation Details and Hyperparameters

This section reports the implementation details used in the experiments.

C.1 LAMP Update Pseudocode

At each reverse step, the base sampler computes the measurement-aware estimate

$$\boldsymbol{D}_t = \mathcal{C}_t(\hat{\boldsymbol{x}}_{0|t}, \boldsymbol{y}, \boldsymbol{K}). \tag{103}$$

After a warm-up phase, LAMP replaces $\boldsymbol{D}_t$ with

$$\widetilde{\boldsymbol{D}}_t = (1-\beta_t)\boldsymbol{D}_t + \beta_t\boldsymbol{D}_{t+\Delta t}, \tag{104}$$

and then performs the posterior update

$$\boldsymbol{x}_{t-\Delta t}^{\mathrm{LAMP}} = \alpha_{t-\Delta t}\widetilde{\boldsymbol{D}}_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t). \tag{105}$$

Here, $t+\Delta t$ denotes the previous reverse time, so that $\boldsymbol{D}_{t+\Delta t}$ is already available. The filter uses the previous *unfiltered* measurement-aware estimate. Its effective weight $\beta_t = -\gamma A_1(h)/h_{\mathrm{prev}}$ varies with the reverse schedule. We activate LAMP from the third reverse update onward.

Algorithm 1 LAMP posterior sampling

Input: measurement $\boldsymbol{y}$, operator $\boldsymbol{K}$, pretrained denoiser $\boldsymbol{\epsilon}_\Theta$, timestep $\Delta t > 0$, backbone correction $\mathcal{C}_t$, correction strength parameter $\gamma < 0$, warm-up length N_{warm}
Initialize $\boldsymbol{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$
Set $\boldsymbol{D}_{\text{prev}} = \texttt{None}$
for $i = N, N-1, \ldots, 1$ **do**
  $t \leftarrow i\Delta t$
  $\hat{\boldsymbol{x}}_{0|t} \leftarrow (\boldsymbol{x}_t - \sigma_t \boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t))/\alpha_t$
  $\boldsymbol{D}_t \leftarrow \mathcal{C}_t(\hat{\boldsymbol{x}}_{0|t}, \boldsymbol{y}, \boldsymbol{K})$
  if $i < N - N_{\text{warm}}$ and $\boldsymbol{D}_{\text{prev}} \neq \texttt{None}$ **then**
    $\beta_t \leftarrow -\gamma \frac{A_1(h)}{h_{\text{prev}}}$
    $\widetilde{\boldsymbol{D}}_t \leftarrow (1-\beta_t)\boldsymbol{D}_t + \beta_t \boldsymbol{D}_{\text{prev}}$
  else
    $\widetilde{\boldsymbol{D}}_t \leftarrow \boldsymbol{D}_t$
  end if
  $\boldsymbol{x}_{t-\Delta t} \leftarrow \alpha_{t-\Delta t}\widetilde{\boldsymbol{D}}_t + \sigma_{t-\Delta t}\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t)$
  $\boldsymbol{D}_{\text{prev}} \leftarrow \boldsymbol{D}_t$
end for
return $\boldsymbol{x}_{t_0}$

C.2 Hyperparameters

Dataset	Task	Backbone	NFE	N_{warm}	γ	μ	ζ	η	η_B
Noiseless ($\sigma_y = 0$)									
FFHQ	Gaussian deblur	DiffPIR+LAMP	100	3	-3.0	7.0	0.3	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	Motion deblur	DiffPIR+LAMP	100	3	-3.0	7.0	0.4	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	SR $\times 4$	DiffPIR+LAMP	100	3	-3.0	7.0	0.2	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
ImageNet	Gaussian deblur	DiffPIR+LAMP	100	3	-3.0	8.0	0.3	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	Motion deblur	DiffPIR+LAMP	100	3	-3.0	8.0	0.7	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	SR $\times 4$	DiffPIR+LAMP	100	3	-3.0	9.0	0.5	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
CelebA	Gaussian deblur	DiffPIR+LAMP	100	3	-3.0	50.0	0.0	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	Motion deblur	DiffPIR+LAMP	100	3	-3.0	8.0	0.0	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	SR $\times 4$	DiffPIR+LAMP	100	3	-3.0	9.0	0.0	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
Noisy ($\sigma_y = 0.05$)									
FFHQ	Gaussian deblur	DiffPIR+LAMP	100	3	-3.0	7.0	0.3	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	Motion deblur	DiffPIR+LAMP	100	3	-3.0	7.0	0.4	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	SR $\times 4$	DiffPIR+LAMP	100	3	-3.0	7.0	0.2	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
ImageNet	Gaussian deblur	DiffPIR+LAMP	100	3	-3.0	8.0	0.3	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	Motion deblur	DiffPIR+LAMP	100	3	-3.0	8.0	0.7	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	SR $\times 4$	DiffPIR+LAMP	100	3	-3.0	9.0	0.5	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0

CelebA	Gaussian deblur	DiffPIR+LAMP	100	3	-3.0	50.0	0.0	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	Motion deblur	DiffPIR+LAMP	100	3	-3.0	8.0	0.0	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0
	SR $\times 4$	DiffPIR+LAMP	100	3	-3.0	9.0	0.0	–	–
		DDRM+LAMP	20	3	-0.15	–	–	0.85	1.0

Table 2: Hyperparameters used by LAMP in the noiseless setting ($\sigma_{\boldsymbol{y}} = 0$) and noisy setting ($\sigma_{\boldsymbol{y}} = 0.05$). The parameters μ and ζ are used by DiffPIR, while η and η_B are used by DDRM.

