Image Denoising and Inverse Problem Solving with Pretrained Diffusion Models

Holland Ferguson

Abstract—Diffusion models have emerged as powerful generative priors for both image synthesis and solving inverse problems in computational imaging. In this work, we investigate the use of a pretrained Denoising Diffusion Probabilistic Model (DDPM) for single-step image denoising, unconditional face generation, and two representative linear inverse problems: inpainting and deconvolution. We implement and systematically compare three approaches for leveraging the diffusion prior in inverse problems which differ in how measurement information is incorporated during the reverse diffusion process. Our experiments on FFHQ-256 reveal that explicit measurement conditioning via likelihood gradients is critical for high-fidelity reconstruction: DPS achieves 37.18 dB PSNR on inpainting, a 14 dB improvement over SDEdit (23.15 dB) and a 3 dB improvement over ScoreALD (34.21 dB). We further show that DPS's gradient normalization provides substantial robustness to hyperparameter selection compared to ScoreALD's annealing schedule. Through ablation studies over key hyperparameters for each method, we provide practical guidance for applying diffusion-based posterior sampling to computational imaging tasks.

Index Terms—Computational Photography, Diffusion Models

1 Introduction

Recovering high-quality images from corrupted or incomplete observations is a central challenge in computational imaging. Inverse problems such as inpainting, deblurring, and super-resolution arise across scientific and consumer applications, yet they are fundamentally ill-posed: many plausible clean images are consistent with a given set of degraded measurements. Classical approaches rely on hand-crafted priors that impose generic smoothness assumptions and often fail to capture the rich statistical structure of natural images. End-to-end supervised methods can learn powerful mappings from degraded to clean images, but they require task-specific training data and must be retrained whenever the forward measurement model changes, limiting their flexibility.

Denoising diffusion probabilistic models (DDPMs) offer an appealing alternative: a single generative model, trained once on clean data, encodes a highly expressive prior over the image distribution that can be repurposed across arbitrary downstream inverse problems at test time. Through an iterative denoising process, diffusion models learn to reverse a fixed noising schedule, effectively parameterizing the score function of the data distribution at every noise level. This learned score provides a gradient signal that can guide sampling toward regions of high probability under the image prior, making diffusion models natural candidates for posterior sampling in inverse problems. Crucially, because the prior is decoupled from the measurement model, a single pretrained diffusion model can serve as a plug-and-play prior for inpainting, deconvolution, and other tasks without retraining.

In this work, we investigate a pretrained DDPM as both a generative model and an inverse problem solver. We implement and compare five approaches: (1) single-step denoising via Tweedie's formula, (2) unconditional generation through full DDPM sampling, (3) SDEdit, which solves inverse problems by partially noising the measurement and running unconditional reverse diffusion, and (4/5) two posterior sampling methods, ScoreALD and DPS, which incorporate measurement consistency through likelihood gradients at each reverse step. We evaluate all methods on FFHQ-256 face images, reporting PSNR and LPIPS for inpainting and deconvolution.

2 Related Work

The foundation of this work rests on two convergent lines of research in deep generative modeling. Ho et al. [1] introduced denoising diffusion probabilistic models (DDPMs), which learn to reverse a fixed Markov chain that progressively corrupts data with Gaussian noise. By training a neural network to predict the noise added at each step, DDPMs achieve high-fidelity image generation through an iterative refinement process over a thousand or more timesteps. In parallel, Song and Ermon [2] developed score-based generative models, which estimate the gradient of the log data density (the score function) at multiple noise scales and generate samples via annealed Langevin dynamics. Song et al. [3] later unified these perspectives by showing that both DDPMs and score-based models can be understood as discretizations of stochastic differential equations, establishing that noise prediction and score estimation are mathematically equivalent up to a scaling factor. This equivalence is practically important: the pretrained model used in our work predicts scores, but the same model can be interpreted as a noise predictor when implementing DDPM sampling.

Inverse problems in imaging have a long history rooted in classical signal processing and optimization. Compressed sensing [4] demonstrated that sparse signals can be recovered from far fewer measurements than traditional Nyquist sampling would require, using convex relaxations such as basis pursuit. More broadly, maximum a posteriori (MAP) estimation frames image reconstruction as an optimization

problem that balances a data fidelity term against a prior, with total variation [5] and wavelet sparsity serving as common regularizers. While effective for simple image models, these hand-crafted priors lack the expressiveness to capture the full complexity of natural image statistics, motivating the use of learned generative models as implicit priors.

