Geometry-Correct Diffusion Posterior Sampling with Denoiser-Pullback Curvature Guidance and Manifold-Aligned Damping

Anonymous Authors[1]

Abstract

Diffusion posterior sampling conditions diffusion priors on measurements, but data-consistency updates are typically scaled by hand-tuned guidance weights and can destabilize sampling under stiff, operator-dependent curvature. We replace scalar guidance with a per-noise-level damped Gauss–Newton correction computed in diffusion-state coordinates. The correction pulls likelihood gradients back through the denoiser, uses a one-sided curvature model that avoids forward denoiser Jacobians, and applies diffusion-calibrated rank-one damping aligned with the denoiser residual. Each correction is solved with matrix-free GMRES using automatic differentiation, and sampling proceeds with a variance-preserving Langevin transition with a closed-form drift/noise split. Aside from compute-budget choices (T diffusion steps and K Krylov iterations), the method has a single damping hyperparameter (λ_{id}), kept nearly unchanged across tasks. On FFHQ and ImageNet across inverse problems, it achieves competitive PSNR/SSIM/LPIPS while running markedly faster than most of the compared baselines; on accelerated MRI reconstruction, it achieves the best PSNR/SSIM among the compared baselines.

1. Introduction

Inverse problems aim to infer an unknown signal from indirect and noisy measurements. We consider the standard additive Gaussian model

$$\mathbf{y} = A(\mathbf{x}_0) + \mathbf{n}, \qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}), \tag{1}$$

where A is a (possibly nonlinear) forward operator and $\mathbf{x}_0$ is the unknown clean image. Because the mapping $\mathbf{x}_0 \mapsto \mathbf{y}$ is typically ill-posed, many $\mathbf{x}_0$ can explain the same observation (Bertero & Boccacci, 1998; Calvetti & Somersalo, 2007). In such settings, posterior sampling can be more informative than returning a single reconstruction: it generates multiple plausible explanations of the data and supports principled uncertainty quantification (Stuart, 2010; Dashti & Stuart, 2015).

Pretrained diffusion models are strong natural-image priors and are widely used in training-free inverse problem solvers (Feng et al., 2023). They generate samples by progressively denoising noisy states (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019; Song et al., 2021b), and EDM provides a practical noise-conditioned denoiser interface and schedule (Karras et al., 2022). Because runtime is dominated by denoiser calls, acceleration (e.g., distillation) is relevant under fixed compute budgets (Salimans & Ho, 2022).

A widely used conditional strategy is to run a diffusion sampler while injecting likelihood-driven data-consistency updates, as in diffusion posterior sampling (DPS) (Chung et al., 2023a). Under (1), the negative log-likelihood is $\Phi(\mathbf{x}) = \frac{1}{2\sigma_n^2}\|A(\mathbf{x}) - \mathbf{y}\|_2^2$. Interleaving "prior" diffusion steps with likelihood corrections is conceptually straightforward, but the likelihood correction is often fragile: it must be strong enough to reduce measurement error yet small enough to avoid pushing iterates into low-density regions of the diffusion prior.

Two coupled issues make this correction hard to calibrate across operators and noise levels. First, the local geometry of the likelihood can be strongly anisotropic: a single scalar step size must be small enough to remain stable in stiff directions, which can slow progress in flatter directions (especially for nonlinear A or latent parameterizations). Second, measurement consistency is defined on a clean variable (the input of A), while the sampler state is a noisy diffusion variable. In practice, one evaluates the measurement residual on the denoised prediction produced at each noise level, so the effective mismatch depends on the composition of the forward model with the denoiser. As a result, clean-space likelihood information must be mapped back to the diffusion-state update; ignoring this mapping can mis-scale directions when the denoiser is anisotropic.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