C.3 Neural network architecture

All experiments use a pretrained DDPM denoiser with the `score_sde` UNet architecture. The checkpoint is automatically detected by the model loader and loaded through the `DDPMModel` implementation in `sde_model.py`, rather than through the `guided_diffusion` UNet implementation. The model predicts the diffusion noise $\boldsymbol{\epsilon}_\Theta(\boldsymbol{x}_t, t)$ and is not class-conditional.

Parameter	**Value**
Image size	256×256
Base channels	128
Residual blocks per level	2
Head channels	64
Attention resolutions	16
Channel multipliers	$(1, 1, 2, 2, 4, 4)$
Dropout	0.0

Table 3: UNet architecture used by the pretrained `score_sde` DDPM denoiser.

C.4 Diffusion model and sampling schedule

The denoiser is a Ho et al.-style DDPM model trained to predict only the noise component, rather than both noise and variance. Accordingly, we use

$$\text{learn_sigma} = \texttt{false}, \tag{106}$$

so that the sampling code uses the fixed small variance parameterization instead of the learned-range variance parameterization. The model uses a linear diffusion schedule with 1000 training diffusion steps. At inference time, we use a respaced deterministic reverse schedule with a number of selected timesteps depending on the method.

Parameter	**Value**
Learned variance	No
Number of training diffusion steps	1000
Noise schedule	Linear

Table 4: Diffusion-model configuration used by the pretrained denoiser.

C.5 Datasets and preprocessing

Dataset	Resolution	Number of test images	Crop/resize	Pretrained model
FFHQ	256×256	100	`None`	`diffusion_ffhq_10m.pt` [3]
ImageNet	256×256	100	`None`	`256x256_diffusion_uncond.pt` [3]
CelebA	256×256	100	`None`	`celeba_hq.ckpt` [3]

Table 5: Dataset and preprocessing details.

C.6 Forward operators

We consider the following linear inverse problems.

Task	Operator	Parameters
Gaussian deblurring	Gaussian convolution	`kernel_size` 61×61, `sigma` 3.0
Motion deblurring	Motion convolution	`kernel_size` 61×61, `intensity` 0.5
Super-resolution	Average Downsampling	`downsampling_type` average, `factor` 4

Table 6: Forward operators used in the experiments.

D Additional Quantitative Results

This section presents additional quantitative results that complement the paper. While the paper presents the main comparisons of PSNR, SSIM, and LPIPS mean values for noisy settings, here we consider experiments in noiseless settings, as well as ablations on additional degradation levels, confidence intervals, and per-task breakdowns.

D.1 First- and second-order variants without the residual correction

Table 7 reports the 1M and 2M variants of DiffPIR and DDRM in the noisy setting. These variants omit the residual correction term; the 2M variants include the multistep temporal term. Together with Table 1, the results show that the temporal discretization alone does not reproduce the gains obtained with LAMP.

CelebA		**Deblur (Gaussian)**			**Deblur (Motion)**			**SR** ($\times 4$)		
Method	NFE	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DiffPIR [18]	100	24.84	0.705	0.107	23.20	0.567	0.254	26.90	0.690	0.202
DDRM [6]	20	26.11	0.789	0.169	25.52	0.770	0.188	28.97	0.818	0.131
DiffPIR-1M	100	24.84	0.705	0.107	23.20	0.567	0.254	26.90	0.690	0.202
DiffPIR-2M	100	24.56	0.687	0.114	23.06	0.559	0.258	26.59	0.675	0.215
DDRM-1M	20	15.51	0.146	0.944	14.37	0.108	1.024	27.92	0.794	0.201
DDRM-2M	20	15.61	0.149	0.943	14.51	0.111	1.023	27.76	0.789	0.201
DiffPIR-LAMP	100	25.04	0.716	0.106	23.26	0.566	0.230	27.11	0.699	0.191
DDRM-LAMP	20	26.14	0.791	0.118	25.55	0.772	0.129	29.12	0.823	0.105
FFHQ		**Deblur (Gaussian)**			**Deblur (Motion)**			**SR** ($\times 4$)		
Method	NFE	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DiffPIR [18]	100	24.84	0.741	0.132	23.66	0.696	0.170	26.54	0.676	0.239
DDRM [6]	20	25.51	0.776	0.238	24.48	0.732	0.249	27.94	0.804	0.189
DiffPIR-1M	100	23.71	0.610	0.188	21.23	0.386	0.391	25.01	0.592	0.309
DiffPIR-2M	100	23.35	0.585	0.205	21.00	0.376	0.401	24.65	0.574	0.315
DDRM-1M	20	11.80	0.073	1.203	11.51	0.061	1.141	26.11	0.696	0.273
DDRM-2M	20	11.85	0.074	1.210	11.58	0.062	1.142	25.99	0.693	0.271
DiffPIR-LAMP	100	25.18	0.761	0.160	23.94	0.713	0.202	27.05	0.704	0.242
DDRM-LAMP	20	25.55	0.779	0.164	24.52	0.737	0.181	28.07	0.808	0.146
ImageNet		**Deblur (Gaussian)**			**Deblur (Motion)**			**SR** ($\times 4$)		
Method	NFE	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DiffPIR [18]	100	21.77	0.581	0.337	20.96	0.528	0.460	23.17	0.569	0.481
DDRM [6]	20	22.38	0.611	0.500	21.36	0.544	0.480	23.73	0.642	0.405
DiffPIR-1M	100	19.27	0.361	0.435	13.92	0.106	0.997	20.85	0.439	0.432
DiffPIR-2M	100	18.73	0.340	0.450	13.46	0.100	1.000	20.02	0.392	0.474
DDRM-1M	20	8.53	0.043	1.113	8.71	0.039	1.160	21.86	0.548	0.401
DDRM-2M	20	8.50	0.043	1.115	8.68	0.039	1.163	21.70	0.539	0.405
DiffPIR-LAMP	100	22.01	0.596	0.397	21.01	0.529	0.485	23.43	0.588	0.502

