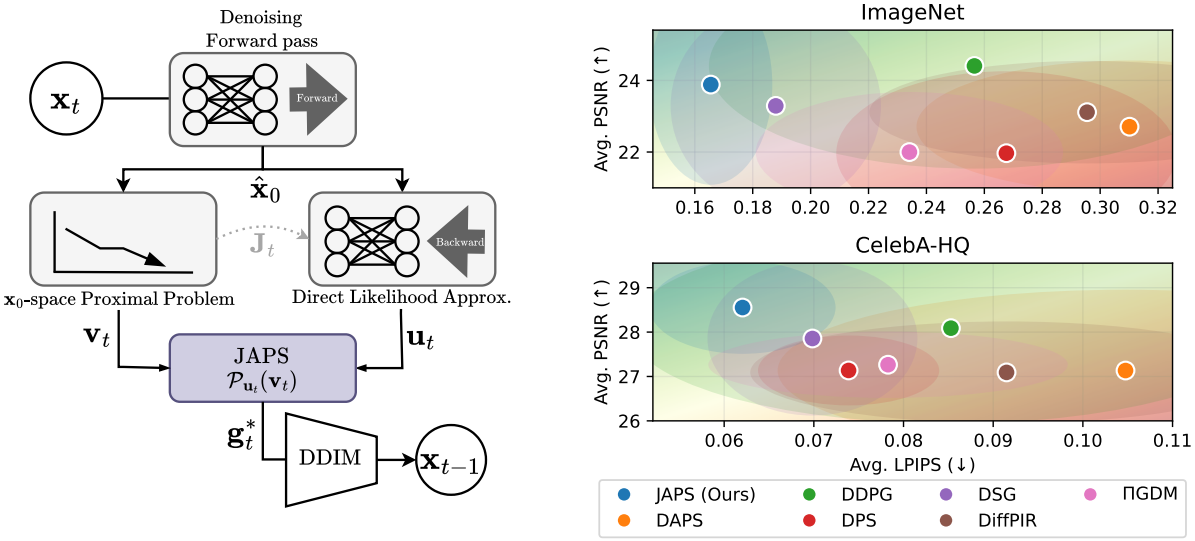
Jacobian-aware Posterior Sampling for Inverse Problems

Anonymous authors
Paper under double-blind review

Abstract

Diffusion models provide powerful generative priors for solving inverse problems by sampling from a posterior distribution conditioned on corrupted measurements. Existing methods primarily follow two paradigms: direct methods, which approximate the likelihood term, and proximal methods, which incorporate intermediate solutions satisfying measurement constraints into the sampling process. We demonstrate that these approaches differ fundamentally in their treatment of the diffusion denoiser's Jacobian within the likelihood term. While this Jacobian encodes critical prior knowledge of the data distribution, training-induced non-idealities can degrade performance in zero-shot settings. In this work, we bridge direct and proximal approaches by proposing a principled **J**acobian-**A**ware **P**osterior **S**ampler (JAPS). JAPS leverages the Jacobian's prior knowledge while mitigating its detrimental effects through a corresponding proximal solution, requiring no additional computational cost. Our method enhances reconstruction quality across diverse linear and nonlinear noisy imaging tasks, outperforming existing diffusion-based baselines in perceptual quality while maintaining or improving distortion metrics.



(a) **Schematic overview of our approach.** We bridge between direct and proximal methods, which essentially differ by the presence of the denoiser's Jacobian, leveraging both to formulate an enhanced likelihood surrogate.

(b) **Summary of experimental results.** Avg. PSNR/LPIPS across various reconstruction tasks—SR×4, Gaussian and motion deblurring, and box/random inpainting—with shaded ellipsoids for standard deviation.

Figure 1: **Proposed approach and experimental summary. (a)** We introduce an enhanced likelihood surrogate by bridging direct and proximal methods via the denoiser's Jacobian. **(b)** Performance comparison across various inverse problems, demonstrating the efficacy of our approach.

1 Introduction

Image restoration aims to recover a high-quality image $\mathbf{x} \in \mathbb{R}^n$ from a degraded observation $\mathbf{y} = \mathcal{A}(\mathbf{x}) + \boldsymbol{\varepsilon}$, where $\mathcal{A}$ is a measurement operator and $\boldsymbol{\varepsilon}$ is additive noise. Common restoration tasks include denoising, deblurring, super-resolution and inpainting, which can also benefit high-level downstream tasks (Turgeman and Tirer, 2026). Since these inverse problems are typically ill-posed, incorporating strong structural priors is essential for accurate recovery.

While task-specific deep neural networks (DNNs) offer high performance, they often suffer from performance degradation when test-time observations deviate from training assumptions (Hussein et al., 2020; Shocher et al., 2018; Tirer and Giryes, 2019). To address this, zero-shot frameworks such as Plug-and-Play (PnP) and Regularization by Denoising (RED) (Romano et al., 2017; Tirer and Giryes, 2018; Venkatakrishnan et al., 2013; Zhang et al., 2017) leverage pretrained denoisers as priors. More recently, diffusion models (Ho et al., 2020; Song and Ermon, 2019; Song et al., 2020b) have emerged as powerful generative priors, enabling reconstructions that are both perceptually high-fidelity and consistent with measurements (Abu-Hussein et al., 2022; Chung et al., 2022; 2023; Kawar et al., 2022; Mardani et al., 2023; Song et al., 2023; 2021; Wang et al., 2023; Zhu et al., 2023; Garber and Tirer, 2024; 2025).

Existing diffusion-based posterior samplers primarily follow two paradigms: **direct methods**, which approximate the log-likelihood gradient $\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x})$ along the sampling process, and **proximal methods**, which incorporate intermediate solutions satisfying measurement constraints (Kawar et al., 2022; Chung et al., 2023; Zhu et al., 2023; Garber and Tirer, 2024; Yang et al., 2024). We show that these paradigms differ fundamentally in their treatment of the diffusion denoiser's Jacobian within the likelihood term. While this Jacobian encodes critical local geometry of the data distribution, training-induced non-idealities often lead to artifacts when used directly in zero-shot settings.

In this work, we propose **J**acobian-**A**ware **P**osterior **S**ampling (JAPS). JAPS utilizes the Jacobian's prior information while mitigating detrimental effects through a corresponding proximal estimate, without additional computational cost. Furthermore, we extend this approach to nonlinear problems and provide a mathematical framework for incorporating likelihood terms into DDIM sampling (Song et al., 2020a).

Our main contributions are summarized as follows:

(1) Jacobian-Aware Posterior Sampling: We identify the role of the Jacobian in posterior sampling and introduce JAPS, a sampler that bridges direct and proximal solutions to improve reconstruction quality without extra computational overhead.

(2) Nonlinear extension: We extend our framework to nonlinear inverse problems, demonstrating improved performance across representative nonlinear tasks compared to existing baselines.

(3) DDIM reformulation for conditional guidance: We provide a formulation to incorporate the log-likelihood gradient into the DDIM sampler via a conditional noise estimator. This approach preserves DDIM's original scheduling and scales with time re-spacing and varying levels of stochasticity.

2 Background

2.1 Inverse Problems

We consider the general inverse problem of recovering a high-quality image $\mathbf{x} \in \mathbb{R}^n$ from a degraded observation $\mathbf{y} = \mathcal{A}(\mathbf{x}) + \boldsymbol{\varepsilon}$, where $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^m$ is a measurement operator and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2 \mathbf{I}_m)$ is additive white Gaussian noise. From a Bayesian perspective, $\mathbf{x}$ is a random vector drawn from a prior $p(\mathbf{x})$. While maximizing the likelihood $p(\mathbf{y}|\mathbf{x}) \propto \exp(-\frac{1}{2\sigma_y^2}\|\mathcal{A}(\mathbf{x}) - \mathbf{y}\|_2^2)$ is often insufficient, maximizing the posterior $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$ incorporates structural priors that improve the reconstruction. A central challenge lies in utilizing an accurate prior $p(\mathbf{x})$ that captures the true data distribution.

2.2 Diffusion Models

Diffusion models synthesize data by reversing a gradual noising process (Ho et al., 2020; Song et al., 2020b). A forward SDE progressively corrupts clean data until, at terminal time T, the distribution is approximately Gaussian; generation then amounts to solving the corresponding reverse-time SDE, which requires the *score function* $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ (Anderson, 1982):

$$\mathrm{d}\mathbf{x} = \Big[f(\mathbf{x}, t) - g(t)^2 \, \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\Big] \mathrm{d}t + g(t) \, \mathrm{d}\bar{\mathbf{w}}. \tag{1}$$

In the variance-preserving (VP) formulation, the noisy state is $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\bar{\alpha}_t$ being the noise schedule. A neural network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ is trained to predict $\boldsymbol{\epsilon}$, providing a score estimate:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \approx -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t). \tag{2}$$

Via Tweedie's formula (Efron, 2011), this network yields an estimate of the clean image:

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}} \;\approx\; \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]. \tag{3}$$

For practical sampling, we employ the DDIM algorithm (Song et al., 2020a) for any $t \in [T, T-1, ...1]$:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0(\mathbf{x}_t, t) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + \sigma_t \boldsymbol{\epsilon}_t, \tag{4}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and σ_t determines the level of stochasticity.

2.3 Posterior Sampling

Zero-shot diffusion-based solvers for inverse problems approximate the posterior score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$, which decomposes via Bayes' rule as

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}_{\text{prior score}} + \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)}_{\text{likelihood score}}. \tag{5}$$

The prior score is provided by the pretrained model via Eq. 2. Since only the likelihood $p(\mathbf{y}|\mathbf{x}_0)$ is given through the observation model, the central difficulty lies in approximating the intractable likelihood score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$. Following Peng et al. (2024), existing methods generally fall into two categories.

Direct likelihood approximation. These methods directly approximate the likelihood score by marginalizing over a tractable surrogate for the denoising posterior $p(\mathbf{x}_0|\mathbf{x}_t)$. DPS (Chung et al., 2023) replaces $p(\mathbf{x}_0|\mathbf{x}_t)$ with a Dirac delta $\delta(\mathbf{x}_0 - \hat{\mathbf{x}}_0)$; ΠGDM (Song et al., 2023) uses an isotropic Gaussian $\mathcal{N}(\mathbf{x}_0\,; \hat{\mathbf{x}}_0, r_t^2\mathbf{I})$, while TMPD (Boys et al., 2023) relaxes this assumption to a diagonal Gaussian. DSG (Yang et al., 2024) generalizes this perspective by framing the likelihood score through a spherical Gaussian surrogate, broadening the class of tractable guidance functions beyond the standard isotropic approximation.