The recognition that diffusion models encode powerful image priors has spurred several methods for posterior sampling in inverse problems. SDEdit [6] takes the most direct approach: it adds noise to the degraded observation to reach an intermediate point in the diffusion process, then runs the standard reverse process from that point. This avoids any modification to the sampling algorithm but conflates the prior and likelihood, offering only indirect control over measurement consistency. Jalal et al. [7] proposed ScoreALD, which interleaves score-based denoising updates with likelihood gradient steps derived from the forward measurement model, using annealed dynamics to traverse noise scales. Chung et al. [8] introduced DPS, which integrates a normalized likelihood gradient directly into each reverse diffusion step, approximating the posterior score via Bayes' rule. Unlike ScoreALD, DPS extends naturally to nonlinear forward models. This paper compares these three approaches using a shared pretrained model and evaluation protocol.

3 Proposed Method

3.1 3.1 Diffusion Model Background

We adopt the variance-preserving (VP) formulation of denoising diffusion probabilistic models introduced by Ho et al. [1]. The forward diffusion process defines a Markov chain that progressively corrupts a clean image $\mathbf{x}_0$ by adding Gaussian noise over $T = 1000$ timesteps. At each step t, the transition is given by

$$\mathbf{x}_t = \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathbf{z}_{t-1}, \quad \mathbf{z}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{1}$$

where $\{\beta_t\}_{t=1}^T$ is a linear noise schedule. By defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, the marginal distribution at any timestep t can be written in closed form as

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2}$$

This expression allows direct sampling of $\mathbf{x}_t$ from $\mathbf{x}_0$ without iterating through intermediate steps, which is essential for both the single-step denoising experiments and the partial-noising procedure used in SDEdit.

The pretrained model used in this work is a score-predicting network $\mathbf{s}_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$, trained on the FFHQ-256 dataset [8]. The score and noise-prediction formulations are related by

$$\mathbf{s}_\theta(\mathbf{x}_t, t) = -\frac{\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}} \tag{3}$$

so the two parameterizations are interchangeable up to a known scaling factor. Given a noisy observation $\mathbf{x}_t$ at any timestep, Tweedie's formula yields a single-step estimate of the clean image:

$$\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t + (1-\bar{\alpha}_t)\,\mathbf{s}_\theta(\mathbf{x}_t, t)\right). \tag{4}$$

This denoiser forms the core building block for all methods described below. Applying this formula once to a noisy image at a known timestep t constitutes single-step denoising, which we use to validate the pretrained model before proceeding to iterative generation and inverse problem solving.

3.2 Unconditional Generation

Unconditional image generation proceeds by reversing the forward diffusion chain. Starting from pure Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we iteratively apply the reverse transition for $t = T, \ldots, 1$. Using the $\hat{\mathbf{x}}_0$-based formulation, each reverse step first computes the Tweedie estimate $\hat{\mathbf{x}}_0$ and then obtains $\mathbf{x}_{t-1}$ via the posterior mean of $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \hat{\mathbf{x}}_0)$:

$$\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t + (1-\bar{\alpha}_t)\,\mathbf{s}_\theta(\mathbf{x}_t, t)\right), \tag{5}$$

$$\mathbf{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\,\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\,\hat{\mathbf{x}}_0 + \sqrt{1-\alpha_t}\,\mathbf{z} \tag{6}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $t > 1$ and $\mathbf{z} = \mathbf{0}$ for $t = 1$. The variance of the added noise is determined by the learned variance output of the model, which interpolates between the posterior variance bounds $\tilde{\beta}_t = \beta_t(1-\bar{\alpha}_{t-1})/(1-\bar{\alpha}_t)$ and β_t. Running this chain for all $T = 1000$ steps produces a sample from the learned image distribution.