DDRM-LAMP	20	22.41	0.613	0.416	21.38	0.547	0.410	23.80	0.647	0.372

Table 7: Comparison of DiffPIR and DDRM with their 1M, 2M, and LAMP variants for Gaussian deblurring, motion deblurring, and $\times 4$ super-resolution with $\sigma_{\boldsymbol{y}} = 0.05$.

D.2 FID evaluation

We additionally report FID for the original three degradations and all four backbones. Lower is better. From the values below, LAMP reduces FID in 35 of 36 paired comparisons. The exception is DPS on CelebA $\times 4$ super-resolution, where FID changes from 73.39 to 76.26.

	Deblur (Gaussian)			**Deblur (Motion)**			**SR** ($\times 4$)		
Method	CelebA	FFHQ	ImageNet	CelebA	FFHQ	ImageNet	CelebA	FFHQ	ImageNet
DDRM	85.75	102.15	152.99	90.99	116.80	206.97	78.34	100.03	131.85
DiffPIR	68.67	62.35	96.96	93.40	76.93	186.66	82.16	100.68	134.92
DPS	67.15	66.82	122.41	77.06	81.68	195.72	73.39	79.42	174.91
ΠGDM	63.35	72.15	93.66	294.38	363.32	297.39	65.21	67.07	81.31
DDRM-LAMP	62.92	74.11	125.11	64.72	85.43	183.51	68.91	68.81	109.35
DiffPIR-LAMP	67.79	61.87	93.95	87.80	73.25	179.21	80.46	100.43	126.31
DPS-LAMP	63.39	57.74	106.02	67.18	73.77	177.87	76.26	68.51	164.92
ΠGDM-LAMP	60.72	69.95	92.84	277.36	357.53	277.81	62.27	61.84	76.36

Table 8: FID↓ for Gaussian deblurring, motion deblurring, and $\times 4$ super-resolution, computed from 100 reconstructed images for each dataset and degradation at $\sigma_{\boldsymbol{y}} = 0.05$.

D.3 Noiseless box inpainting on FFHQ

We additionally consider noiseless box inpainting on FFHQ, which complements the blur and super-resolution experiments with a spatially localized inverse problem. Table 9 compares the four posterior samplers with their LAMP variants. LAMP improves PSNR and SSIM and reduces LPIPS and FID for every backbone in this setting.

Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
DPS	23.70	0.822	0.123	69.02
DiffPIR	23.29	0.860	0.077	42.38
DDRM	23.98	0.871	0.065	34.95
ΠGDM	23.97	0.870	0.067	34.40
DPS-LAMP	23.87	0.838	0.118	65.12
DiffPIR-LAMP	23.40	0.864	0.069	39.22
DDRM-LAMP	24.39	0.882	0.060	32.92
ΠGDM-LAMP	24.04	0.872	0.064	32.33

Table 9: Noiseless box inpainting on FFHQ.

D.4 Perceptual sensitivity to lag strength

Table 10 reports the sensitivity to the lag strength for noisy motion deblurring on FFHQ. A mild lag ($\gamma = -0.15$) gives the highest PSNR, while a stronger lag ($\gamma = -0.8$) gives the lowest LPIPS at a small cost in PSNR. Larger effective weights substantially degrade both metrics, and the positive value of γ tested here also performs worse than the unmodified sampler ($\gamma = 0$). These results illustrate the trade-off between distortion and perceptual quality when choosing the lag strength.

γ	$\bar{\beta}$	PSNR↑	LPIPS↓
0.5	-0.097	22.776	0.1854
0	0.000	23.125	0.1834
-0.15	0.029	23.149	0.1827
-0.8	0.155	22.841	0.1811
-2.5	0.483	20.438	0.4018
-5	0.967	15.918	0.4457
-10	1.934	7.484	0.5761

Table 10: Sensitivity to the lag strength in noisy motion deblurring on FFHQ. The highest PSNR occurs at $\gamma = -0.15$, whereas the lowest LPIPS occurs at $\gamma = -0.8$.