Proximal estimation. These methods avoid direct likelihood approximation, and rather use the unconditional Tweedie estimate $\hat{\mathbf{x}}_0$ to construct a proximal problem in $\mathbf{x}_0$-space and compute an approximation to the conditional posterior mean $\mathbf{x}_0^* \approx \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}]$ by solving the optimization problem $\mathbf{x}_0^* = \arg\min_{\mathbf{x}_0} \big(\mathcal{L}(\mathbf{x}_0) + \lambda\mathcal{R}(\mathbf{x}_0)\big)$, with $\mathcal{L}$ being some data fidelity objective, $\mathcal{R}$ some regularization term, and λ a weighting scalar. These problems can also be formulated within a Bayesian Maximum A-Posteriori (MAP) estimation approach, such that the proximal problem takes the form:

$$\mathbf{x}_0^* = \arg\max_{\mathbf{x}_0} \log p(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}) = \arg\max_{\mathbf{x}_0} \big(\log p(\mathbf{x}_0 \mid \mathbf{x}_t) + \log p(\mathbf{y} \mid \mathbf{x}_0)\big). \tag{6}$$

While $\mathbf{x}_0^*$ is the MAP estimator, in linear settings where $\mathcal{A}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ and under specific Gaussian assumptions, it coincides with the conditional mean $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}]$ (proof in Section A.4). For nonlinear operators, it serves

as a tractable approximation of the mean, which is then substituted into the sampling path. Methods in this category include DDNM (Wang et al., 2023), DiffPIR (Zhu et al., 2023) (as interpreted by Peng et al. (2024)), DDPG (Garber and Tirer, 2024), and DAPS (Zhang et al., 2025), which samples from the distribution $p(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y})$ using Markov Chain Monte Carlo algorithms rather than estimating its mean.

Comparison. Direct methods offer a principled derivation but are computationally expensive, as they require backpropagation through the denoiser $\hat{\mathbf{x}}_0(\mathbf{x}_t, t)$ to compute gradients with respect to $\mathbf{x}_t$. Furthermore, as we discuss in this work, the Jacobian of a trained denoiser inherently reflects training-induced non-idealities, which can impact performance when used directly in zero-shot settings. In contrast, Proximal estimation approaches bypass this need but require heuristics to perform on par with direct methods that leverage the knowledge encoded in the learned denoiser. Both approaches are similar in their requirement of an assumption on the prior distribution of $\mathbf{x}_0$, via the regularization term $\mathcal{R}$ or via $p(\mathbf{x}_0 \mid \mathbf{x}_t)$.

3 Jacobian-aware Posterior Sampling

This section presents the theoretical foundation of **J**acobian-**A**ware **P**osterior **S**ampling (JAPS). We bridge direct and proximal likelihood solvers (3.1), analyze the non-idealities of trained Jacobians 3.2, and propose a projection-based guidance mechanism (3.3). Finally, we reformulate DDIM for conditional sampling (3.4) and extend JAPS to nonlinear operators (3.5).

3.1 Bridging Direct and Proximal Likelihood Solvers

We begin by stating the following proposition:

Proposition 3.1. *The likelihood score admits the representation*

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\Big(\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}] - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]\Big). \tag{7}$$

See the proof in Section A.1. This fundamental result suggests that when treating any proximal estimator from Eq. 6 as an approximation of $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}]$, it can be used to form a likelihood approximated surrogate as in the direct approaches. However, as proximal problems are solved in $\mathbf{x}_0$-space, the solution will not include the Jacobian $\mathbf{J}_t := (\partial\hat{\mathbf{x}}_0/\partial\mathbf{x}_t)^\top$ which maps the $\mathbf{x}_0$-space solution to the noisy intermediate state $\mathbf{x}_t$. In this work, we are interested in exploring the impact of $\mathbf{J}_t$ on posterior samplers.

Next, we formulate the settings to build a coupled pair of approximations, a direct and a proximal surrogate, that only differ by the presence of the Jacobian. Under the Gaussian isotropic assumption $p(\mathbf{x}_0 \mid \mathbf{x}_t) \sim \mathcal{N}(\mathbf{x}_0\,; \hat{\mathbf{x}}_0, r_t^2\mathbf{I})$, the proximal problem is quadratic thus the minimum can be approximately achieved via a single-step Gauss-Newton descent. With the further assumption that the measurement operator $\mathbf{A}$ is linear, it is guaranteed that $\arg\max_{\mathbf{x}_0} \log p(\mathbf{x}_0 \mid \mathbf{x}_t\,, \mathbf{y}) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}]$ (see Section A.4 for a proof). We can thus form the likelihood surrogate, which we denote by $\mathbf{v}_t$, using Proposition 3.1, which is similar to a regularized back-projection guidance (Garber and Tirer, 2024; Tirer and Giryes, 2018). In the direct approach, applying the same assumptions yields the solution presented in ΠGDM (Song et al., 2023), which we denote by $\mathbf{u}_t$. Specifically,

$$\mathbf{v}_t := \tfrac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \tfrac{\sigma_y^2}{r_t^2}\mathbf{I})^{-1}(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0), \tag{8}$$

$$\mathbf{u}_t := \tfrac{1}{r_t^2}\mathbf{J}_t\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \tfrac{\sigma_y^2}{r_t^2}\mathbf{I})^{-1}(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0). \tag{9}$$

For detailed derivations see Sections A.2 and A.3.

In these settings, we get that $\mathbf{u}_t = \frac{1-\alpha_t}{\sqrt{\bar{\alpha}_t}}\frac{1}{r_t^2}\mathbf{J}_t\mathbf{v}_t$. In order to tightly couple these direct and proximal approximations, we choose $r_t^2 = \frac{1-\alpha_t}{\sqrt{\bar{\alpha}_t}}$ and obtain the pair $\mathbf{u}_t$, $\mathbf{v}_t$, such that $\mathbf{u}_t = \mathbf{J}_t\mathbf{v}_t$. Under this parameterization, we obtain a direct and a proximal likelihood surrogate that differ only by the presence of the Jacobian, both derived using the same set of assumptions. We discuss the choice of r_t^2 in Section B.1.

Remark 3.2. For many linear operators $\mathbf{A}$, the operator $\left(\mathbf{A}\mathbf{A}^\top + \frac{\sigma_y^2}{r_t^2}\mathbf{I}\right)^{-1}$ can be implemented efficiently: exactly via SVD (for small dense $\mathbf{A}$), via FFT diagonalization for circulant models (e.g., in super-resolution and deblurring), or via Conjugate Gradients in the general case.

Probabilistic Interpretation. The Jacobian $\mathbf{J}_t$ of the ideal MMSE denoiser $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$ is related to the posterior covariance $\mathrm{Cov}[\mathbf{x}_0 \mid \mathbf{x}_t]$ via the following relation: $\mathrm{Cov}[\mathbf{x}_0 \mid \mathbf{x}_t] = \frac{1-\bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}}\mathbf{J}_t$ (proof in Section A.5) (Bishop, 2006; Efron, 2011). Since proximal estimation approaches treat $\hat{\mathbf{x}}_0$ as a fixed-point, the Jacobian attachment of direct likelihood approximations can be viewed as *preconditioning* the proximal gradients according to the uncertainty curvature applied via the Jacobian, up to the scheduled constants.

3.2 Empirical Analysis of Jacobian Deviations

In practice, $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$ is estimated using a neural network (NN) trained on a finite set of samples. Furthermore, these NNs are usually trained for denoising score matching (Meng et al., 2021; Song and Ermon, 2019), such that the Jacobian – a higher-order quantity – is not necessarily matched. These inevitably cause deviations of $\mathbf{J}_t$ from the ideal Jacobian, which we find harmful for posterior sampling.

To further demonstrate the non-ideality of trained Jacobians, we construct a synthetic dataset of a Gaussian Mixture Model (GMM), such that the ground truth distribution is known, and we can analytically compute the prior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ and the likelihood score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t)$, as well as the Jacobian of the MMSE denoiser and any of its properties. We also train a toy diffusion model to have an estimator for $\hat{\mathbf{x}}_0$, and show the difference between its Jacobian properties and those of an analytically derived Jacobian. Since $\mathbf{J}_t \propto \mathrm{Cov}[\mathbf{x}_0 \mid \mathbf{x}_t]$ for the MMSE denoiser, it is expected that an approximated Jacobian will at least be symmetric and Positive Semi-Definite (PSD). These properties are also required for a preconditioner in the optimization literature. We show that trained denoisers may easily violate this. Figure 2 shows that the minimal eigenvalue of a trained denoiser is *negative*, indicating it is not PSD, and that it significantly deviates from symmetry. Further details on the synthetic environment implementation are given in Section B.3.

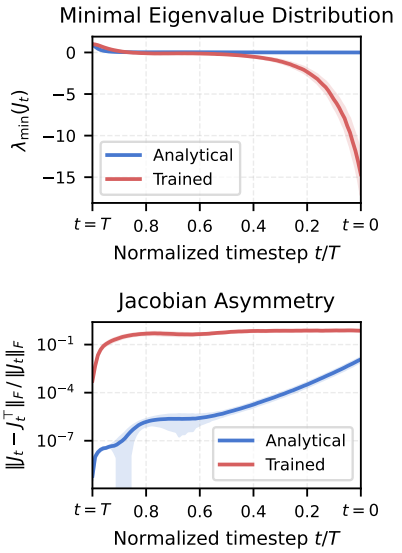


Figure 2: PSD and symmetry violations of trained Jacobians. The plots show mean curves with standard deviation over 1024 samples.

In this work, our main insight is that by constructing the direct and proximal likelihood approximations together, as shown in Section 3.1, we are able to mitigate harmful impact of the non-ideality of the Jacobian, while still leveraging the encoded knowledge within it, without extra computational cost.

3.3 The Proposed Guidance

Let us begin by constructing $\mathbf{u}_t$ and $\mathbf{v}_t$ as detailed in Section 3.1. We are interested in constructing an adaptive, enhanced likelihood surrogate $\mathbf{g}_t^*$ that leverages the knowledge within the Jacobian but is less sensitive to deviations resulting from non-idealities. Since $\mathbf{v}_t$ is already a decent approximation, it is natural to find the vector $\mathbf{g}_t \in \mathrm{span}\{\mathbf{u}_t\}$ that is closest to $\mathbf{v}_t$. In other words, we optimize a constant c_t for $\mathbf{g}_t = c_t\mathbf{u}_t$:

$$c_t^* = \arg\min_{c_t} \|c_t\mathbf{u}_t - \mathbf{v}_t\|_2^2 = \frac{\langle \mathbf{v}_t, \mathbf{u}_t\rangle}{\langle \mathbf{u}_t, \mathbf{u}_t\rangle} \quad \Longrightarrow \quad \mathbf{g}_t^* = \frac{\langle \mathbf{v}_t, \mathbf{u}_t\rangle}{\langle \mathbf{u}_t, \mathbf{u}_t\rangle}\mathbf{u}_t = \mathcal{P}_{\mathbf{u}_t}(\mathbf{v}_t). \tag{10}$$

Here, $\mathcal{P}_{\mathbf{u}_t}(\mathbf{v}_t)$ denotes the orthogonal projection of $\mathbf{v}_t$ onto $\mathbf{u}_t$. Notice that while in general $\mathbf{u}_t$ and $\mathbf{v}_t$ may be computed differently, or conceptually even come from a decoupled pair of proximal and direct likelihood approximations, one can obtain both with no extra computational cost. By choosing $\mathbf{u}_t$ and $\mathbf{v}_t$ as in Eq. 9 and Eq. 8, $\mathbf{v}_t$ can be extracted from the same computational graph as $\mathbf{u}_t$ for free. Our method, therefore, suggests

projecting the proximal solution $\mathbf{v}_t$ onto the direct, Jacobian-preconditioned solution $\mathbf{u}_t$, thus minimizing non-ideal Jacobian-related artifacts. Throughout the paper, we refer to our method as **J**acobian-**A**ware **P**osterior **S**ampling (JAPS).