3.3 Inverse Problems

We consider linear inverse problems of the form

$$\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \mathbf{n} \tag{7}$$

where $\mathbf{y} \in \mathbb{R}^m$ is the observed measurement, $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^m$ is a known linear forward operator, and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is additive Gaussian noise. We study two instantiations: inpainting, where $\mathcal{A}$ is a binary mask that zeros out a rectangular region of the image, and deconvolution, where $\mathcal{A}$ applies a Gaussian blur kernel. In both cases, the goal is to recover $\mathbf{x}_0$ from $\mathbf{y}$ by leveraging the pretrained diffusion model as an image prior. The three methods below differ in how they incorporate the measurement constraint $\mathbf{y} = \mathcal{A}(\mathbf{x}_0)$ into the reverse diffusion sampling process.

3.4 SDEdit

SDEdit [6] is the simplest approach we consider. Rather than starting the reverse chain from pure noise at $t = T$, SDEdit begins from an intermediate timestep t^* by applying the forward process directly to the measurement:

$$\mathbf{x}_{t^*} = \sqrt{\bar{\alpha}_{t^*}}\,\mathbf{y} + \sqrt{1-\bar{\alpha}_{t^*}}\,\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{8}$$

The standard unconditional reverse chain (Section 3.2) is then run from t^* down to $t = 0$. The hyperparameter t^* controls a fundamental tradeoff: large values of t^* add more noise, giving the model greater freedom to hallucinate missing content and produce realistic images, but at the cost of reduced fidelity to the observed measurements. Small values of t^* preserve more of the measurement information but may fail to correct severe degradations, since the model has fewer denoising steps to work with. Notably, SDEdit does not modify the reverse sampling algorithm itself. Rather,

measuremnt consistency is imposed only through initialization, making the method straightforward to implement but limited in its ability to enforce data fidelity throughout the sampling process.

3.5 ScoreALD

ScoreALD [7] addresses the limitation of SDEdit by explicitly incorporating the likelihood $p(\mathbf{y} \mid \mathbf{x}_t)$ into the reverse sampling dynamics. The algorithm follows the standard reverse diffusion chain but appends a likelihood gradient step after each DDPM update. Concretely, at each timestep t, we first compute the unconditional reverse step to obtain $\mathbf{x}_{t-1}$, then apply the correction:

$$\mathbf{x}_{t-1} \leftarrow \mathbf{x}_{t-1} - \frac{1}{2(\sigma^2 + \gamma_t^2)} \nabla_{\mathbf{x}_t} \left\| \mathcal{A}(\mathbf{x}_t) - \mathbf{y} \right\|^2 \tag{9}$$

where σ^2 is the measurement noise variance and γ_t^2 is an annealed regularization term. The annealing schedule ties γ_t to the diffusion noise level, ensuring that the likelihood gradient is scaled appropriately: at early timesteps (large t), the diffusion noise dominates and the gradient step is small, while at later timesteps (small t), the correction becomes stronger as the signal-to-noise ratio improves. The gradient $\nabla_{\mathbf{x}_t} \|\mathcal{A}(\mathbf{x}_t) - \mathbf{y}\|^2$ is computed via automatic differentiation through the forward operator. Because ScoreALD operates on $\mathbf{x}_t$ directly rather than through the Tweedie estimate $\hat{\mathbf{x}}_0$, the gradient magnitude can vary substantially across timesteps, which in practice requires careful tuning of the annealing schedule.

3.6 DPS

DPS [8] refines the posterior sampling strategy by computing the likelihood gradient through the Tweedie estimate $\hat{\mathbf{x}}_0$ rather than $\mathbf{x}_t$, and by normalizing the gradient to stabilize updates across timesteps. At each step, the algorithm first performs the standard unconditional reverse update to obtain a preliminary $\mathbf{x}'_{t-1}$, then applies:

$$\mathbf{x}_{t-1} = \mathbf{x}'_{t-1} - \zeta_t \nabla_{\mathbf{x}_t} \left\| \mathcal{A}(\hat{\mathbf{x}}_0) - \mathbf{y} \right\|^2, \tag{10}$$

where the step size is normalized as

$$\zeta_t = \frac{\zeta}{\left\| \nabla_{\mathbf{x}_t} \left\| \mathcal{A}(\hat{\mathbf{x}}_0) - \mathbf{y} \right\|^2 \right\|}. \tag{11}$$

The scalar ζ is a global hyperparameter controlling the overall strength of the data fidelity correction. The key insight of DPS is that normalizing by the gradient norm ensures a consistent step size regardless of the timestep-dependent scaling of the Tweedie estimate. This addresses a practical difficulty of ScoreALD, where the unnormalized gradient magnitude can change by orders of magnitude across the diffusion trajectory, making a single annealing schedule difficult to tune. Also, since DPS only requires differentiating through $\hat{\mathbf{x}}_0 = f(\mathbf{x}_t)$ and the forward operator $\mathcal{A}$, it extends naturally to nonlinear measurement models, whereas ScoreALD's formulation assumes linearity.