D.5 Noiseless Case

FFHQ		**Deblur (Gaussian)**		**Deblur (Motion)**		**SR** ($\times 4$)	
Method	NFE	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DPS [1]	1000	**26.32**	0.741	22.69	0.636	24.68	0.689
DiffPIR [18]	100	25.10	0.779	24.51	0.767	28.60	0.837
DDRM [6]	20	25.53	0.796	24.69	0.773	29.08	0.843
DiffPIR-1M	100	24.51	0.699	24.17	0.744	26.50	0.720
DiffPIR-2M	100	24.26	0.684	23.95	0.722	26.09	0.707
DDRM-1M	20	24.87	0.688	24.40	0.777	26.98	0.738
DDRM-2M	20	24.85	0.691	24.33	0.772	26.82	0.732
DiffPIR+LAMP (Ours)	100	25.28	0.790	24.62	0.775	29.01	**0.850**
DDRM+LAMP (Ours)	20	25.59	**0.802**	**24.75**	**0.784**	**29.23**	0.849
ImageNet		**Deblur (Gaussian)**		**Deblur (Motion)**		**SR** ($\times 4$)	
Method	NFE	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DPS [1]	1000	21.66	0.534	17.29	0.388	20.94	0.484
DiffPIR [18]	100	22.18	0.639	21.57	0.614	24.64	0.719
DDRM [6]	20	22.58	0.651	21.82	0.619	24.35	0.692
DiffPIR-1M	100	21.34	0.579	20.78	0.562	21.77	0.562
DiffPIR-2M	100	20.81	0.541	20.50	0.534	20.98	0.517
DDRM-1M	20	22.29	0.651	21.09	0.610	22.42	0.599
DDRM-2M	20	22.23	0.646	21.02	0.604	22.24	0.587
DiffPIR+LAMP (Ours)	100	22.34	0.648	21.62	0.617	**24.80**	**0.726**
DDRM+LAMP (Ours)	20	**22.61**	**0.655**	**21.85**	**0.627**	24.43	0.697
CelebA		**Deblur (Gaussian)**		**Deblur (Motion)**		**SR** ($\times 4$)	
Method	NFE	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DPS [1]	1000	**27.36**	0.750	25.02	0.697	26.43	0.725
DiffPIR [18]	100	25.30	0.745	24.86	0.754	28.56	0.818
DDRM [6]	20	26.06	0.804	25.49	0.795	29.96	0.847
DiffPIR-1M	100	25.30	0.745	24.86	0.754	28.56	0.818
DiffPIR-2M	100	25.07	0.729	24.71	0.740	28.20	0.805
DDRM-1M	20	25.86	0.793	25.05	0.783	28.80	0.825
DDRM-2M	20	25.79	0.788	24.99	0.778	28.58	0.817
DiffPIR+LAMP (Ours)	100	25.47	0.756	24.97	0.763	28.82	0.827
DDRM+LAMP (Ours)	20	26.11	**0.808**	**25.53**	**0.800**	**30.15**	**0.854**

Table 11: Quantitative comparison in the noiseless setting. Best values are in bold and second-best values are underlined.

D.6 Robustness to Increasing Measurement Noise

In this section, we evaluate the robustness of the considered methods under increasing measurement noise. We report results on FFHQ for Gaussian deblurring and motion deblurring, varying the noise standard deviation $\sigma \in \{0.01, 0.03, 0.05, 0.07, 0.10\}$. This analysis complements the main results by

showing how reconstruction quality degrades as the inverse problem becomes progressively more corrupted by noise.

FFHQ	Deblur (Gaussian)									
Method	$\sigma = 0.01$		$\sigma = 0.03$		$\sigma = 0.05$		$\sigma = 0.07$		$\sigma = 0.10$	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DiffPIR	25.08	0.774	24.94	0.751	24.79	0.737	24.64	0.725	24.44	0.712
DDRM	<u>25.52</u>	<u>0.799</u>	<u>25.54</u>	<u>0.785</u>	<u>25.51</u>	<u>0.776</u>	<u>25.45</u>	<u>0.768</u>	<u>25.33</u>	<u>0.758</u>
DiffPIR-1M	24.36	0.680	24.04	0.644	23.79	0.623	23.59	0.607	23.32	0.588
DiffPIR-2M	24.05	0.658	23.68	0.620	23.42	0.598	23.20	0.582	22.93	0.564
DDRM-1M	12.71	0.092	12.09	0.079	11.80	0.073	11.60	0.069	11.33	0.063
DDRM-2M	12.78	0.093	12.14	0.080	11.85	0.074	11.65	0.069	11.38	0.064
DiffPIR-LAMP (Ours)	25.26	0.786	25.15	0.764	25.01	0.750	24.87	0.739	24.67	0.726
DDRM-LAMP (Ours)	**25.57**	**0.802**	**25.58**	**0.789**	**25.55**	**0.779**	**25.49**	**0.771**	**25.38**	**0.761**
FFHQ	**Deblur (Motion)**									
Method	$\sigma = 0.01$		$\sigma = 0.03$		$\sigma = 0.05$		$\sigma = 0.07$		$\sigma = 0.10$	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DiffPIR	24.47	0.758	24.00	0.714	23.50	0.688	23.09	0.670	22.59	0.650
DDRM	<u>24.69</u>	<u>0.779</u>	<u>24.69</u>	<u>0.750</u>	<u>24.48</u>	<u>0.732</u>	<u>24.25</u>	<u>0.719</u>	<u>23.93</u>	<u>0.704</u>
DiffPIR-1M	23.10	0.549	22.08	0.452	21.54	0.426	21.14	0.414	20.69	0.403
DiffPIR-2M	22.88	0.534	21.83	0.438	21.27	0.412	20.87	0.399	20.41	0.387
DDRM-1M	19.08	0.256	13.38	0.094	11.51	0.061	10.50	0.046	9.70	0.036
DDRM-2M	19.21	0.262	13.47	0.095	11.58	0.062	10.57	0.047	9.76	0.036
DiffPIR-LAMP (Ours)	24.59	0.768	24.16	0.725	23.68	0.700	23.27	0.682	22.77	0.663
DDRM-LAMP (Ours)	**24.72**	**0.785**	**24.73**	**0.756**	**24.52**	**0.737**	**24.29**	**0.724**	**23.97**	**0.708**

Table 12: Robustness to increasing measurement noise on FFHQ. We report PSNR and SSIM for Gaussian and motion deblurring under different noise levels σ. Best values are in bold and second-best values are underlined.