3.4 Reformulating DDIM for Conditional Settings

Substituting Eq. 3 into the unconditional DDIM update Eq. 4, we can write it in a Markovian form:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\,\mathbf{x}_t - \underbrace{\left(\frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\alpha_t}} - \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\right)}_{\gamma_t} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + \sigma_t\,\boldsymbol{\epsilon}_t \;=:\; \mathrm{DDIM}(\mathbf{x}_t), \tag{11}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and γ_t collects time-dependent coefficients. Building on $\mathbb{E}[\boldsymbol{\epsilon}|\mathbf{x}_t, \mathbf{y}] = -\sqrt{1-\bar{\alpha}_t}\,\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$, which is a straightforward generalization of Tweedie's formula to the conditional case (see, e.g., Lemma A.2 in Peng et al. (2024)), we introduce a *posterior* noise $\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, \mathbf{y})$ analogously related to the *posterior* score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$. Using Eq. 5 and Eq. 2,

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, \mathbf{y}) := \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \lambda_t\,\sqrt{1-\bar{\alpha}_t}\,\mathbf{g}(\mathbf{y}, \mathbf{x}_t), \tag{12}$$

where $\mathbf{g}(\mathbf{y}, \mathbf{x}_t)$ is any tractable estimator of the likelihood-score term $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y} \mid \mathbf{x}_t)$ and $\lambda_t \in \mathbb{R}$ balances the (approximate) likelihood and prior scores. Plugging Eq. 12 into Eq. 11 (in lieu of $\boldsymbol{\epsilon}_\theta$) yields the conditional DDIM step:

$$\mathbf{x}_{t-1} \;=\; \mathrm{DDIM}(\mathbf{x}_t) + \gamma_t\,\lambda_t\,\sqrt{1-\bar{\alpha}_t}\;\mathbf{g}(\mathbf{y}, \mathbf{x}_t). \tag{13}$$

This formulation makes explicit that posterior information affects both the intermediate estimate $\hat{\mathbf{x}}_0(\mathbf{x}_t, t)$ (implicitly, through $\boldsymbol{\epsilon}_\theta$) and the projection back to $t-1$ via an effective noise that interpolates between the predicted noise and fresh randomness. In contrast, most direct methods add $\lambda_t\,\mathbf{g}(\mathbf{y}, \mathbf{x}_t)$ directly to DDIM, without explicitly accounting for the factor γ_t (Chung et al., 2023; Song et al., 2023), while proximal methods heuristically modify DDIM's injected noise estimate (Garber and Tirer, 2024; Zhu et al., 2023).

To conclude, our method is obtained by using Eq. 13 with $\mathbf{g}(\mathbf{y}, \mathbf{x}_t) = \mathbf{g}_t^*(\mathbf{y}, \mathbf{x}_t)$, given in Eq. 10.

3.5 Extension For Nonlinear Measurement Operators

Notice that while under the linear-Gaussian assumption $\mathbf{x}_0^* = \arg\max_{\mathbf{x}_0} \log p(\mathbf{x}_0 \mid \mathbf{x}_t\,, \mathbf{y}) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}]$, our method can be applied to other types of degradation models by treating $\mathbf{x}_0^*$ as an approximation of $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}]$. Specifically, other types of measurement noise would simply change the term $\log p(\mathbf{y}|\mathbf{x}_0)$. In addition, for nonlinear degradation models, the gradients of Eq. 6 with respect to $\mathbf{x}_0$ may be obtained via backpropagation through any differentiable model $\mathcal{A}(\cdot)$.

We now derive a likelihood score term for nonlinear models, in place of Eq. 9 and Eq. 8. For nonlinear models $\mathcal{A}(\cdot)$, we can locally linearize them using a Taylor expansion around $\hat{\mathbf{x}}_0$, namely:

$$\mathcal{A}(\mathbf{x}_0) \approx \mathcal{A}(\hat{\mathbf{x}}_0) + \mathbf{J}_{\mathcal{A}}(\hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0) \tag{14}$$

where $\mathbf{J}_{\mathcal{A}} = (\partial\mathcal{A}/\partial\mathbf{x}_0)^\top$ is the Jacobian of $\mathcal{A}$. Plugging Eq. 14 into the explicit distributions in Eq. 6 (as in Section A.2) yields the following MAP solution:

$$\mathbf{v}_t^{\mathrm{NL}} \;:=\; \tfrac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\mathbf{J}_{\mathcal{A}}^\top(\mathbf{J}_{\mathcal{A}}\mathbf{J}_{\mathcal{A}}^\top + \tfrac{\sigma_y^2}{r_t^2}\mathbf{I})^{-1}(\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0)), \tag{15}$$

and correspondingly $\mathbf{u}_t^{\mathrm{NL}} = \mathbf{J}_t\mathbf{v}_t^{\mathrm{NL}}$. For the rest of this paper, we omit the superscript $(\cdot)^{\mathrm{NL}}$ for brevity. The Jacobians can be computed using modern auto-differentiation frameworks, and the operator $(\mathbf{J}_{\mathcal{A}}\mathbf{J}_{\mathcal{A}}^\top + \frac{\sigma_y^2}{r_t^2}\mathbf{I})^{-1}$ can be computed via Conjugate Gradients.

	Method	Super Resolution	Gaussian Blur	Motion Blur	Random Inpainting	Box Inpainting
CelebA-HQ	DDPG	**29.40** / **0.838** / 0.086 / 40.95	**30.42** / **0.853** / 0.054 / **35.94**	**29.01** / **0.827** / 0.066 / **37.22**	25.63 / 0.780 / 0.146 / 54.59	25.98 / 0.852 / 0.073 / **30.11**
	DiffPIR	26.58 / 0.763 / 0.081 / **38.78**	28.92 / 0.813 / 0.064 / 37.32	27.34 / 0.783 / 0.078 / 39.14	26.74 / 0.808 / 0.138 / 52.45	25.88 / 0.859 / 0.096 / 46.27
	DAPS	28.12 / 0.781 / 0.075 / 28.63	28.34 / 0.756 / 0.091 / 32.66	28.42 / 0.794 / 0.072 / 26.36	26.65 / 0.753 / 0.120 / 38.14	24.14 / 0.723 / 0.165 / 33.74
	DPS	27.56 / 0.777 / 0.072 / 40.01	28.28 / 0.790 / 0.065 / 38.66	26.28 / 0.749 / 0.083 / 40.75	26.77 / 0.778 / 0.085 / 42.41	26.79 / 0.857 / 0.064 / 35.05
	DSG	27.57 / 0.780 / 0.072 / 40.77	30.29 / 0.845 / 0.051 / 36.05	27.57 / 0.790 / 0.079 / 38.48	**28.39** / **0.828** / 0.079 / 43.56	25.47 / 0.855 / 0.068 / 36.54
	ΠGDM	27.23 / 0.767 / 0.078 / 40.83	27.67 / 0.778 / 0.087 / 41.41	26.15 / 0.741 / 0.104 / 41.98	28.16 / 0.827 / 0.072 / **40.76**	27.11 / **0.875** / 0.050 / 32.55
	JAPS	28.39 / 0.806 / **0.070** / 39.76	30.25 / 0.845 / **0.050** / 36.28	28.70 / 0.814 / **0.061** / 37.58	27.90 / 0.805 / **0.069** / 42.61	**27.58** / 0.874 / **0.046** / 32.26
ImageNet	DDPG	**25.55** / **0.722** / 0.302 / 68.34	**27.74** / **0.793** / 0.170 / 37.15	**25.90** / **0.735** / 0.210 / 42.08	21.85 / 0.602 / 0.395 / 146.84	20.98 / 0.749 / 0.206 / 61.34
	DiffPIR	23.53 / 0.643 / 0.331 / 64.35	24.56 / 0.645 / 0.192 / 38.86	24.13 / 0.665 / 0.334 / 70.57	22.27 / 0.617 / 0.376 / 133.02	**21.07** / 0.761 / 0.244 / 85.88
	DAPS	23.60 / 0.626 / 0.422 / 77.95	21.74 / 0.504 / 0.331 / 124.94	25.14 / 0.696 / 0.240 / 51.09	22.80 / 0.595 / 0.309 / 81.09	20.26 / 0.675 / 0.248 / 57.67
	DPS	24.10 / 0.655 / 0.242 / 46.28	23.62 / 0.629 / 0.262 / 49.61	20.09 / 0.503 / 0.365 / 68.78	23.05 / 0.646 / 0.263 / 53.73	18.97 / 0.750 / 0.207 / 52.70
	DSG	24.33 / 0.669 / 0.203 / 39.89	26.69 / 0.744 / 0.153 / 30.19	24.33 / 0.667 / 0.197 / 41.23	21.91 / 0.576 / 0.236 / 54.88	19.18 / 0.769 / 0.151 / 40.13
	ΠGDM	22.85 / 0.595 / 0.268 / 57.04	22.83 / 0.607 / 0.227 / 48.30	20.97 / 0.528 / 0.292 / 58.49	**23.74** / 0.690 / 0.231 / 55.30	19.60 / **0.788** / 0.152 / 43.50
	JAPS	24.55 / 0.687 / **0.195** / **40.24**	27.16 / 0.770 / **0.149** / **32.69**	25.31 / 0.713 / **0.184** / **38.94**	23.71 / **0.694** / **0.168** / **48.75**	19.44 / 0.785 / **0.143** / **37.91**

Table 1: Performance metrics on CelebA-HQ and ImageNet datasets. Cells show PSNR ↑ / SSIM ↑ / LPIPS ↓ / FID ↓. Best results are in bold, and second best are underlined among the methods using standard configurations. Gray-texted methods are evaluated using 1000 NFEs, and are excluded from the ranking.

4 Experimental Results

Tasks and datasets. We test our method on the CelebA-HQ and ImageNet-256 validation sets, with backbone denoisers trained by Lugmayr et al. (2022) and Dhariwal and Nichol (2021), respectively. Our evaluation is performed by conducting several key image restoration tasks, used also in previous works: (i) *Super-resolution* ×4 with a bicubic downsampling kernel; (ii) *Gaussian deblurring* with a 5×5 Gaussian kernel (standard deviation 10); (iii) *Motion deblurring* with randomized 61×61 kernels of intensity 0.5, generated using the public implementation[1]; (iv) *Inpainting*, with masks of both a 128 × 128 box, or 90% random pixels (Box/Random Inpainting, respectively). Unless noted otherwise, we add zero-mean i.i.d. Gaussian noise with $\sigma_y = 0.05$, conventionally expressed in $[0, 1]$ intensity units. Since we normalize images to the range $[-1, 1]$, we accordingly multiply the noise level by a factor of 2.

Baselines. We compare against DAPS (Zhang et al., 2025), DiffPIR (Zhu et al., 2023), DDPG (Garber and Tirer, 2024), DPS (Chung et al., 2023) ΠGDM (Song et al., 2023), and DSG (Yang et al., 2024). The first 3 are referred as proximal-typed solutions, while the last 3 are direct-typed works. We report the PSNR, SSIM, LPIPS (AlexNet variant) and FID metrics averaged or computed over 1K samples per dataset.

Sampling details. We use a DDIM sampler with $\eta = 1$ (i.e., DDPM-equivalent) and 100 diffusion steps, which amount to 100 NFEs. Unless noted otherwise, all baselines also use 100 NFEs. Exceptions are DPS and DAPS, which require 1,000 NFEs, and DSG, which likewise uses 1,000 steps for ImageNet-256.