TABLE 1
Single-step denoising results at different noise levels. The model performs well even on out-of-distribution content (parrots), with degradation at higher noise.

Image	Timestep	PSNR	LPIPS
Parrots	$t = 100$	31.25	0.0605
	$t = 250$	26.29	0.1609
	$t = 500$	21.13	0.3900

Fig. 1. Example Denoising Result

4 Experimental Results

4.1 Single-Step Denoising

We first evaluate the pretrained diffusion model as a single-step Gaussian denoiser. A clean test image is corrupted using the forward process at three noise levels ($t \in \{100, 250, 500\}$) and apply Tweedie's formula to obtain the denoised estimate $\hat{\mathbf{x}}_0$.

Table 1 reports quantitative results on two test images: an out-of-distribution natural image (parrots) and an in-distribution face image from FFHQ. At low noise ($t = 100$), the model achieves strong reconstruction with PSNR above 31 dB and LPIPS below 0.09 for both images. Performance degrades as noise increases: at $t = 250$, PSNR drops to approximately 26–27 dB, and at $t = 500$, to 21–22 dB. Qualitatively, the denoised outputs at $t = 100$ are nearly indistinguishable from the originals, while at $t = 500$ the model produces plausible but smoothed reconstructions that lose fine detail (see Figure 1).

Interestingly, the model performs comparably on the out-of-distribution parrot image and the in-distribution face image, suggesting that the learned denoiser captures general natural image statistics beyond just faces. However, at higher noise levels, the face image reconstructions exhibit slightly sharper features, consistent with the FFHQ training distribution bias.

4.2 Unconditional Image Generation

We generate face images by running the full 1000-step DDPM reverse process starting from pure Gaussian noise. Figure 2 shows three independently generated samples. All outputs exhibit sharp, coherent facial features with realistic skin textures, hair, and lighting, confirming that the iterative denoising procedure successfully transforms unstructured noise into high-quality samples from the learned distribution. The diversity of the generated faces, which vary in age, gender, and expression, demonstrate that the model has captured the breadth of the FFHQ training distribution rather than collapsing to a narrow subset. Since unconditional generation has no ground-truth reference, quality is evaluated purely through visual inspection.

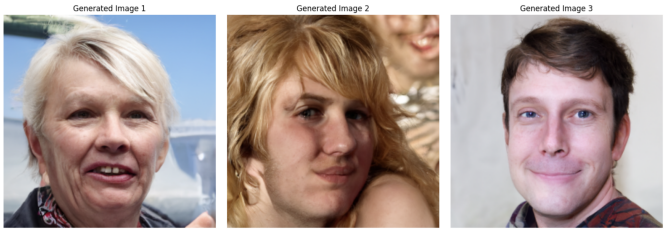

Fig. 2. Unconditionally Generated Faces

TABLE 2
SDEdit results for inpainting and deconvolution at different starting timesteps. Lower t_{start} preserves more measurement fidelity but offers less correction.

Task	t_{start}	PSNR	LPIPS
Inpainting	250	23.15	0.1413
	500	20.51	0.2079
	750	14.01	0.3742
Deconvolution	250	23.95	0.1944
	500	20.36	0.2109
	750	14.93	0.3217

4.3 Inverse Problems: SDEdit

We evaluate SDEdit on both inpainting (box mask) and deconvolution, varying the starting timestep $t_{\text{start}} \in \{250, 500, 750\}$. Table 2 summarizes the quantitative results.

For inpainting, $t_{\text{start}} = 250$ yields the best PSNR (23.15 dB) and LPIPS (0.1413), as the low noise level preserves the unmasked regions while providing just enough stochasticity to fill in the masked area. Increasing to $t_{\text{start}} = 500$ degrades both metrics (PSNR 20.51 dB, LPIPS 0.2079), and $t_{\text{start}} = 750$ produces poor reconstructions (PSNR 14.01 dB, LPIPS 0.3742) where the model essentially generates a new face with little resemblance to the original.