D.7 Robustness Across Multiple Runs

To assess the robustness of the evaluated methods, we repeat each experiment across multiple independent runs and analyze the variability of the results. In particular, for each run we compute the average PSNR and SSIM over the dataset, and we then report the standard deviation of these averages across runs. This provides a measure of stability with respect to stochasticity in the sampling process and implementation details. The results in Table 13 complement the main quantitative comparison by highlighting not only performance but also consistency across repeated executions.

CelebA		Deblur (Gaussian)		Deblur (Motion)		SR ($\times 4$)	
Method	NFE	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DPS [1]	1000	26.45 ± 0.036	0.720 ± 0.0012	23.74 ± 0.055	0.659 ± 0.0007	25.88 ± 0.018	0.710 ± 0.0009
DiffPIR [18]	100	24.84 ± 0.044	0.705 ± 0.0013	23.20 ± 0.045	0.567 ± 0.0025	26.90 ± 0.024	0.690 ± 0.0009
DDRM [6]	20	26.11 ± 0.018	<u>0.789</u> ± 0.0006	<u>25.52</u> ± 0.021	<u>0.770</u> ± 0.0005	<u>28.97</u> ± 0.025	<u>0.818</u> ± 0.0005
ΠGDM	100	<u>27.86</u> ± 0.041	0.773 ± 0.0011	10.66 ± 0.063	0.260 ± 0.0023	28.29 ± 0.029	0.793 ± 0.0008
DiffPIR-1M	100	24.84 ± 0.044	0.705 ± 0.0013	23.20 ± 0.045	0.567 ± 0.0025	26.90 ± 0.024	0.690 ± 0.0009
DiffPIR-2M	100	24.56 ± 0.046	0.687 ± 0.0012	23.06 ± 0.046	0.559 ± 0.0027	26.59 ± 0.024	0.675 ± 0.0009
DDRM-1M	20	15.51 ± 0.017	0.146 ± 0.0003	14.37 ± 0.009	0.108 ± 0.0003	27.92 ± 0.022	0.794 ± 0.0008
DDRM-2M	20	15.61 ± 0.017	0.149 ± 0.0003	14.51 ± 0.009	0.111 ± 0.0003	27.76 ± 0.022	0.789 ± 0.0009
DiffPIR-LAMP (Ours)	100	25.04 ± 0.043	0.716 ± 0.0013	23.26 ± 0.042	0.566 ± 0.0022	27.11 ± 0.024	0.699 ± 0.0009
DDRM-LAMP (Ours)	20	26.14 ± 0.018	**0.791** ± 0.0006	**25.55** ± 0.021	**0.772** ± 0.0005	**29.12** ± 0.026	**0.823** ± 0.0005
DPS-LAMP (Ours)	1000	26.65 ± 0.034	0.725 ± 0.0010	23.91 ± 0.052	0.664 ± 0.0008	26.09 ± 0.020	0.715 ± 0.0010
ΠGDM-LAMP (Ours)	100	**28.36** ± 0.038	0.787 ± 0.0010	11.53 ± 0.061	0.334 ± 0.0022	28.74 ± 0.027	0.805 ± 0.0007

FFHQ		Deblur (Gaussian)		Deblur (Motion)		SR ($\times 4$)	
Method	NFE	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DPS [1]	1000	25.35 ± 0.046	0.706 ± 0.0019	21.58 ± 0.077	0.595 ± 0.0035	24.27 ± 0.032	0.676 ± 0.0010
DiffPIR [18]	100	24.84 ± 0.026	0.741 ± 0.0009	23.66 ± 0.019	0.696 ± 0.0010	26.54 ± 0.014	0.676 ± 0.0005
DDRM [6]	20	25.51 ± 0.024	<u>0.776</u> ± 0.0006	<u>24.48</u> ± 0.010	<u>0.732</u> ± 0.0005	<u>27.94</u> ± 0.029	<u>0.804</u> ± 0.0005
ΠGDM	100	25.45 ± 0.031	0.729 ± 0.0012	8.30 ± 0.049	0.155 ± 0.0027	27.27 ± 0.025	0.772 ± 0.0009
DiffPIR-1M	100	23.71 ± 0.049	0.610 ± 0.0056	21.23 ± 0.028	0.386 ± 0.0015	25.01 ± 0.094	0.592 ± 0.0047
DiffPIR-2M	100	23.35 ± 0.058	0.585 ± 0.0058	21.00 ± 0.035	0.376 ± 0.0018	24.65 ± 0.091	0.574 ± 0.0044
DDRM-1M	20	11.80 ± 0.011	0.073 ± 0.0001	11.51 ± 0.005	0.061 ± 0.0002	26.11 ± 0.103	0.696 ± 0.0064
DDRM-2M	20	11.85 ± 0.012	0.074 ± 0.0001	11.58 ± 0.005	0.062 ± 0.0002	25.99 ± 0.100	0.693 ± 0.0064
DiffPIR-LAMP (Ours)	100	25.18 ± 0.023	0.761 ± 0.0007	23.94 ± 0.016	0.713 ± 0.0008	27.05 ± 0.010	0.704 ± 0.0004
DDRM-LAMP (Ours)	20	25.55 ± 0.023	**0.779** ± 0.0005	**24.52** ± 0.009	**0.737** ± 0.0004	**28.07** ± 0.028	**0.808** ± 0.0005
DPS-LAMP (Ours)	1000	<u>25.98</u> ± 0.042	0.732 ± 0.0017	22.06 ± 0.071	0.608 ± 0.0032	24.70 ± 0.030	0.694 ± 0.0008
ΠGDM-LAMP (Ours)	100	**26.42** ± 0.028	0.741 ± 0.0010	8.68 ± 0.045	0.179 ± 0.0025	27.39 ± 0.024	0.781 ± 0.0008