Visualizations. Results from the experiments in Sections 4.1 and 4.2 are presented in Section E.

4.1 Linear Inverse Problems

Table 1 compares JAPS with representative posterior samplers built on diffusion priors. Across both datasets, JAPS delivers state-of-the-art perceptual quality, attaining the best or second-best LPIPS in all settings. At the same time, it maintains strong PSNR, incurring only a modest distortion cost, consistent with the perception–distortion trade-off (Blau and Michaeli, 2018). Notably, several PSNR-oriented baselines significantly sacrifice LPIPS, especially on noisy tasks, resulting in visibly blurrier reconstructions (e.g., DDPG on SR×4, or DSG on Random Inpainting). In contrast, JAPS remains competitive on *both* metrics across tasks and datasets. These results are also visualized in Figure 1b, which averages the PSNR and LPIPS of each method across these linear tasks, demonstrating that, on average, our method excels in both perceptual and distortion metrics.

[1] `https://github.com/LeviBorodenko/motionblur`

4.2 Nonlinear Inverse Problems

We test our method for nonlinear deblurring and high dynamic range correction as representative nonlinear tasks. Following Chung et al. (2023); Zhang et al. (2025), which are not naturally restricted to linear inverse problems, we adopt the neural blur model of Tran et al. (2021) with measurement noise $\sigma_y = 0.05$, and compare against them. Additionally, we evaluate high dynamic range correction with a factor of 2, also with measurement noise $\sigma_y = 0.05$, and compare to Zhang et al. (2025). In both cases we test on 100 CelebA-HQ and ImageNet-256 images. For a complete comparison, we also evaluate ΠGDM (Song et al., 2023) with our nonlinear extension derived in Section 3.5. We report PSNR, SSIM and LPIPS. The results in Table 2 show that our nonlinear extension consistently performs best among compared methods, despite being an approximation.

Table 2: Comparison of nonlinear tasks across methods and datasets, measured by PSNR (↑) / SSIM (↑) / LPIPS (↓).

Method	ImageNet-256	CelebA-HQ
Nonlinear Blur		
DPS	19.12 / 0.517 / 0.363	23.31 / 0.702 / 0.133
DAPS	19.98 / 0.364 / 0.586	24.75 / 0.667 / 0.144
ΠGDM	21.87 / 0.598 / 0.345	25.52 / 0.754 / 0.109
JAPS	**23.64** / **0.677** / **0.213**	**26.40** / **0.769** / **0.097**
High Dynamic Range		
DAPS	24.55 / 0.822 / 0.110	25.52 / 0.810 / 0.084
ΠGDM	11.11 / 0.434 / 0.522	22.56 / 0.794 / 0.124
JAPS	**24.70** / **0.841** / **0.099**	**26.74** / **0.848** / **0.067**

4.3 Ablation Studies

In this section, we analyze various aspects in our method.

Analyzing our Likelihood Score Surrogate In Section 3.3, we propose to orthogonally project a proximal likelihood score surrogate $\mathbf{v}_t$ onto a direct one $\mathbf{u}_t$ (e.g. the terms in Eq. 8 and Eq. 9). To validate that our solution in Eq. 10 indeed performs best, we vary our choice for the likelihood score surrogate $\mathbf{g}(\mathbf{y}, \mathbf{x}_t)$ from Eq. 13 by plugging each of the options $\mathbf{g}(\mathbf{y}, \mathbf{x}_t) \in \{\mathbf{u}_t, \mathbf{v}_t, \mathcal{P}_{\mathbf{v}_t}(\mathbf{u}_t), \mathcal{P}_{\mathbf{u}_t}(\mathbf{v}_t), (\mathbf{u}_t + \mathbf{v}_t)/2\}$. For a fair comparison, we sweep over different values $\lambda_t \in \{0.5, 1, 2, 3, 4, 5, 6, 7, 8\}$, and run 3 representative tasks on a subset of 30 ImageNet-256 images with a fixed seed. Figure 3 summarizes the results, and shows that our choice of $\mathbf{g}(\mathbf{y}, \mathbf{x}_t) = \mathcal{P}_{\mathbf{u}_t}(\mathbf{v}_t)$ attains the best performance across all demonstrated tasks upon proper tuning.

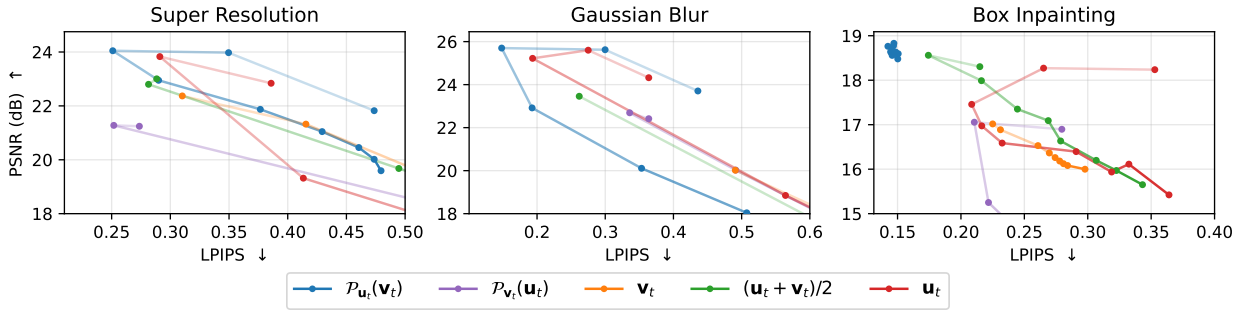


Figure 3: **Ablation study of guidance surrogates.** We compare the sampling scheme in Eq. 13 using various candidates for $\mathbf{g}(\mathbf{y}, \mathbf{x}_t) \in \{\mathbf{u}_t, \mathbf{v}_t, \mathcal{P}_{\mathbf{v}_t}(\mathbf{u}_t), \mathcal{P}_{\mathbf{u}_t}(\mathbf{v}_t), (\mathbf{u}_t + \mathbf{v}_t)/2\}$. The curves illustrate the perception–distortion trade-off (LPIPS vs. PSNR) on ImageNet-256, where each point corresponds to a different guidance scale $\lambda_t \in \{0.5, 1, 2, 3, 4, 5, 6, 7, 8\}$.

Additional Experiments. We further evaluate JAPS under increasing measurement noise levels and varying degrees of stochasticity in the diffusion sampling process. Our results show that JAPS attenuates the unavoidable degradation in reconstruction quality at high noise levels and is substantially less sensitive to the stochasticity setting of the diffusion process. Additional details are provided in Section C.1.

5 Conclusion

We introduced **JAPS**, a principled approach for posterior sampling that bridges direct and proximal likelihood approximations. By leveraging the denoiser's Jacobian while mitigating its training-induced non-idealities, JAPS achieves a competitive balance between perceptual quality and distortion across diverse linear and nonlinear tasks.

Limitations. A primary limitation is the current reliance on pixel-domain diffusion. Extending JAPS to latent diffusion models (Rombach et al., 2022) is non-trivial, as posterior sampling becomes entangled with the decoder when measurements are defined in the image domain (Rout et al., 2023; Song et al., 2024). Furthermore, while we focus on diffusion models, current state-of-the-art generative modeling is increasingly shifting toward flow-based models (Lipman et al., 2023; Liu et al., 2022). Adapting our insights regarding the Jacobian of the trained backbone to such architectures remains a promising direction for future research.

Outlook. To the best of our knowledge, this is the first work to analyze the impact of Jacobian non-ideality in zero-shot posterior sampling settings. We believe JAPS provides a mathematically grounded foundation for more reliable inverse problem solving without task-specific tuning. Future work should further explore methods to explicitly adjust or regularize the desired properties of the Jacobian to better suit specific downstream tasks, facilitating more robust posterior guidance in both pixel and latent generative models.

Our code will be released upon acceptance.

References

Shady Abu-Hussein, Tom Tirer, and Raja Giryes. Adir: Adaptive diffusion for image reconstruction. *arXiv preprint arXiv:2212.03221*, 2022.

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3): 313–326, 1982.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 6228–6237. IEEE, 2018.

Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and O Deniz Akyildiz. Tweedie moment projected diffusions for inverse problems. *arXiv preprint arXiv:2310.06721*, 2023.

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.

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*, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496): 1602–1614, 2011.

Tomer Garber and Tom Tirer. Image restoration by denoising diffusion models with iteratively preconditioned guidance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25245–25254, 2024.

Tomer Garber and Tom Tirer. Zero-shot image restoration using few-step guidance of consistency models (and beyond). In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2398–2407, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Shady Abu Hussein, Tom Tirer, and Raja Giryes. Correction filter for single image super-resolution: Robustifying off-the-shelf deep super-resolvers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1428–1437, 2020.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in neural information processing systems*, 35:23593–23606, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. *arXiv preprint arXiv:2305.04391*, 2023.

Chenlin Meng, Yang Song, Wenzhe Li, and Stefano Ermon. Estimating high order gradients of the data distribution by denoising. *Advances in Neural Information Processing Systems*, 34:25359–25369, 2021.

Xinyu Peng, Ziyang Zheng, Wenrui Dai, Nuoqian Xiao, Chenglin Li, Junni Zou, and Hongkai Xiong. Improving diffusion models for inverse problems using optimal posterior covariance. In *International Conference on Machine Learning*, pages 40347–40370. PMLR, 2024.

Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM journal on imaging sciences*, 10(4):1804–1844, 2017.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alexandros G. Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models, 2023.

Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3118–3126, 2018.

Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency, 2024.

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

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

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005*, 2021.

Tom Tirer and Raja Giryes. Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing*, 28(3):1220–1234, 2018.

Tom Tirer and Raja Giryes. Super-resolution via image-adapted denoising cnns: Incorporating external and internal learning. *IEEE Signal Processing Letters*, 26(7):1080–1084, 2019.

Phong Tran, Anh Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via blur kernel space. *arXiv preprint arXiv:2104.00317*, 2021.

Roy Turgeman and Tom Tirer. Does the data processing inequality reflect practice? on the utility of low-level tasks. In *The Fourteenth International Conference on Learning Representations*, 2026.

Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE global conference on signal and information processing*, pages 945–948. IEEE, 2013.

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

Lingxiao Yang, Shutong Ding, Yifan Cai, Jingyi Yu, Jingya Wang, and Ye Shi. Guidance with spherical gaussian constraint for conditional diffusion. In *International Conference on Machine Learning*, pages 56071–56095. PMLR, 2024.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20895–20905, 2025.

Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1219–1229, 2023.