Deconvolution follows a similar pattern: $t_{\text{start}} = 250$ achieves the best results (PSNR 23.95 dB, LPIPS 0.1944), with performance degrading monotonically as t_{start} increases. Qualitatively, low t_{start} values produce outputs that retain the identity of the original subject but may exhibit residual blur, while high values yield sharp but identity-inconsistent faces (see Figure 4).

This behavior highlights a fundamental limitation of SDEdit: since no explicit measurement conditioning is applied during reverse sampling, the method relies entirely on the partial noising to encode measurement information. At high t_{start}, too much noise is added and the measurement signal is effectively destroyed.

4.4 Inverse Problems: ScoreALD

ScoreALD incorporates measurement information through annealed likelihood gradients at every reverse step. Table 3 reports results using the default annealing range $(10, 15)$, and Table 4 presents the full ablation over annealing schedules.

With the default schedule, ScoreALD achieves PSNR of 32.70 dB and LPIPS of 0.0758 on box inpainting—a substantial improvement over SDEdit's best result (23.15 dB / 0.1413). The explicit likelihood conditioning enables ScoreALD to produce reconstructions that are both perceptually sharp and highly consistent with the observed measurements.

The annealing ablation reveals a clear trend: weaker annealing factors yield better reconstruction metrics. The range $(5, 10)$ achieves the best inpainting performance (PSNR 34.21 dB, LPIPS 0.0370), while more aggressive ranges progressively degrade quality. At $(20, 30)$, the inpainting PSNR drops to 24.35 dB with LPIPS of 0.1257. This effect is even more pronounced for deconvolution, where the range $(20, 30)$ causes catastrophic degradation (PSNR 12.68 dB, LPIPS 0.6938), producing heavily distorted outputs with visible artifacts. This sensitivity to the annealing schedule is a practical limitation of ScoreALD: too-strong conditioning overwhelms the prior, while too-weak conditioning fails to enforce measurement consistency.

Fig. 3. SDEdit Inpainting Results

Fig. 4. SDEdit Deconvolution Results

TABLE 3
ScoreALD results with default annealing range $(10, 15)$.

Task	PSNR	LPIPS
Inpainting (box)	32.70	0.0758

4.5 Inverse Problems: DPS

DPS replaces the annealing schedule with a normalized gradient update controlled by a single scale parameter ζ.

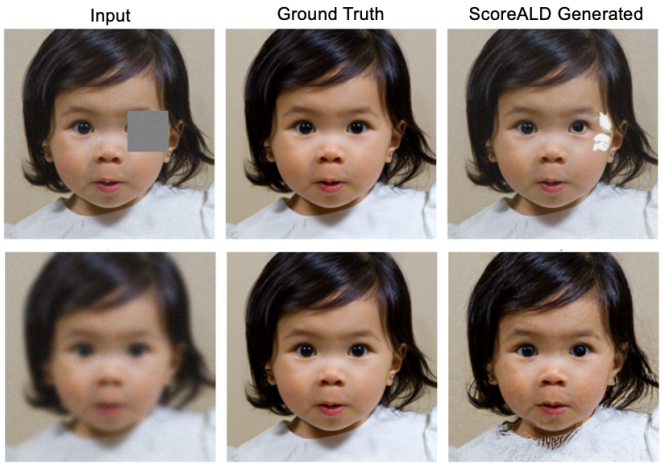

Fig. 5. ScoreALD Results

TABLE 4
ScoreALD ablation over annealing schedules. Weaker annealing factors consistently yield better reconstruction, with aggressive schedules causing severe degradation, especially for deconvolution.

Task	Anneal Range	PSNR	LPIPS
Inpainting	$(5, 10)$	34.21	0.0370
	$(10, 15)$	31.02	0.0806
	$(15, 20)$	26.64	0.0972
	$(20, 30)$	24.35	0.1257
Deconvolution	$(5, 10)$	26.44	0.1006
	$(10, 15)$	25.19	0.1484
	$(15, 20)$	17.63	0.6244
	$(20, 30)$	12.68	0.6938

TABLE 5
DPS result with default $\zeta = 0.3$ on deconvolution.

Task	PSNR	LPIPS
Deconvolution	28.64	0.0573

Table 5 reports the result with the default deconvolution setting, and Table 6 presents the full scale ablation.