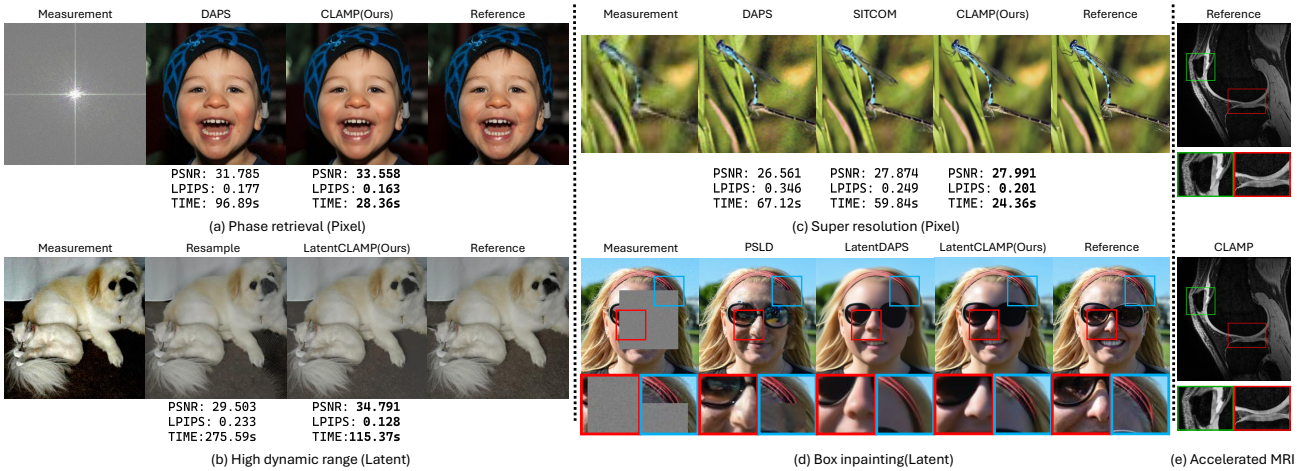

Figure 1. **Qualitative results on diverse inverse problems.** (a–d) Qualitative comparisons on FFHQ and ImageNet at 256×256, covering both pixel- and latent-space variants of CLAMP and baseline methods across a range of general inverse problems. For each task, we report the best competing results for a representative visual comparison. (e) Visualization of CLAMP on accelerated MRI reconstruction. Overall, CLAMP produces high-fidelity, visually plausible reconstructions and remains effective across pixel/latent settings, linear/nonlinear operators, and both natural and medical imaging domains.

Existing work mitigates these failure modes in complementary ways, including trajectory modifications (e.g., DAPS (Zhang et al., 2025)), per-step optimization (e.g., SITCOM (Alkhouri et al., 2025)), stabilization via preconditioning (Garber & Tirer, 2024) or projections (Chung et al., 2022), diffusion-bridge constructions (Chung et al., 2023b), and likelihood-update redesigns (Yismaw et al., 2025; Graikos et al., 2025; Hamidi et al., 2025). Despite this progress, many practical samplers still rely on scalar guidance schedules, and they do not explicitly combine coordinate-correct likelihood geometry with direction-wise curvature scaling and diffusion-calibrated step control in a single per-noise-level correction.

This paper introduces CLAMP (**C**urvature-aware **L**angevin with **A**ligned **M**anifold **P**ullback), a diffusion posterior sampler that directly targets the two issues above. To handle anisotropic likelihood geometry, CLAMP replaces scalar guidance with a curvature-scaled correction based on a damped Gauss–Newton (Fisher) quadratic model of the measurement residual at each noise level. To ensure coordinate correctness, the correction is computed in diffusion-state coordinates by pulling likelihood sensitivity back through the denoiser, rather than applying clean-space derivatives directly. For robustness and efficiency, we use a one-sided curvature approximation that avoids forward denoiser-Jacobian products inside curvature terms while retaining the pullback that maps measurement sensitivity into diffusion-state updates. Step sizes are stabilized via a trust-region/Levenberg–Marquardt viewpoint (Nocedal & Wright, 2006; Conn et al., 2000): we regularize the correction in a prior-aligned geometry and calibrate its strength to the diffusion noise level, avoiding per-task step-size schedules. The resulting update is computed with a fixed, lightweight matrix-free solve using Jacobian actions, yielding a favorable accuracy–runtime trade-off. After the correction, CLAMP advances the diffusion schedule with a variance-preserving Langevin-style step with a closed-form drift/noise split, avoiding an additional per-step stochasticity parameter. The same construction applies in latent space by composing the forward model with a decoder, replacing A by $A \circ D$.