A Proofs and Derivations

A.1 Proof of Proposition 3.1

Proof. By Bayes' rule, $p(\mathbf{y} \mid \mathbf{x}_t) = p(\mathbf{x}_t \mid \mathbf{y})\, p(\mathbf{y}) \,/\, p(\mathbf{x}_t)$. Taking the gradient of the logarithm with respect to $\mathbf{x}_t$,

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{y}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t). \tag{16}$$

The rightmost term is the score, which relates to the mean $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$ via Tweedie's Formula (Efron, 2011):

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) = \frac{-\mathbf{x}_t + \sqrt{\bar{\alpha}_t}\, \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]}{1 - \bar{\alpha}_t}\,. \tag{17}$$

Now, the assumption $\mathbf{x}_t \perp\!\!\!\perp \mathbf{y} \mid \mathbf{x}_0$ implies $p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{y}) = p(\mathbf{x}_t \mid \mathbf{x}_0)$, so the class-conditional marginal factorises as

$$p(\mathbf{x}_t \mid \mathbf{y}) = \int p(\mathbf{x}_t \mid \mathbf{x}_0)\, p(\mathbf{x}_0 \mid \mathbf{y})\, d\mathbf{x}_0\,.$$

$p(\mathbf{x}_0 \mid \mathbf{y})$ in place of $p(\mathbf{x}_0)$ in Tweedie's Formula, the posterior that appears is now $p(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y})$, giving

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{y}) = \frac{-\mathbf{x}_t + \sqrt{\bar{\alpha}_t}\, \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}]}{1 - \bar{\alpha}_t}\,. \tag{18}$$

Substituting Eq. 17 and Eq. 18 into Eq. 16, the $\mathbf{x}_t$ terms cancel, yielding

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) = \frac{\sqrt{\bar{\alpha}_t}}{1 - \bar{\alpha}_t} \Big(\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}] - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t] \Big). \qquad \square$$

Remark. This identity is exact and requires no Gaussian approximation on $p(\mathbf{x}_0 \mid \mathbf{x}_t)$. It decomposes the intractable likelihood score into the difference of two posterior means, scaled by the signal-to-noise geometry of the forward process.

A.2 Derivation of Eq. 8

Let $\Phi(\mathbf{x}_0) = -\log p(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{y})$. Under the isotropic Gaussian assumption on $p(\mathbf{x}_0 \mid \mathbf{x}_t)$ and the linearity of $\mathbf{A}$, $\Phi(\mathbf{x}_0)$ gets the form:

$$\Phi(\mathbf{x}_0) = \frac{1}{2\sigma_y^2} \|\mathbf{y} - \mathbf{A}\mathbf{x}_0\|_2^2 + \frac{1}{2r_t^2} \|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|_2^2 + C. \tag{19}$$

Moreover, under these assumptions, $\mathbf{x}_0^* = \arg\min_{\mathbf{x}_0} \Phi(\mathbf{x}_0) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$.

Since Φ is quadratic, its minima can be achieved using a single step from any initial point $\mathbf{x}_p$, in the Newton descent direction, defined by $\mathbf{d} = -\mathbf{H}^{-1}\mathbf{g}$, where $\mathbf{g} := \nabla_{\mathbf{x}_0}\Phi(\mathbf{x}_0)|_{\mathbf{x}_0=\mathbf{x}_p}$ and $\mathbf{H} := \nabla^2_{\mathbf{x}_0}\Phi(\mathbf{x}_0)|_{\mathbf{x}_0=\mathbf{x}_p}$, the gradient and the Hessian at the point $\mathbf{x}_p$.

Taking $\mathbf{x}_p = \hat{\mathbf{x}}_0$, we get $\mathbf{g} = -\sigma_y^{-2}\mathbf{A}^\top(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0)$, and $\mathbf{H} = \sigma_y^{-2}\mathbf{A}^\top\mathbf{A} + r_t^{-2}\mathbf{I}$, thus,

$$\mathbf{d} = \big(\mathbf{A}^\top\mathbf{A} + \tfrac{\sigma_y^2}{r_t^2}\mathbf{I}\big)^{-1}\mathbf{A}^\top(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0) \tag{20}$$

For any $\mathbf{A} \in \mathbb{R}^{n\times m}$, recall the "push-through" identity,

$$\big(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I}_n\big)^{-1}\mathbf{A}^\top \;=\; \mathbf{A}^\top\big(\mathbf{A}\mathbf{A}^\top + \lambda\mathbf{I}_m\big)^{-1} \qquad \forall\, \lambda > 0, \tag{21}$$

Using Eq. 21, we get $\mathbf{d} = \mathbf{A}^\top\big(\mathbf{A}\mathbf{A}^\top + \frac{\sigma_y^2}{r_t^2}\mathbf{I}\big)^{-1}(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0)$

Hence, since the optimum $\mathbf{x}_0^*$ is achieved with a single step descent $\mathbf{d}$ from $\hat{\mathbf{x}}_0$, $\mathbf{x}_0^* - \hat{\mathbf{x}}_0 = \mathbf{d}$. Plugging this into Prop. 3.1, we get the expression for $\mathbf{v}_t$ in Eq. 8.

A.3 Derivation of Eq. 9

Directly from Song et al. (2023),

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) = \big(\frac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t}\big)^\top \mathbf{A}^\top \big(r_t^2 \mathbf{A}\mathbf{A}^\top + \sigma_y^2 \mathbf{I}\big)^{-1} (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0). \tag{22}$$

Taking r_t^2 outside of $\big(r_t^2 \mathbf{A}\mathbf{A}^\top + \sigma_y^2 \mathbf{I}\big)^{-1}$, we get the expression of $\mathbf{u}_t$ in Eq. 9.

A.4 Proof of Mean-Mode Coincidence for a Linear-Gaussian Model

Proposition A.1. *Let* $\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \boldsymbol{\eta}$, $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2 \mathbf{I})$, $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$ *the forward diffusion kernel, with* $\mathbf{x}_t \perp\!\!\!\perp \mathbf{y} \mid \mathbf{x}_0$. *If* $p(\mathbf{x}_0 \mid \mathbf{x}_t)$ *is Gaussian, then* $\arg\max_{\mathbf{x}_0} \log p(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}]$.

Proof. By Bayes' rule and the conditional independence assumption,

$$\log p(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}) = \underbrace{\log p(\mathbf{y} \mid \mathbf{x}_0)}_{\text{likelihood}} + \underbrace{\log p(\mathbf{x}_0 \mid \mathbf{x}_t)}_{\text{prior}} + \text{const.}$$

The prior is Gaussian by assumption. The likelihood $p(\mathbf{y} \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{A}\mathbf{x}_0, \sigma_y^2 \mathbf{I})$ is quadratic in $\mathbf{x}_0$ because $\mathbf{A}$ is linear. The sum of two quadratics is quadratic, so $p(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y})$ is Gaussian. Since the mode and mean of a Gaussian coincide, the MAP estimate equals the posterior mean $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{y}]$. □

A.5 The Relation between the Posterior Covariance and the Jacobian

Proposition A.2. *Let* $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ *define a forward noising process where* $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, *and* $\bar{\alpha}_t \in (0,1)$ *is the variance-preserving noise schedule. Let* $\hat{\mathbf{x}}_0(\mathbf{x}_t) = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ *denote the ideal Minimum Mean Squared Error (MMSE) denoiser. The Jacobian matrix of the denoiser with respect to the noisy input* $\mathbf{x}_t$, *denoted as* $\mathbf{J}_t = \nabla_{\mathbf{x}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t)$, *satisfies:*

$$\mathbf{J}_t = \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\mathrm{Cov}(\mathbf{x}_0|\mathbf{x}_t) \tag{23}$$

where $\mathrm{Cov}(\mathbf{x}_0|\mathbf{x}_t)$ *is the posterior covariance matrix of* $\mathbf{x}_0$ *given* $\mathbf{x}_t$.

Proof. By the definition of the conditional expectation, the Jacobian $\mathbf{J}_t = \nabla_{\mathbf{x}_t}\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ can be written as:

$$\mathbf{J}_t = \nabla_{\mathbf{x}_t} \int \mathbf{x}_0 p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 = \int \mathbf{x}_0 \nabla_{\mathbf{x}_t} p(\mathbf{x}_0|\mathbf{x}_t)^\top d\mathbf{x}_0 \tag{24}$$

Applying the log-derivative trick, $\nabla_{\mathbf{x}_t} p(\mathbf{x}_0|\mathbf{x}_t) = p(\mathbf{x}_0|\mathbf{x}_t)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_0|\mathbf{x}_t)$, yielding:

$$\mathbf{J}_t = \int \mathbf{x}_0 \left(\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_0|\mathbf{x}_t)\right)^\top p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \tag{25}$$

Using Bayes' theorem, the log-posterior is $\log p(\mathbf{x}_0|\mathbf{x}_t) = \log p(\mathbf{x}_t|\mathbf{x}_0) + \log p(\mathbf{x}_0) - \log p(\mathbf{x}_t)$. Differentiating with respect to $\mathbf{x}_t$ gives:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_0|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \tag{26}$$

Given the forward noising process $\mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, the gradient of the log-likelihood is:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x}_0) = -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{1-\bar{\alpha}_t} \tag{27}$$

By Tweedie's formula, the gradient of the log-marginal distribution is related to the MMSE denoiser via:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) = \frac{\sqrt{\bar{\alpha}_t}\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] - \mathbf{x}_t}{1-\bar{\alpha}_t} \tag{28}$$

Substituting these into Equation 26:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_0|\mathbf{x}_t) = -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{1-\bar{\alpha}_t} - \frac{\sqrt{\bar{\alpha}_t}\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] - \mathbf{x}_t}{1-\bar{\alpha}_t} \tag{29}$$

$$= \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}(\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]) \tag{30}$$

Substituting this result back into the integral for $\mathbf{J}_t$:

$$\mathbf{J}_t = \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t} \int \mathbf{x}_0(\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t])^\top p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \tag{31}$$

$$= \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t} \mathbb{E}\left[\mathbf{x}_0(\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t])^\top \middle| \mathbf{x}_t\right] \tag{32}$$

$$= \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t} \mathrm{Cov}(\mathbf{x}_0|\mathbf{x}_t) \tag{33}$$

This completes the proof. □

B Additional Method Details

B.1 The Choice of r_t^2

While Song et al. (2023) set $r_t^2 = 1 - \bar{\alpha}_t$ in ΠGDM, we propose $r_t^2 = (1-\bar{\alpha}_t)/\sqrt{\bar{\alpha}_t}$. Both choices yield a decaying noise level along the reverse process, from $t = T$ toward the clean image at $t = 0$. However, under the widely adopted schedules in which $\mathrm{Var}(\mathbf{x}_{t+1} \mid \mathbf{x}_t) = \mathrm{Var}(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ for every t, the original choice $r_t^2 = 1 - \bar{\alpha}_t$ remains nearly constant from $t = T$ to $t = T/2$ (see Figure 4). Combined with a fixed σ_y throughout sampling, this results in an essentially unchanged prior-data trade-off over a large portion of the reverse process. Our choice, by contrast, exhibits a consistently steeper decay across the entire sampling trajectory, leading to a more gradual and effective transition from prior to data.

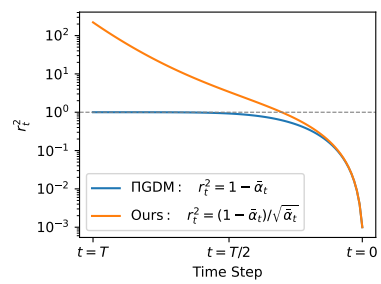


Figure 4: r_t^2 schedules across the diffusion process.