With $\zeta = 0.3$ on deconvolution, DPS achieves PSNR of 28.64 dB and LPIPS of 0.0573, outperforming ScoreALD's best deconvolution result (26.44 dB / 0.1006) by a significant margin. The gradient normalization in DPS provides more stable updates, making it particularly effective for deconvolution where the measurement residual magnitudes can vary widely across timesteps.

The scale ablation reveals task-dependent optimal values. For inpainting, $\zeta = 0.5$ achieves the best performance (PSNR 37.18 dB, LPIPS 0.0101), representing a dramatic improvement over both SDEdit and ScoreALD. Both $\zeta = 0.3$ (PSNR 34.63 dB) and $\zeta = 1.0$ (PSNR 34.75 dB) also perform well, while $\zeta = 0.1$ underperforms (PSNR 26.79 dB) due to insufficient measurement conditioning. For deconvolution, the metrics are more stable across scale values: $\zeta \in \{0.3, 0.5\}$ achieve similar PSNR (≈28.3 dB) and LPIPS (≈0.06), while $\zeta = 1.0$ shows slightly higher perceptual distortion (LPIPS 0.1044) despite comparable PSNR.

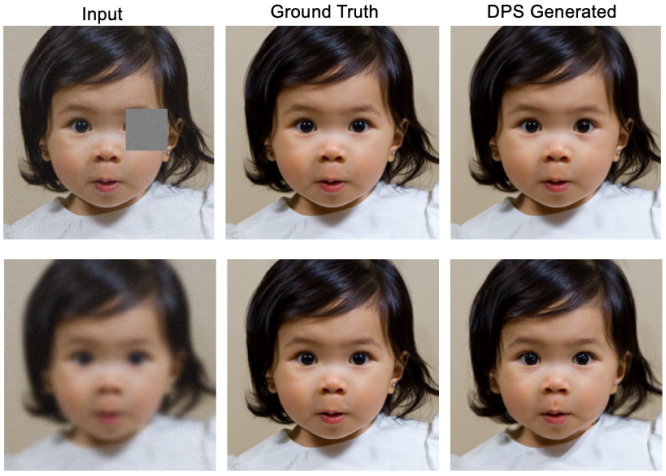

Fig. 6. DPS Results

TABLE 6
DPS ablation over scale parameter ζ. The optimal scale is task-dependent, with inpainting favoring $\zeta = 0.5$ and deconvolution performing well across $\zeta \in \{0.3, 0.5\}$.

Task	ζ	PSNR	LPIPS
Inpainting	0.1	26.79	0.1164
	0.3	34.63	0.0249
	0.5	37.18	0.0101
	1.0	34.75	0.0234
Deconvolution	0.1	24.87	0.1165
	0.3	28.36	0.0616
	0.5	28.33	0.0613
	1.0	28.25	0.1044

TABLE 7
Best results across methods for each task. DPS consistently achieves the highest reconstruction quality, followed by ScoreALD, with SDEdit trailing significantly.

Task	Method	PSNR	LPIPS
Inpainting	SDEdit ($t_{\text{start}} = 250$)	23.15	0.1413
	ScoreALD (anneal 5–10)	34.21	0.0370
	DPS ($\zeta = 0.5$)	**37.18**	**0.0101**
Deconvolution	SDEdit ($t_{\text{start}} = 250$)	23.95	0.1944
	ScoreALD (anneal 5–10)	26.44	0.1006
	DPS ($\zeta = 0.3$)	**28.36**	**0.0616**

4.6 Comparative Analysis

Table 7 summarizes the best results achieved by each method on both tasks, enabling a direct comparison.

For inpainting, the posterior sampling methods dramatically outperform SDEdit: DPS achieves 37.18 dB PSNR (with $\zeta = 0.5$), compared to ScoreALD's 34.21 dB and SDEdit's 23.15 dB. The gap is even more striking in LPIPS, where DPS achieves 0.0101 versus SDEdit's 0.1413 – an order-of-magnitude improvement in perceptual quality. For deconvolution, DPS again leads with 28.36 dB PSNR and 0.0616 LPIPS, compared to ScoreALD's 26.44 dB / 0.1006 and SDEdit's 23.95 dB / 0.1944.