Our contributions are:

- **Curvature-scaled, coordinate-correct likelihood corrections.** We formulate per-noise-level data-consistency updates in diffusion-state coordinates using denoiser pullback geometry and a damped Gauss–Newton curvature model, addressing both anisotropic stiffness and coordinate mismatch.
- **Diffusion-calibrated, prior-aligned anisotropic step control.** We introduce a rank-one damping metric aligned with the denoiser residual direction and a schedule-calibrated regularization rule that stabilizes likelihood corrections across noise levels without an explicit guidance schedule.
- **Fast matrix-free implementation in pixel and latent spaces.** We solve each correction with a fixed-budget Krylov method using only Jacobian actions and evaluate CLAMP on natural-image and medical-imaging inverse problems, demonstrating improved reconstruction quality and sampling efficiency over representative baselines.

Section 2 reviews diffusion denoisers, damped Gauss–Newton, and matrix-free Krylov solvers. Section 3 derives

CLAMP, and Section 4 reports quantitative results and ablations on FFHQ/ImageNet inverse problems and MRI reconstruction; Section 5 concludes.

2. Background

This section summarizes the technical prerequisites for CLAMP: (i) diffusion denoisers and noise schedules, (ii) Gauss–Newton (GN) curvature with damping via local quadratic surrogates, and (iii) matrix-free Krylov solvers based on Jacobian actions.

2.1. Diffusion denoisers and noise schedules

Diffusion models generate samples by iteratively denoising a sequence of noisy states (Ho et al., 2020; Song et al., 2021b). We work with a decreasing noise schedule $\{\sigma_t\}_{t=T}^{0}$ and an EDM-style denoiser interface (Karras et al., 2022):

$$\hat{\mathbf{x}}_{0|t} = F_\theta(\mathbf{x}_t; \sigma_t), \quad \epsilon_\theta(\mathbf{x}_t, \sigma_t) := \frac{\mathbf{x}_t - \hat{\mathbf{x}}_{0|t}}{\sigma_t}. \tag{2}$$

Here $\mathbf{x}_t$ is the diffusion-state iterate at noise level σ_t, and $\hat{\mathbf{x}}_{0|t}$ is the associated clean prediction used for measurement evaluation in conditional sampling.

2.2. Gauss–Newton quadratic surrogate and damping

Consider the Gaussian negative log-likelihood

$$\Phi(\mathbf{x}) = \frac{1}{2\sigma_n^2}\|\mathbf{r}(\mathbf{x})\|_2^2, \ \mathbf{r}(\mathbf{x}) := A(\mathbf{x}) - \mathbf{y}, \ \mathbf{J} := \nabla_{\mathbf{x}}\mathbf{r}(\mathbf{x}). \tag{3}$$

For an increment Δ, linearize the residual $\mathbf{r}(\mathbf{x}+\Delta) \approx \mathbf{r}(\mathbf{x}) + \mathbf{J}\Delta$ and substitute into Φ:

$$\begin{aligned}\Phi(\mathbf{x}+\Delta) &\approx \frac{1}{2\sigma_n^2}\|\mathbf{r} + \mathbf{J}\Delta\|_2^2 \\ &= \Phi(\mathbf{x}) + \frac{1}{\sigma_n^2}\langle \mathbf{J}^\top\mathbf{r}, \Delta\rangle + \frac{1}{2\sigma_n^2}\Delta^\top(\mathbf{J}^\top\mathbf{J})\Delta,\end{aligned} \tag{4}$$

where $\top$ denotes the Euclidean adjoint. Minimizing the quadratic model (4) yields the GN normal equations (Nocedal & Wright, 2006)

$$(\mathbf{J}^\top\mathbf{J})\,\Delta = -\mathbf{J}^\top\mathbf{r}. \tag{5}$$

Damping (trust-region / Levenberg–Marquardt) stabilizes GN updates when the local model is imperfect or ill-conditioned by adding a positive semidefinite metric (Nocedal & Wright, 2006; Conn et al., 2000; Hanke, 1997):

$$(\mathbf{J}^\top\mathbf{J} + \lambda\mathbf{H})\,\Delta = -\mathbf{J}^\top\mathbf{r}, \quad \mathbf{H} \succeq 0. \tag{6}$$

The curvature $\mathbf{J}^\top\mathbf{J}$ acts as a matrix-valued step size, suppressing updates along sensitive directions; the metric $\mathbf{H}$ provides (possibly anisotropic) step control.