ImageNet		Deblur (Gaussian)		Deblur (Motion)		SR ($\times 4$)	
Method	NFE	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DPS [1]	1000	21.02 ± 0.198	0.495 ± 0.0063	16.97 ± 0.110	0.367 ± 0.0069	20.76 ± 0.074	0.477 ± 0.0047
DiffPIR [18]	100	21.77 ± 0.044	0.581 ± 0.0019	20.96 ± 0.024	0.528 ± 0.0015	23.17 ± 0.027	0.569 ± 0.0007
DDRM [6]	20	22.38 ± 0.006	<u>0.611</u> ± 0.0004	<u>21.36</u> ± 0.013	<u>0.544</u> ± 0.0002	<u>23.73</u> ± 0.042	<u>0.642</u> ± 0.0014
ΠGDM	100	22.02 ± 0.055	0.574 ± 0.0020	6.11 ± 0.041	0.027 ± 0.0014	22.92 ± 0.038	0.628 ± 0.0011
DiffPIR-1M	100	19.27 ± 0.089	0.361 ± 0.0047	13.92 ± 0.018	0.106 ± 0.0014	20.85 ± 0.109	0.439 ± 0.0034
DiffPIR-2M	100	18.73 ± 0.099	0.340 ± 0.0045	13.46 ± 0.022	0.100 ± 0.0016	20.02 ± 0.102	0.392 ± 0.0024
DDRM-1M	20	8.53 ± 0.006	0.043 ± 0.0002	8.71 ± 0.005	0.039 ± 0.0002	21.86 ± 0.047	0.548 ± 0.0038
DDRM-2M	20	8.50 ± 0.006	0.043 ± 0.0002	8.68 ± 0.005	0.039 ± 0.0002	21.70 ± 0.050	0.539 ± 0.0038
DiffPIR-LAMP (Ours)	100	22.01 ± 0.030	0.596 ± 0.0010	21.01 ± 0.023	0.529 ± 0.0014	23.43 ± 0.024	0.588 ± 0.0005
DDRM-LAMP (Ours)	20	<u>22.41</u> ± 0.005	**0.613** ± 0.0004	**21.38** ± 0.012	**0.547** ± 0.0002	**23.80** ± 0.042	**0.647** ± 0.0014
DPS-LAMP (Ours)	1000	21.19 ± 0.183	0.515 ± 0.0058	18.07 ± 0.102	0.390 ± 0.0065	21.59 ± 0.069	0.495 ± 0.0042
ΠGDM-LAMP (Ours)	100	**22.61** ± 0.051	0.581 ± 0.0018	6.58 ± 0.039	0.045 ± 0.0012	23.08 ± 0.036	0.633 ± 0.0010

Table 13: Robustness evaluation in the noisy setting ($\sigma_{\boldsymbol{y}} = 0.05$) across multiple runs. For each method and task, we report the average PSNR and SSIM, together with the standard deviation computed over $R = 5$ runs with different seeds, indicated in gray. Best values are in bold and second-best values are underlined.

E Additional Qualitative Results

This section contains additional visual comparisons.

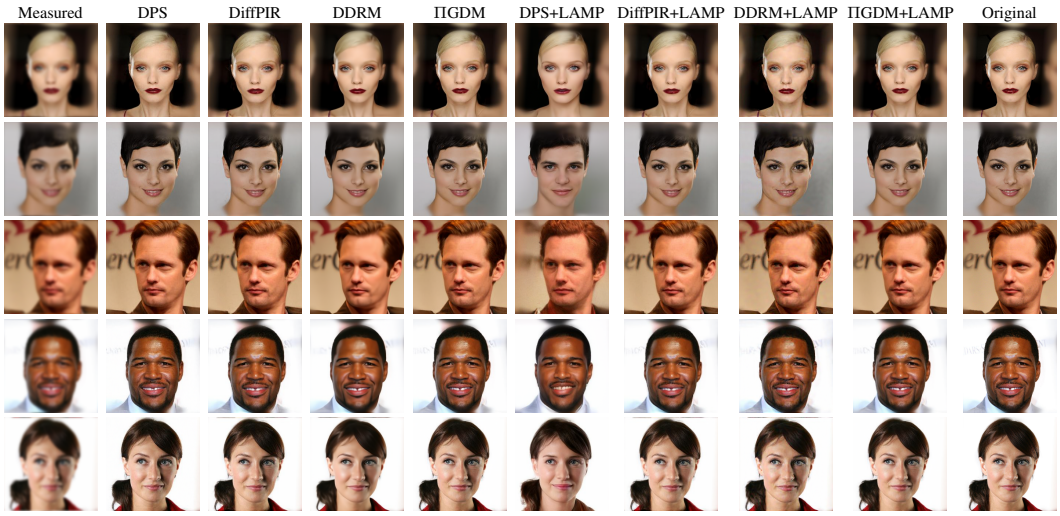

Figure 4: Qualitative comparison on CelebA – Gaussian deblurring (noiseless).