Between the two posterior sampling methods, DPS consistently outperforms ScoreALD on both tasks and both metrics. This advantage stems from the gradient normalization, which decouples the update magnitude from the measurement residual and provides more uniform conditioning throughout the reverse process. Additionally, DPS requires tuning only a single scalar ζ, whereas ScoreALD's performance is sensitive to both the lower and upper bounds of the annealing schedule.

5 Conclusion

We presented a systematic comparison of diffusion-based methods for solving linear inverse problems using a single pretrained DDPM. Our results demonstrate that measurement conditioning during reverse diffusion is essential: SDEdit, which relies solely on partial noising, achieves 23.15 dB PSNR on inpainting, while ScoreALD and DPS improve this to 34.21 dB and 37.18 dB respectively by incorporating likelihood gradients at each sampling step. DPS consistently outperforms ScoreALD across both inpainting and deconvolution, owing to its normalized gradient updates that decouple conditioning strength from the measurement residual magnitude. These findings highlight that simple algorithmic choices – particularly gradient normalization– can

yield significant and consistent improvements in diffusion-based posterior sampling for computational imaging.

6 Appendix

6.1 Proof 1

Given the forward noise model from step $t-1$ to t:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\mathbf{z}_{t-1} \tag{12}$$

And the definitions:

$$\alpha_t = 1 - \beta_t \tag{13}$$

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i \tag{14}$$

First, we substitute α_t into equation 1:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \tag{15}$$

Next, we express $\mathbf{x}_{t-1}$ in terms of $\mathbf{x}_{t-2}$:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\mathbf{z}_{t-2} \tag{16}$$

Substitute $\mathbf{x}_{t-1}$ back into the equation for $\mathbf{x}_t$:

$$\mathbf{x}_t = \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\mathbf{z}_{t-2}\right) + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \tag{17}$$

$$= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}\mathbf{z}_{t-2} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \tag{18}$$

Since $\mathbf{z}_{t-2}$ and $\mathbf{z}_{t-1}$ are indep RVs drawn from a standard normal distribution $\mathcal{N}(0, I)$, any linear combination of them is also normally distributed. The sum of their variances is:

$$\sigma^2 = \left(\sqrt{\alpha_t(1-\alpha_{t-1})}\right)^2 + \left(\sqrt{1-\alpha_t}\right)^2 \tag{19}$$

$$= \alpha_t(1-\alpha_{t-1}) + 1 - \alpha_t \tag{20}$$

$$= \alpha_t - \alpha_t\alpha_{t-1} + 1 - \alpha_t \tag{21}$$

$$= 1 - \alpha_t\alpha_{t-1} \tag{22}$$

We can now merge the two noise terms into a single new standard Gaussian variable $\mathbf{z}' \sim \mathcal{N}(0, I)$:

$$\mathbf{x}_t = \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\mathbf{z}' \tag{23}$$

By recursively applying this substitution all the way down to $t = 0$, the coefficient for $\mathbf{x}_0$ becomes the square root of the product of all α terms:

$$\sqrt{\alpha_t\alpha_{t-1}\dots\alpha_1} = \sqrt{\prod_{i=1}^{t}\alpha_i} = \sqrt{\bar{\alpha}_t} \tag{24}$$

Following the same variance accumulation pattern, the total combined variance of the noise becomes:

$$1 - (\alpha_t\alpha_{t-1}\dots\alpha_1) = 1 - \prod_{i=1}^{t}\alpha_i = 1 - \bar{\alpha}_t \tag{25}$$

Therefore, we can express $\mathbf{x}_t$ conditioned solely on $\mathbf{x}_0$ and a single noise term $\mathbf{z} \sim \mathcal{N}(0, I)$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z} \tag{26}$$

6.2 Proof 1

We start with the given expression for $\mathbf{x}_{t-1}$:

$$\mathbf{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0 \tag{27}$$

Substitute the definition of $\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t))$ into the equation:

$$\mathbf{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\left[\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t)\right. \tag{28}$$

Using the identity $\bar{\alpha}_t = \alpha_t\bar{\alpha}_{t-1}$, we can rewrite the denominator of the second term as $\sqrt{\bar{\alpha}_t} = \sqrt{\alpha_t}\sqrt{\bar{\alpha}_{t-1}}$:

$$\mathbf{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}\sqrt{\bar{\alpha}_{t-1}}}(\mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t)) \tag{29}$$

$$= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}(\mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t)) \tag{30}$$