2.3. Matrix-free Jacobian actions and Krylov solvers

Large-scale inverse problems rarely permit explicit formation of $\mathbf{J}$ or $\mathbf{J}^\top\mathbf{J}$. Instead, second-order updates are computed by solving linear systems $\mathbf{M}\Delta = \mathbf{b}$ using only matrix–vector products $\mathbf{M}(\mathbf{v})$. These products are implemented via Jacobian actions: a Jacobian–vector product (JVP) computes $\mathbf{J}\mathbf{v}$ and a vector–Jacobian product (VJP) computes $\mathbf{J}^\top\mathbf{u}$ (Pearlmutter, 1994; Baydin et al., 2018). For linear A, $\mathbf{J}\mathbf{v} = A\mathbf{v}$ and $\mathbf{J}^\top\mathbf{u} = A^\top\mathbf{u}$ exactly; for differentiable nonlinear A, JVP/VJP can be obtained via autodiff.

When $\mathbf{M}$ is symmetric positive-(semi)definite, conjugate gradients (and truncated trust-region variants) are standard (Steihaug, 1983; Conn et al., 2000). When $\mathbf{M}$ is not symmetric, one instead uses a nonsymmetric Krylov solver such as GMRES (Saad & Schultz, 1986; Dembo et al., 1982; Knoll & Keyes, 2004), which remains fully matrix-free and only requires repeated evaluations of $\mathbf{M}(\mathbf{v})$.

3. Method

3.1. Setup, notation, and per-step structure

We consider inverse problems with additive Gaussian measurement noise as in (1) and target posterior sampling

$$p(\mathbf{x}_0 \mid \mathbf{y}) \ \propto \ \exp\!\left(-\frac{1}{2\sigma_n^2}\|A(\mathbf{x}_0) - \mathbf{y}\|_2^2\right) p(\mathbf{x}_0). \tag{7}$$

We access the prior through a pretrained diffusion model with a decreasing noise schedule $\{\sigma_t\}_{t=T}^{0}$, $\sigma_{t-1} < \sigma_t$, and an EDM/Tweedie denoiser interface

$$\hat{\mathbf{x}}_{0|t} = F_\theta(\mathbf{x}_t; \sigma_t), \ \epsilon_\theta(\mathbf{x}_t, \sigma_t) := \frac{\mathbf{x}_t - \hat{\mathbf{x}}_{0|t}}{\sigma_t}. \tag{8}$$

Here $\mathbf{x}_t \in \mathbb{R}^d$ denotes the diffusion-state iterate at noise level σ_t.

Per-step structure. At each noise level σ_t, CLAMP performs: (i) a likelihood correction Δ_t computed in diffusion-state coordinates by solving a damped linear system, and (ii) a prior propagation step that advances $\mathbf{x}_t \mapsto \mathbf{x}_{t-1}$ while preserving the schedule variance. Algorithm 1 in Appendix A provides the full procedure; below we derive each component.

3.2. Why scalar likelihood guidance is brittle

Many diffusion posterior samplers scale a likelihood gradient in $\mathbf{x}_t$-space by a scalar guidance weight. When the local geometry is stiff (ill-conditioned), stable scalar steps are constrained by the largest-curvature direction, which can leave insufficient progress along lower-curvature ("flatter") directions.