B.2 Measurement Space Computations

In many inverse problems, $m < n$ (e.g. super-resolution). In such cases, it is preferable in terms of memory consumption to compute $\mathbf{u}_t$, $\mathbf{v}_t$ in the measurement space using Eq. 21. Our implementations indeed perform such computations in the measurement space, i.e:

$$\mathbf{v}_t = \tfrac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \tfrac{\sigma_y^2}{r_t^2}\mathbf{I}_m)^{-1}(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0) \tag{34}$$

$$\mathbf{u}_t = \tfrac{1}{r_t^2}(\tfrac{\partial \hat{\mathbf{x}}_0}{\partial \mathbf{x}_t})^\top \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \tfrac{\sigma_y^2}{r_t^2}\mathbf{I}_m)^{-1}(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0) \tag{35}$$

Furthermore, this measurement-space form is tightly connected to $\mathbf{A}^\dagger = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)$, the Moore-Penrose pseudo inverse of $\mathbf{A}$. In noiseless cases ($\sigma_y = 0$), the forms in Eq. 34 coincide with the Moore-Penrose pseudo inverse. In noisy cases, this form is a variant of the Moore-Penrose pseudo inverse, loaded with a regularization scalar $\eta_t := \frac{\sigma_y^2}{r_t^2}$.

B.3 Synthetic GMM Environment

To enable controlled analysis of Jacobian properties, we construct a synthetic environment based on a Gaussian Mixture Model (GMM) in $\mathbb{R}^D$, where the ground-truth distribution and all derived quantities are available in closed form.

GMM Construction. We define a K-component GMM with density $p(\mathbf{x}_0) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $w_k = 1/K$ are uniform weights. The component means $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$ are placed on a hypersphere of radius r=3 by normalizing i.i.d. Gaussian random vectors. Each covariance matrix is constructed as $\boldsymbol{\Sigma}_k = \mathbf{Q}_k \operatorname{diag}(\boldsymbol{\lambda}_k) \mathbf{Q}_k^\top$, where $\mathbf{Q}_k$ is a random orthogonal matrix (via QR decomposition of a Gaussian random matrix) and eigenvalues $\lambda_{k,i}$ are drawn uniformly from $[0, 0.2]$. The main results use D=32 and K=8.

Under the forward diffusion process $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon}$, the noisy marginal $p(\mathbf{x}_t)$ remains a GMM with evolved parameters:

$$p(\mathbf{x}_t) = \sum_{k=1}^{K} w_k \mathcal{N}\big(\mathbf{x}_t;\, \sqrt{\bar{\alpha}_t}\, \boldsymbol{\mu}_k,\, \bar{\alpha}_t \boldsymbol{\Sigma}_k + (1 - \bar{\alpha}_t)\mathbf{I}\big)\,. \tag{36}$$

The prior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ is computed in closed form via the mixture gradient, and serves as the analytical denoiser through $\hat{\mathbf{x}}_0(\mathbf{x}_t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$. To analyze the Jacobian $\mathbf{J}_t = \partial \hat{\mathbf{x}}_0 / \partial \mathbf{x}_t$, we compute the full $D \times D$ Jacobian matrix using automatic differentiation (`torch.autograd.functional.jacobian`) applied to the denoiser mapping $\mathbf{x}_t \mapsto \hat{\mathbf{x}}_0(\mathbf{x}_t)$. This procedure is identical for both the analytical MMSE denoiser and the trained neural network. Given the full Jacobian $\mathbf{J}_t$, we assess PSD violation via the minimum eigenvalue of its symmetric part $\lambda_{\min}\big((\mathbf{J}_t + \mathbf{J}_t^\top)/2\big)$, and asymmetry via the relative Frobenius norm $\|\mathbf{J}_t - \mathbf{J}_t^\top\|_F / \|\mathbf{J}_t\|_F$. Both quantities are computed per sample at each diffusion timestep. Figure 2 reports means and standard deviations of $\lambda_{\min}(\mathbf{J}_t)$ and $\|\mathbf{J}_t - \mathbf{J}_t^\top\|_F / \|\mathbf{J}_t\|_F$ over 1024 samples.

Remark B.1. Because the quadratic form $\mathbf{v}^\top \mathbf{J}_t \mathbf{v}$ is strictly equal to $\mathbf{v}^\top \left(\frac{\mathbf{J}_t + \mathbf{J}_t^\top}{2}\right) \mathbf{v}$ for any real vector $\mathbf{v}$, $\mathbf{J}_t$ is PSD if and only if its symmetric part is PSD. Thus, a negative minimum eigenvalue of the symmetric part is a definitive proof of PSD violation for the original matrix $\mathbf{J}_t$.

Trained Diffusion Model. We train a small MLP-based diffusion model on samples from the GMM. The architecture consists of a sinusoidal time embedding followed by an MLP with 128 hidden channels and 2 residual blocks, predicting the noise $\boldsymbol{\epsilon}$ via the standard denoising score matching objective $\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\big]$. The model is trained for 5,000 epochs using AdamW with learning rate $5{\times}10^{-4}$, weight decay 10^{-6}, cosine annealing schedule, and batch size 1,024. We use a linear noise schedule with T=1,000 diffusion steps and sample with DDIM using 100 steps.

C Additional Results

C.1 Additional Experiments

Measurement noise level. In many posterior sampling schemes, increasing the observation noise not only removes information but can also *leak* through the measurement-consistency gradients into the reconstruction. We evaluate SR×4 across increasing σ_y and compare JAPS to ΠGDM using PSNR and LPIPS. We conduct this comparison against ΠGDM due to the structural resemblance of the likelihood surrogates, differing mainly in magnitude. Figure 5 shows that JAPS mitigates the inevitable degradation, with quality declining roughly linearly as the noise grows, whereas ΠGDM exhibits a markedly sharper drop.

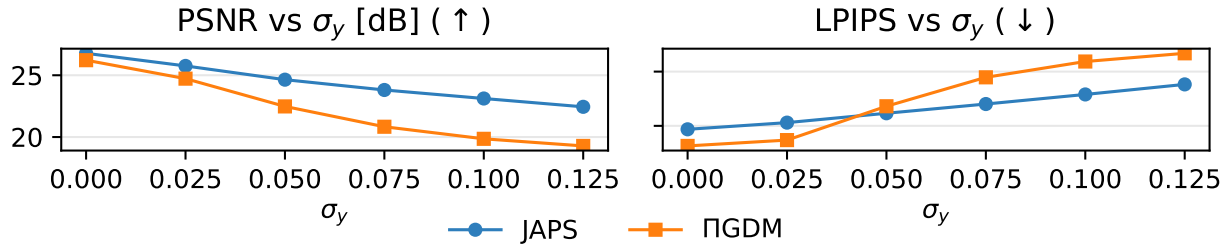


Figure 5: **Response to increasing measurement noise level.** JAPS shows enhanced robustness to increased noise compared to ΠGDM.

Stochasticity. Many posterior samplers set $\eta = 1$ (see Section 2.2), allowing fresh noise to mitigate artifacts that arise when enforcing consistency with noisy observations. We evaluate JAPS on SR×4 across a range of η values and compare against ΠGDM. Figure 6 shows that, whereas ΠGDM benefits primarily from highly stochastic updates (large η), JAPS exhibits markedly weaker dependence on η, maintaining similar performance across stochasticity levels.

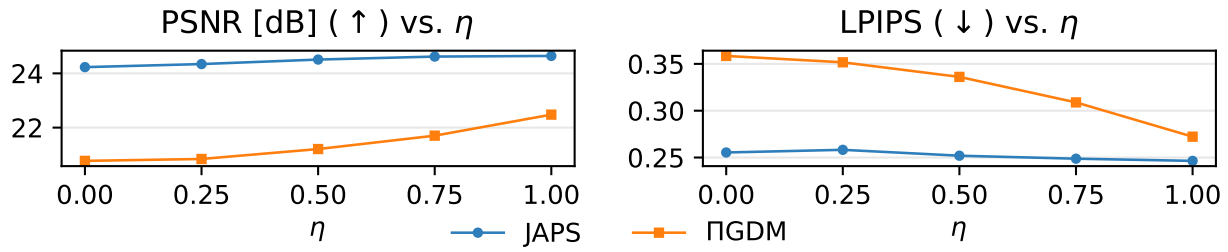
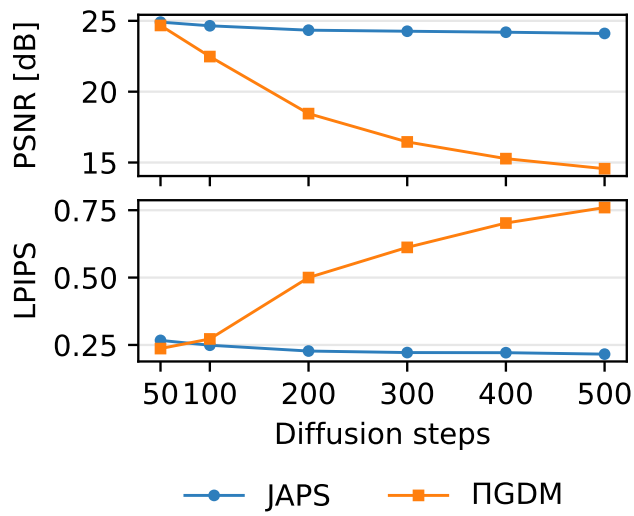
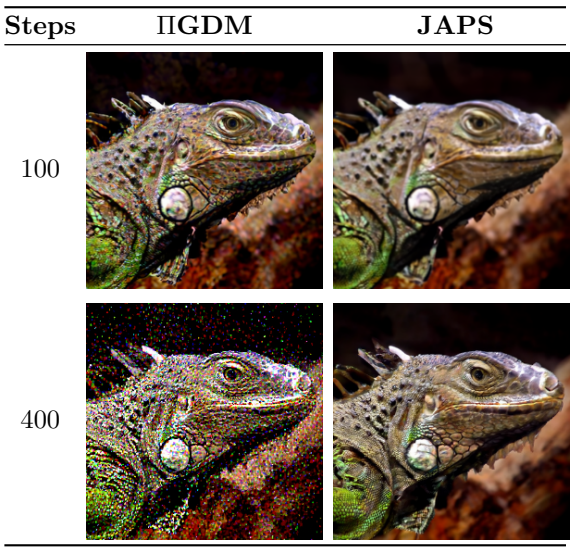


Figure 6: **Sensitivity to stochasticity (η).** PSNR/LPIPS for SR×4 as a function of η. JAPS maintains similar performance across stochasticity levels, whereas ΠGDM is more sensitive to the amount of injected noise.

Direct Comparison to ΠGDM. Beyond the likelihood approximation, Song et al. (2023) introduce a time-decaying multiplicative step size equal to $(1 - \bar{\alpha}_t)$. While this choice performs well at 100 sampling steps, it does not account for the diffusion schedule's discretization (i.e., changing the number of steps). Consequently—and counter-intuitively—*increasing* the number of steps degrades performance in both PSNR and LPIPS, rather than improving it, as was also observed by Mardani et al. (2023). By solving SR×4 with $\sigma_y = 0.05$ on 100 samples from the ImageNet-256 validation set, we show that our method explicitly incorporates step spacing through γ_t, yielding a correctly scalable sampler (Figs. 7a and 7b).



(a) Quantitative comparison. JAPS scales with the number of steps, while ΠGDM does not.



(b) Qualitative comparison. ΠGDM deteriorates at larger step counts, while JAPS remains stable.

Figure 7: Overall comparison of JAPS and ΠGDM across varying step counts. JAPS demonstrates consistent scaling both quantitatively (a) and qualitatively (b).