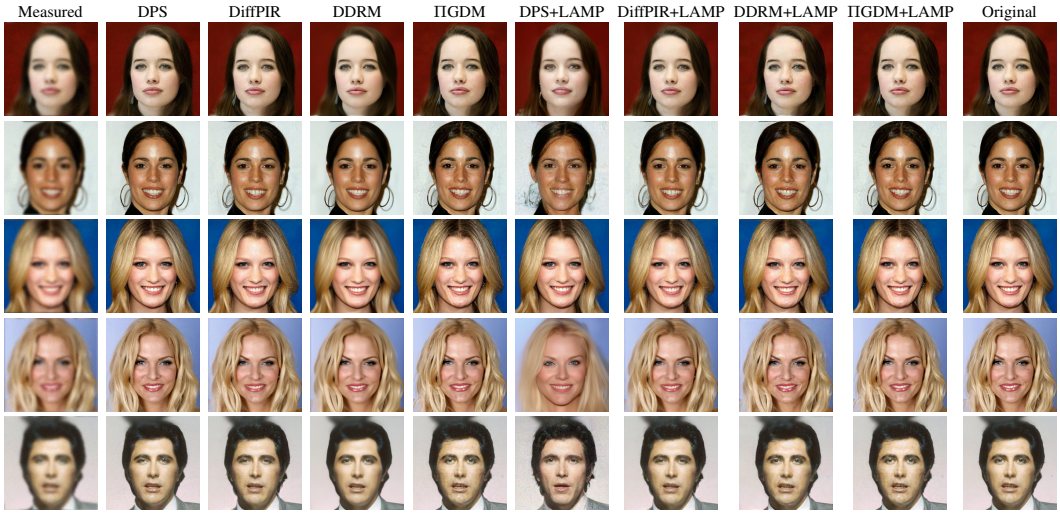

Figure 5: Qualitative comparison on CelebA – Gaussian deblurring (noisy).

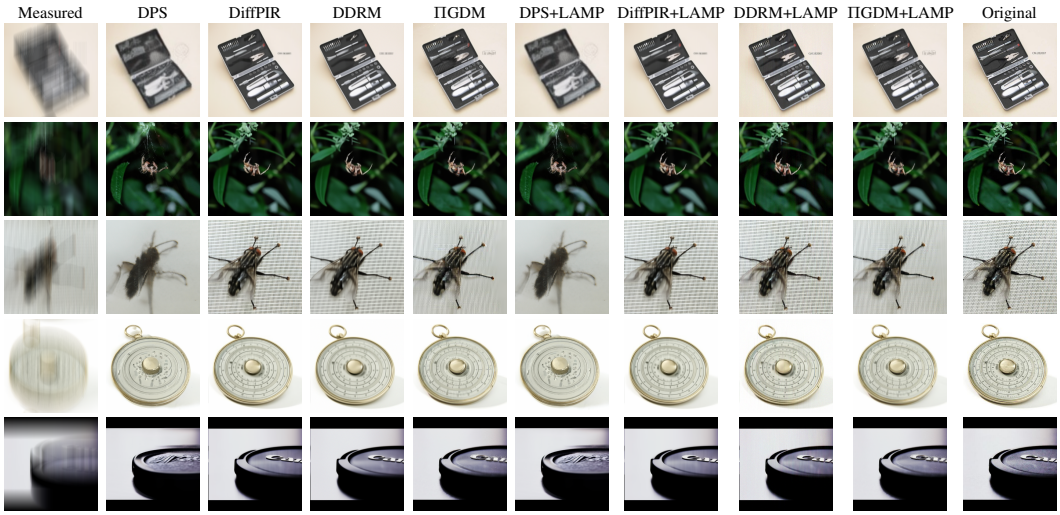

Figure 6: Qualitative comparison on ImageNet – Motion deblurring (noiseless).

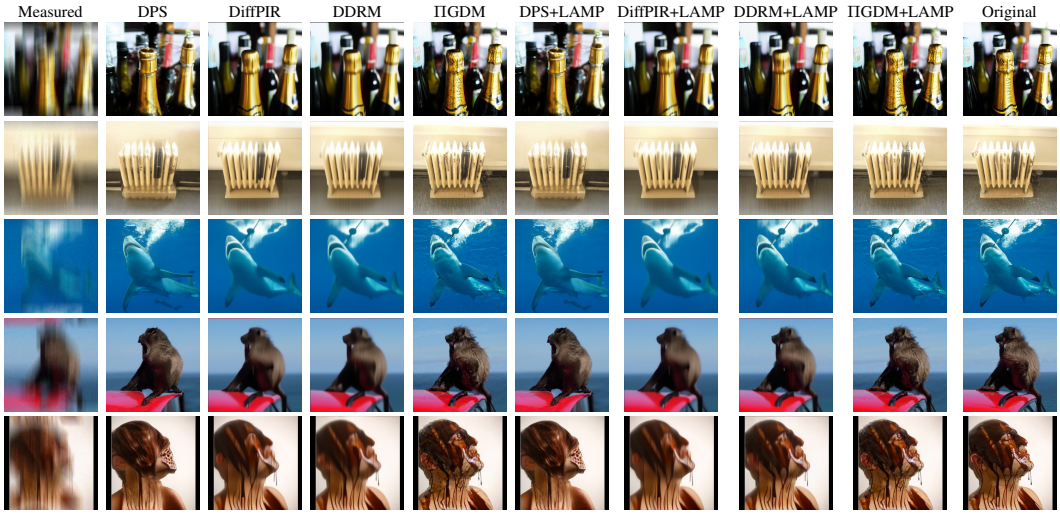

Figure 7: Qualitative comparison on ImageNet – Motion deblurring (noisy).