Proposition 3.1 (Scalar-step barrier under stiff local geometry)**.** *Fix t and suppose a local objective in the increment*

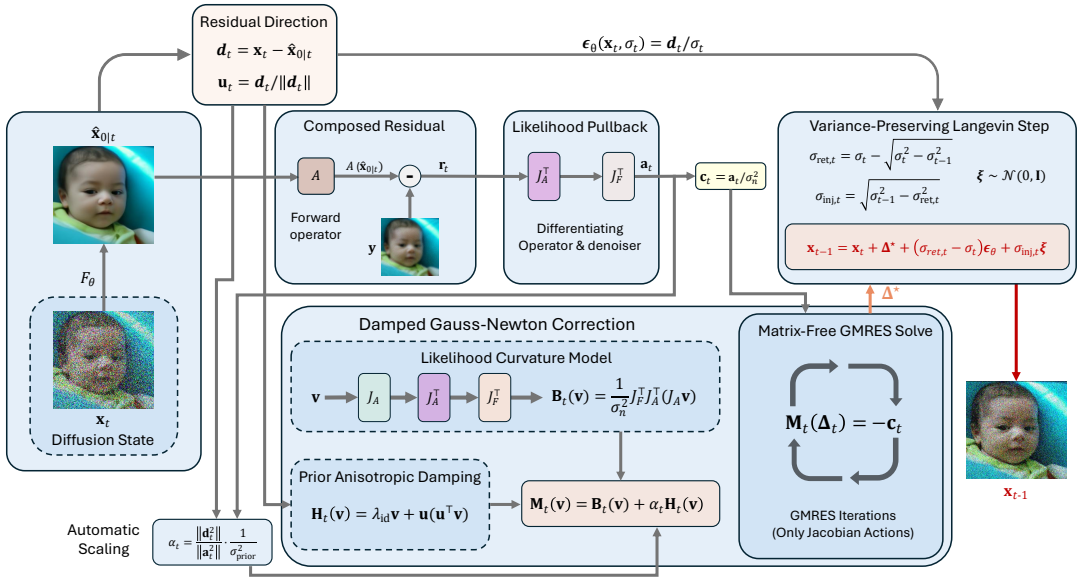

Figure 2. **CLAMP update at one noise level.** (1) *Prior query & residual:* from $\mathbf{x}_t$, compute $\hat{\mathbf{x}}_{0|t}$ and the composed residual by comparing $A(\hat{\mathbf{x}}_{0|t})$ to $\mathbf{y}$. (2) *Curvature-aware correction:* pull back measurement sensitivity through the denoiser to form a one-sided Gauss–Newton model with prior-aligned, automatically scaled damping, then solve the linear system with matrix-free GMRES. (3) *Schedule transition:* apply the correction in a variance-preserving Langevin step to obtain $\mathbf{x}_{t-1}$.

Δ admits a quadratic model $\mathcal{J}(\Delta) \approx \mathcal{J}(\mathbf{0}) + \langle \mathbf{g}_t, \Delta \rangle + \frac{1}{2}\Delta^\top \mathbf{Q}_t \Delta$ with $\mathbf{Q}_t \succ 0$. A sufficient condition for global stability of the scalar-guided update $\Delta_{\text{sc}} = -\eta \mathbf{g}_t$ on this quadratic model is

$$\eta \leq \frac{2}{\lambda_{\max}(\mathbf{Q}_t)}. \tag{9}$$

Let $\Delta^\star = -\mathbf{Q}_t^{-1}\mathbf{g}_t$ denote the quadratic minimizer. In the eigenbasis $\mathbf{Q}_t \mathbf{v}_i = \lambda_i \mathbf{v}_i$, for any i with $\langle \mathbf{v}_i, \mathbf{g}_t \rangle \neq 0$,

$$\frac{|\langle \mathbf{v}_i, \Delta_{\text{sc}} \rangle|}{|\langle \mathbf{v}_i, \Delta^\star \rangle|} = \eta\lambda_i \leq \frac{2\lambda_i}{\lambda_{\max}(\mathbf{Q}_t)}. \tag{10}$$

In particular, along the flattest direction $\mathbf{v}_{\min}$ (associated with $\lambda_{\min}(\mathbf{Q}_t)$), the stability constraint implies $\eta\lambda_{\min}(\mathbf{Q}_t) \leq 2/\kappa(\mathbf{Q}_t)$. Thus when $\kappa(\mathbf{Q}_t) \gg 1$, any globally stable scalar step yields only $O(\lambda_i/\lambda_{\max})$ relative progress in low-curvature directions, and at most $O(1/\kappa)$ relative progress along the flattest direction.

Proposition 3.1 motivates using a matrix-scaled correction that allocates step sizes direction-wise via curvature.