D Implementation Details

D.1 Hyperparameters

In this section, we detail the values of the hyperparameter λ_t as used in our final experiments, and discuss their contribution. Table 3 details the values.

Task	Bicub. SR×4 $\sigma_y = 0.05$	Gauss.Deb. $\sigma_y = 0.05$	Motion Deb. $\sigma_y = 0.05$	Random Inp. $\sigma_y = 0.05$	Box Inp. $\sigma_y = 0.05$	Nonlinear Deb. $\sigma_y = 0.05$	HDR $\sigma_y = 0.05$
ImageNet	$4\sqrt{1-\bar{\alpha}_t}$	1.8	1.8	6	$19\sqrt{1-\bar{\alpha}_t}$	4	4
CelebA-HQ	1	1.25	1.25	6	$16\sqrt{1-\bar{\alpha}_t}$	$4\sqrt{1-\bar{\alpha}_t}$	4

Table 3: Hyperparameter λ_t of our method.

While these hyperparameters improve performance, tuning them is not the primary source of improvement in our method relative to others. To justify this claim, we show a direct comparison of JAPS with different hyperparameters and also of ΠGDM Song et al. (2023) using the same sweep. Specifically, we scale the likelihood score surrogate by a time-dependent constant in lieu of the hyperparameter λ_t from Eq. 13. We note that for ΠGDM, the $\sqrt{1-\bar{\alpha}_t}$ term is absent; we instead use their proposed adaptive weights and scale them to align both methods within the same sweep settings. When using the original ΠGDM scaling (e.g., 1), we find that the optimal choice of λ_t for our method surpasses ΠGDM performance in both PSNR and LPIPS across almost all tasks for ImageNet-256. We note that a 0.5 scale for ΠGDM performs better than the default 1 scaling for most tasks, however, our method still either improves beyond this or stays on par. The full results are shown in Figure 8, indicating that the adaptiveness of the JAPS surrogate to the proximal solution cannot be replicated by simple time-dependent scaling.

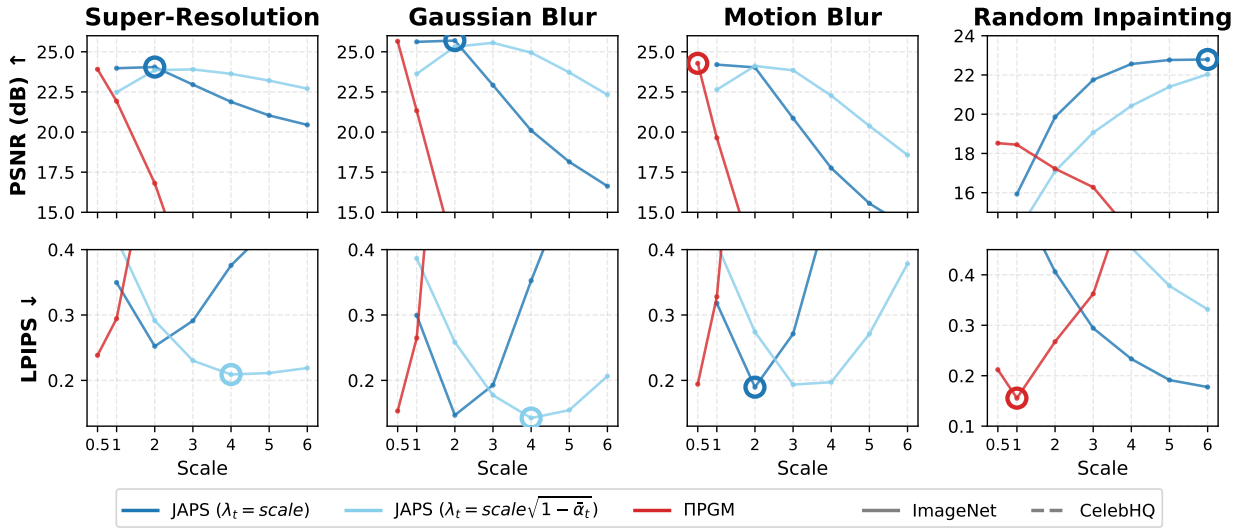


Figure 8: Hyperparameter sweep for ImageNet-256. The best scores per task are circled.

D.2 Reproducibility

D.2.1 Baselines Details

We reproduce all baselines within a unified environment to ensure consistency across tasks, noise levels, and datasets. To verify the accuracy of our reproduction, we first replicated the results reported in the original papers for standard tasks. Once consistency was confirmed, we evaluated each method across the full suite of tasks and datasets considered in this work.

DDPG. We adopt the implementation details and hyperparameters reported by Garber and Tirer (2024). For inpainting tasks not covered in the original work, we conducted a grid search for each hyperparameter. The optimized values are summarized in Table 4.

Table 4: DDPG inpainting hyperparameters.

Task	CelebA-HQ	ImageNet
Random Inpainting ($\sigma_y = 0.05$)	$\gamma = 3.0$, $\zeta = 0.9$, $\tilde{\eta} = 0.1$, $\mu_t = \mu_t^*$	$\gamma = 9.0$, $\zeta = 0.9$, $\tilde{\eta} = 0.1$, $\mu_t = \mu_t^*$
Box Inpainting ($\sigma_y = 0.05$)	$\gamma = 9.0$, $\zeta = 0.8$, $\tilde{\eta} = 0.9$, $\mu_t = \mu_t^*$	$\gamma = 9.0$, $\zeta = 1.0$, $\tilde{\eta} = 0.7$, $\mu_t = \mu_t^*$

DiffPIR. We follow the implementation details from Zhu et al. (2023), including all reported hyperparameters. For our CelebA-HQ experiments, we utilize the hyperparameters specified by Zhu et al. (2023) for the FFHQ dataset.

DAPS. We follow the implementation in Zhang et al. (2025). In our framework, we scale σ_y by a factor of 2 to account for the difference in image normalization (shifting from $[0, 1]$ to $[-1, 1]$). Consequently, our reproduction of Zhang et al. (2025) effectively solves tasks with twice the original noise level. To compensate for this, we adjusted only the $\sigma_{\min}$ value of the annealing process from 0.1 to 0.2, while retaining all other original parameters.

DPS. We adopt the implementation details and hyperparameters from Chung et al. (2023) as reported.

DSG. We utilize the implementation details from Yang et al. (2024). For CelebA-HQ, we adopt the FFHQ hyperparameters. For motion deblurring, we employ the hyperparameters originally reported for Gaussian deblurring, as we found these tasks to be closely related in performance behavior.

ΠGDM. We adopt the implementation from Song et al. (2023), following the publicly released RED-diff codebase (Mardani et al., 2023)[2].

Our code will be released upon acceptance.