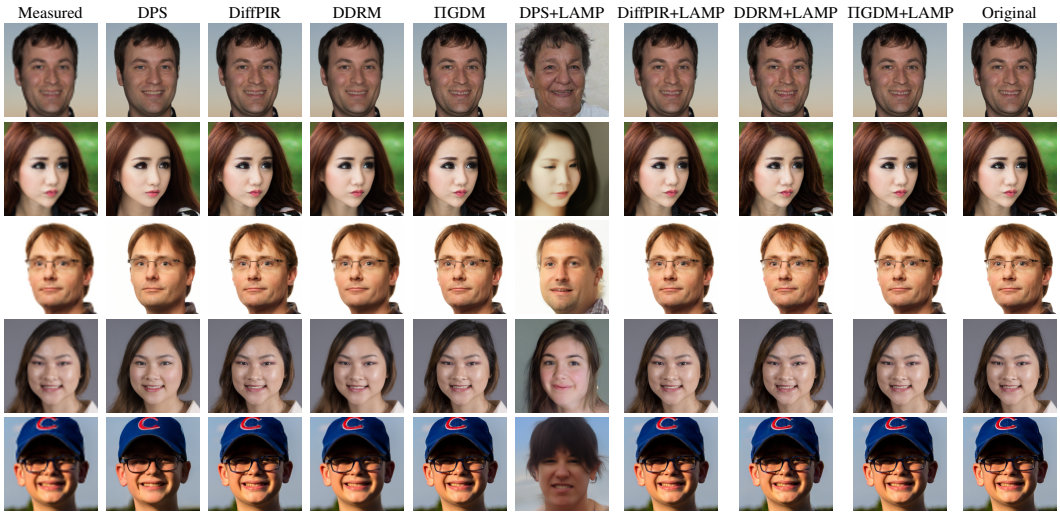

Figure 8: Qualitative comparison on FFHQ – $\times 4$ super-resolution (noiseless).



Figure 9: Qualitative comparison on FFHQ – $\times 4$ super-resolution (noisy).

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions of the paper. These claims are directly supported by the methodological derivations (in Section 3 as well as in Appendices A and B) and by the experimental results across multiple inverse problems (Section 4, Appendices D, D.7 and E).

   Guidelines:

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

2. **Limitations**

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

Answer: [Yes]

Justification: We have included the dedicated Section 5.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The main paper clearly states all the assumptions required to prove the theoretical results. Since the proofs are mostly algebraic steps (that are not suitable for short sketches), we only included the complete proofs in Appendix B. We have specified in the main paper (in Section 3) that all the proofs are available in the Appendix.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes].

Justification: The paper provides sufficient details to ensure reproducibility. The main text describes the methodological formulation and experimental setup, while Appendix C includes complete implementation details, including pseudocode, algorithmic hyperparameters, architectural specifications, and dataset descriptions with preprocessing steps. Experiments are conducted on popular datasets and using standard benchmarks and publicly available pretrained backbones (e.g., DiffPIR and DDRM), enabling replication of the reported results.

Guidelines:

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

5. **Open access to data and code**

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

Answer: [Yes]

Justification: The code for replicating the experiments presented in this paper is fully open and available at the URL indicated in Section 4. The data and model checkpoints are standard and openly available. A guide on how to download the pre-trained models and the test data is given in the README.md of the project repository.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: The paper provides a complete description of the experimental setup. The main text specifies the evaluation protocols, datasets, and number of denoising evaluations, while Appendix C includes detailed information on hyperparameters, algorithmic settings, architectural choices, and dataset preprocessing. Additional information is contained in the official repository associated with this paper, whose link is given in Section 4.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The main paper report, for each experiment, the average PSNR, SSIM, and LPIPS, averaged over the whole test set of 100 images. Each experiment is repeated 5 times with a different seed to ensure that the results reflect consistent performance rather than single-run variability. Due to space limitations, in the main paper, we only reported the mean of each metric over the 5 runs, while we included the standard deviations in Appendix D.7, Table 13.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We included a full description of the computer resources used to reproduce the experiments in Section 4.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: This work complies with the NeurIPS Code of Ethics. The experiments are conducted on standard, publicly available datasets, and do not involve sensitive data or human subjects. No ethical risks or misuse concerns are identified. The submission preserves anonymity in accordance with the double-blind review process.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: The paper presents a methodological contribution for diffusion-based image restoration and does not introduce new application domains or capabilities with direct societal impact. It operates within standard benchmark settings and does not raise specific, positive or negative, societal implications beyond those already associated with existing methods.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The present paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: In our experiments, we only used assets (in particular, datasets and pre-trained models) that are publicly available. We have properly credited them in Table 5, Appendix C.5.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not introduce new datasets, models, or other assets. It proposes a novel algorithm for image reconstruction and relies exclusively on existing datasets and pre-trained methods from the literature. The code implementing the method, together with a link to download the assets used for the experiments, is given in the project repository at the URL indicated in Section 4.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

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

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

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

    Guidelines:

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

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

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: This paper does not use LLMs in a way that impact methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.