Distribute the fraction into the parentheses:

$$\mathbf{x}_{t-1} = \left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} + \frac{1-\alpha_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}\right)\mathbf{x}_t + \frac{(1-\alpha_t)(1-\bar{\alpha}_t)}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}\mathbf{s}_\theta(\mathbf{x}_t, t \tag{31}$$

Cancel the common $(1-\bar{\alpha}_t)$ terms to simplify the coefficient of $\mathbf{s}_\theta(\mathbf{x}_t, t)$:

$$\frac{(1-\alpha_t)(1-\bar{\alpha}_t)}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)} = \frac{1-\alpha_t}{\sqrt{\alpha_t}} \tag{32}$$

Next, find a common denominator for the coefficient of $\mathbf{x}_t$:

$$\frac{\alpha_t(1-\bar{\alpha}_{t-1}) + (1-\alpha_t)}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)} = \frac{\alpha_t - \alpha_t\bar{\alpha}_{t-1} + 1 - \alpha_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)} \tag{33}$$

$$= \frac{1-\alpha_t\bar{\alpha}_{t-1}}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)} \tag{34}$$

Applying the identity $\alpha_t\bar{\alpha}_{t-1} = \bar{\alpha}_t$ again, the numerator simplifies to $1-\bar{\alpha}_t$, leaving:

$$\frac{1-\bar{\alpha}_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)} = \frac{1}{\sqrt{\alpha_t}} \tag{35}$$

Putting both simplified coefficients back into the full equation:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\mathbf{s}_\theta(\mathbf{x}_t, t) \tag{36}$$

Finally, factoring out $\frac{1}{\sqrt{\alpha_t}}$ yields the target equivalent form:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t + (1-\alpha_t)\mathbf{s}_\theta(\mathbf{x}_t, t)) \tag{37}$$

6.3 Proof 3

Based on Tweedie's formula for the variance-preserving (VP) forward diffusion process:

$$\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t)) \tag{38}$$

Rearrange this to solve for the score-predicting network $\mathbf{s}_\theta(\mathbf{x}_t, t)$:

$$\sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0 = \mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t) \tag{39}$$

$$\mathbf{s}_\theta(\mathbf{x}_t, t) = \frac{\sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0 - \mathbf{x}_t}{1-\bar{\alpha}_t} \tag{40}$$

From the one-step VP forward diffusion process, we have the following definition:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \tag{41}$$

By rearranging this to solve for $\mathbf{x}_0$ and substituting our noise-predicting network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, we can express the predicted clean image $\hat{\mathbf{x}}_0$ as:

$$\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}} \tag{42}$$

Substitute this expression for $\hat{\mathbf{x}}_0$ back into the rearranged Tweedie's formula:

$$\mathbf{s}_\theta(\mathbf{x}_t, t) = \frac{\sqrt{\bar{\alpha}_t}\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)}{\sqrt{\bar{\alpha}_t}}\right) - \mathbf{x}_t}{1-\bar{\alpha}_t} \tag{43}$$

$$= \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t}{1-\bar{\alpha}_t} \tag{44}$$

$$= \frac{-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{1-\bar{\alpha}_t} \tag{45}$$

$$= -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \tag{46}$$

subst into score-based reverse diffusion step from the left side of the target equation:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t + (1-\alpha_t)\mathbf{s}_\theta(\mathbf{x}_t, t)) \tag{47}$$

$$= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t + (1-\alpha_t)\left(-\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)\right) \tag{48}$$

$$= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) \tag{49}$$

REFERENCES

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in NeurIPS, 2020.

[2] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in NeurIPS, 2019.

[3] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in ICLR, 2021.

[4] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," IEEE Trans. Inf. Theory, vol. 52, no. 2, pp. 489–509, 2006.

[5] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Physica D, vol. 60, no. 1–4, pp. 259–268, 1992.

[6] C. Meng et al., "SDEdit: Guided image synthesis and editing with stochastic differential equations," in ICLR, 2022.

[7] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, "Robust compressed sensing MRI with deep generative priors," in NeurIPS, 2021.

[8] H. Chung, J. Kim, M. T. McCann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in ICLR, 2023.