[2] https://github.com/NVlabs/RED-diff/blob/master/algos/pgdm.py

E Additional Visual Results

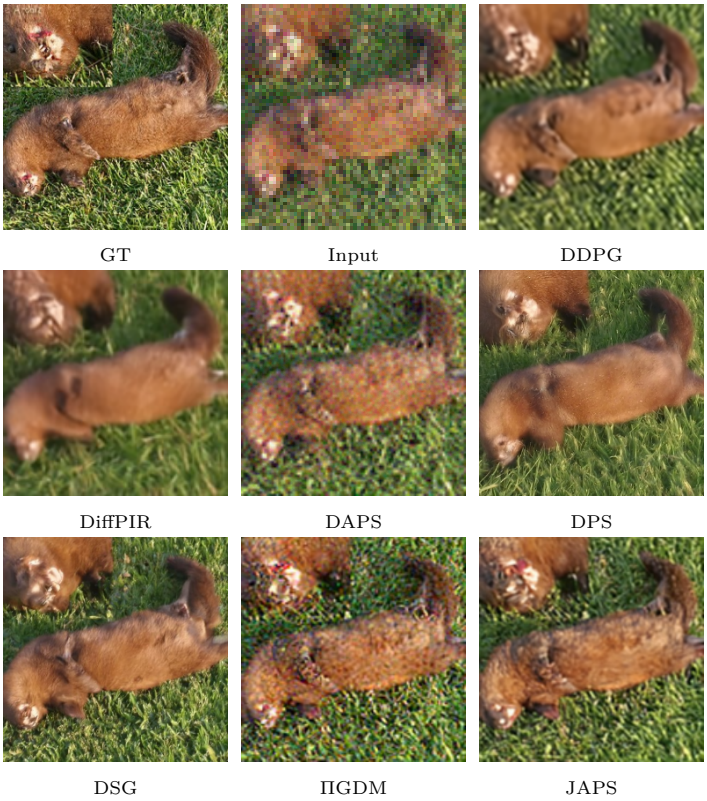


Figure 9: Super-Resolution results on ImageNet

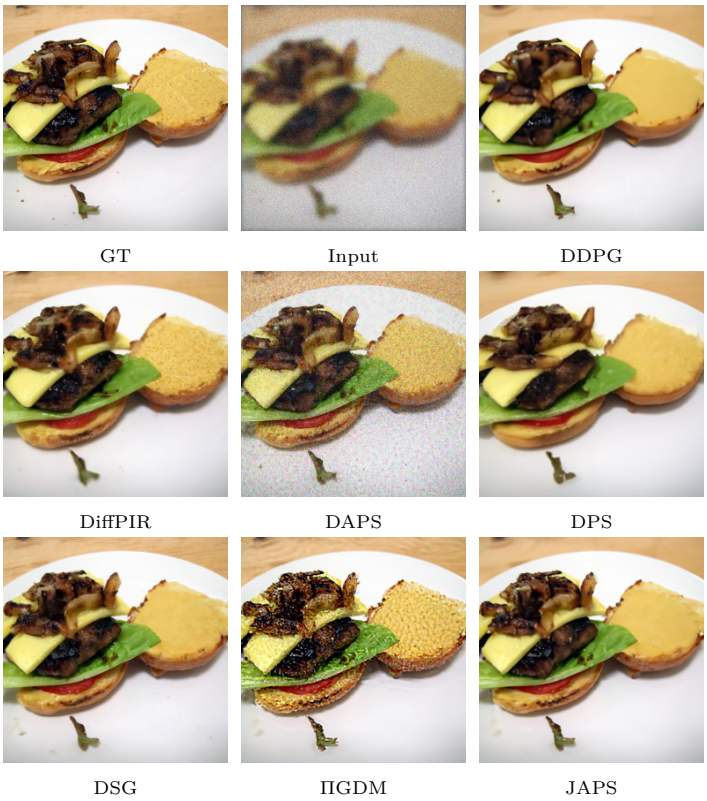


Figure 10: Gaussian Deblurring results on ImageNet

Figure 11: Motion Deblurring results on ImageNet

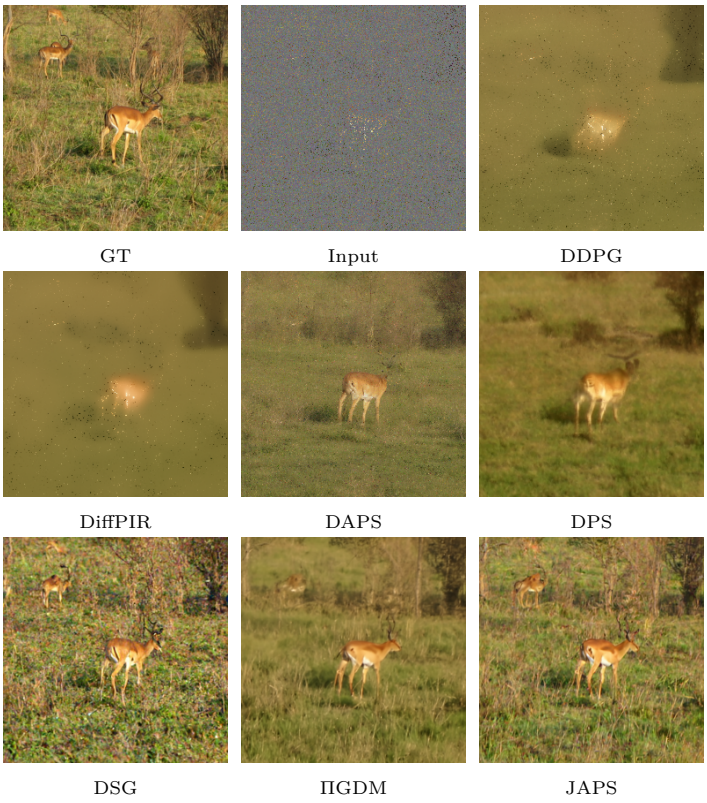


Figure 12: Random Inpainting results on ImageNet

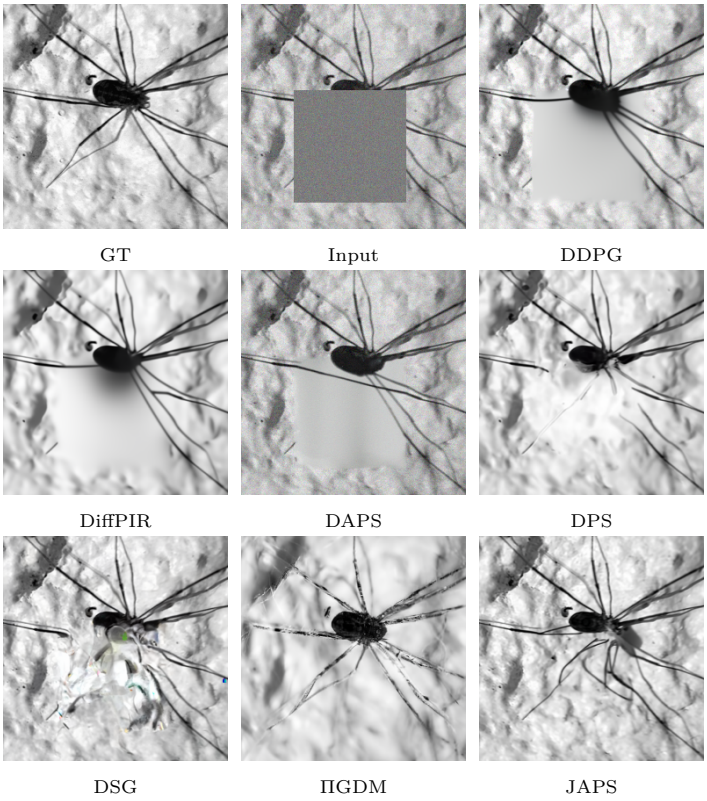


Figure 13: Box Inpainting results on ImageNet

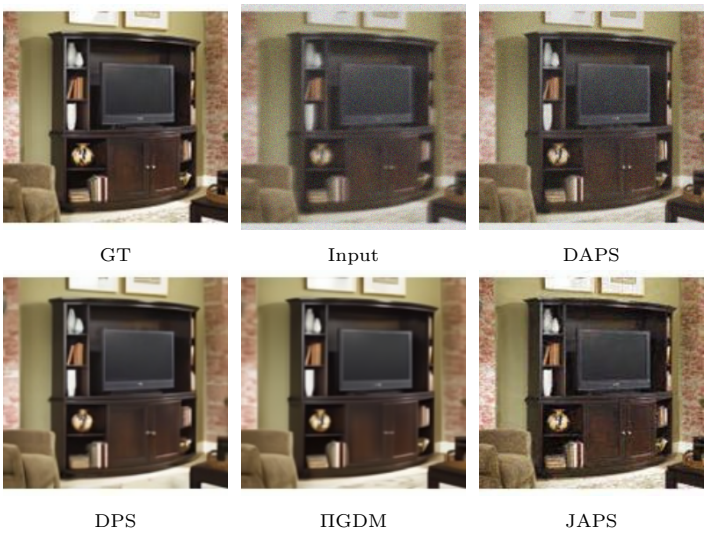


Figure 14: Nonlinear Blur results on ImageNet

Figure 15: HDR results on ImageNet

Figure 16: Super-Resolution results on CelebA-HQ

Figure 17: Gaussian Deblurring results on CelebA-HQ

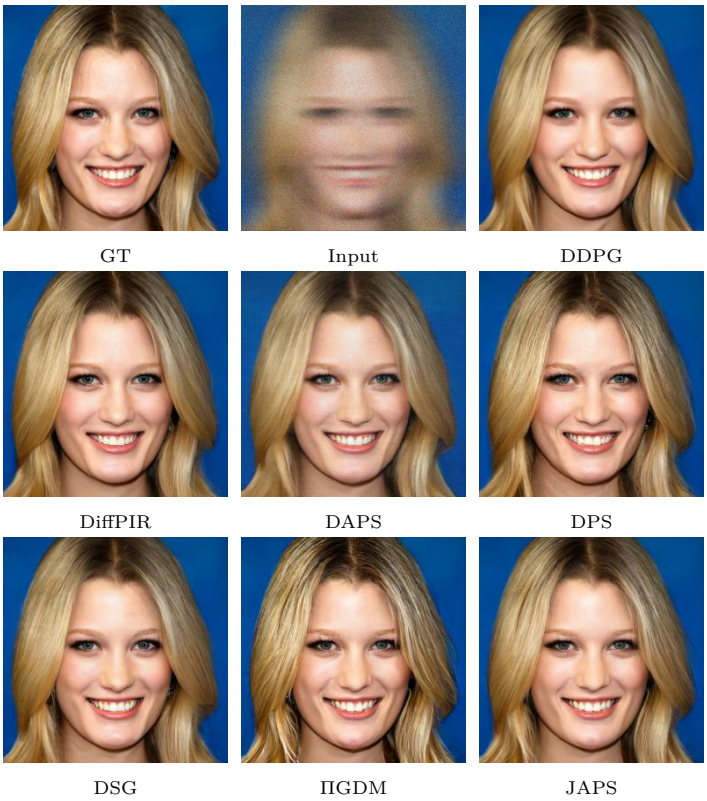


Figure 18: Motion Deblurring results on CelebA-HQ

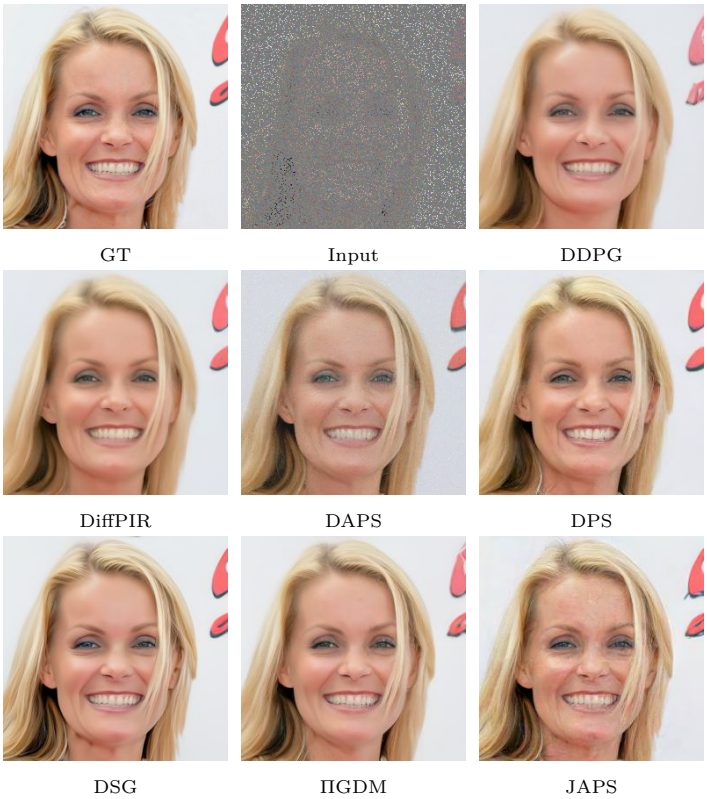


Figure 19: Random Inpainting results on CelebA-HQ

Figure 20: Box Inpainting results on CelebA-HQ

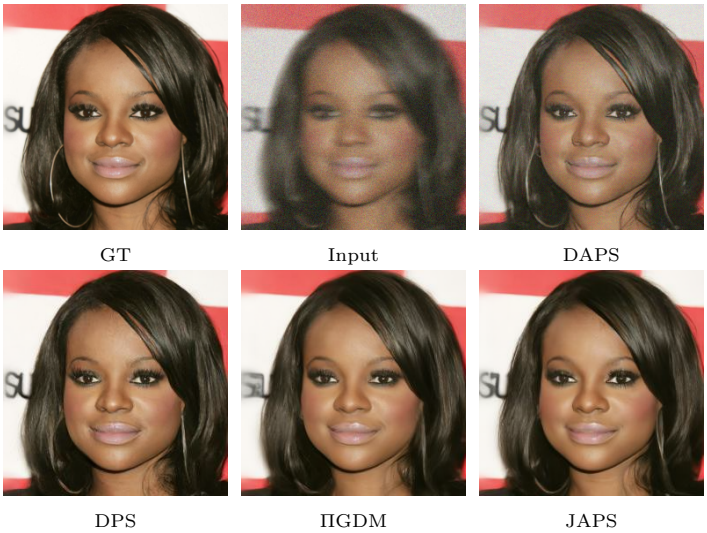


Figure 21: Nonlinear Blur results on CelebA-HQ

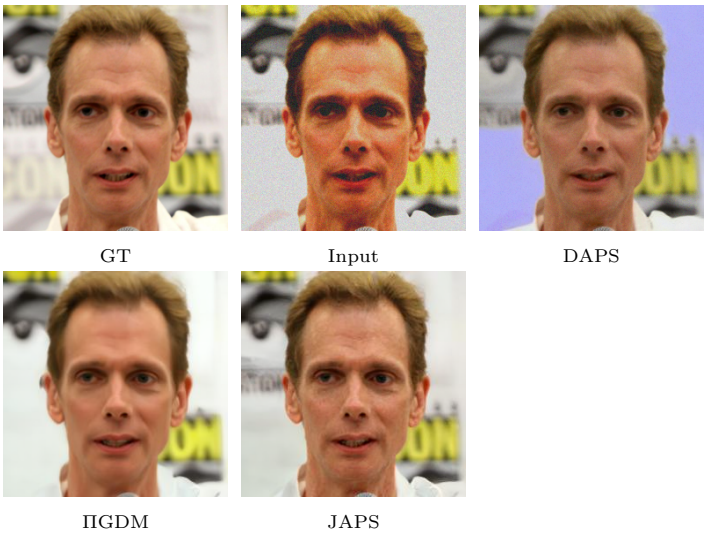


Figure 22: HDR results on CelebA-HQ