3.3. Stepwise surrogate objective in diffusion-state coordinates

At each noise level σ_t, CLAMP computes Δ_t by minimizing a local quadratic surrogate of the stepwise energy

$$\mathcal{E}_t(\mathbf{x}) = \Phi_t(\mathbf{x}) + \frac{\alpha_t}{2}\|\mathbf{x} - \mathbf{x}_t\|_{\mathbf{H}_t}^2, \quad \|\mathbf{v}\|_{\mathbf{H}_t}^2 := \mathbf{v}^\top \mathbf{H}_t \mathbf{v}, \tag{11}$$

where Φ_t is the stepwise likelihood defined in Section 3.4, and $\mathbf{H}_t \succeq 0$ with $\alpha_t > 0$ define a (possibly anisotropic) damping penalty (specified in Section 3.6). This quadratic penalty plays the role of a trust-region/Levenberg–Marquardt stabilizer (Nocedal & Wright, 2006; Conn et al., 2000), used for step control rather than as an exact diffusion transition density.

3.4. Denoiser-pulled-back likelihood geometry

The likelihood in (7) is defined on a clean variable (the input of A), whereas the algorithm updates the noisy diffusion state $\mathbf{x}_t$. We therefore evaluate measurement residuals on the denoised prediction and define the composed residual and stepwise likelihood as functions of the diffusion-state argument $\mathbf{x}$:

$$\mathbf{r}_t(\mathbf{x}) := A\big(F_\theta(\mathbf{x}; \sigma_t)\big) - \mathbf{y}, \ \Phi_t(\mathbf{x}) := \frac{1}{2\sigma_n^2}\|\mathbf{r}_t(\mathbf{x})\|_2^2. \tag{12}$$

Let $\mathbf{z}_t := F_\theta(\mathbf{x}_t; \sigma_t)$ and $\mathbf{r}_t := \mathbf{r}_t(\mathbf{x}_t) = A(\mathbf{z}_t) - \mathbf{y}$. Define Jacobians at the current evaluation point:

$$J_A := \nabla_{\mathbf{z}} A(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_t}, \quad J_F := \nabla_{\mathbf{x}} F_\theta(\mathbf{x}; \sigma_t)\big|_{\mathbf{x}=\mathbf{x}_t} \tag{13}$$

$$J_t := \nabla_{\mathbf{x}} \mathbf{r}_t(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_t} = J_A J_F. \tag{14}$$

Proposition 3.2 (Coordinate-correct guidance requires denoiser pullback)**.** *Let $\tilde{\Phi}(\mathbf{z}) := \frac{1}{2\sigma_n^2}\|A(\mathbf{z}) - \mathbf{y}\|_2^2$ and $\Phi_t(\mathbf{x}) = \tilde{\Phi}(F_\theta(\mathbf{x}; \sigma_t))$. Then the first-order expansion at*

$\mathbf{x}_t$ *is*

$$\Phi_t(\mathbf{x}_t+\Delta) = \Phi_t(\mathbf{x}_t)+\left\langle J_F^\top \nabla_{\mathbf{z}} \tilde{\Phi}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_t}, \Delta \right\rangle + o(\|\Delta\|_2), \tag{15}$$

so coordinate-correct steepest descent in $\mathbf{x}_t$-space is proportional to $-J_F^\top \nabla_{\mathbf{z}} \tilde{\Phi}(\mathbf{z}_t)$. Moreover, under a Gauss–Newton/Fisher approximation (dropping second-order residual terms), the diffusion-state curvature is $\frac{1}{\sigma_n^2} J_F^\top J_A^\top J_A J_F$.

Proposition 3.2 motivates differentiating through the denoiser (via $J_F^\top$): likelihood derivatives in clean coordinates do not, in general, provide correctly scaled directions for updating $\mathbf{x}_t$ when J_F is anisotropic.

3.5. Penalized Gauss–Newton correction and explicit approximations

Building on Sec. 3.4, we compute the likelihood correction Δ_t by minimizing a quadratic surrogate of $\mathcal{E}_t(\mathbf{x}_t + \Delta)$ in (11), using the composed residual $\mathbf{r}_t(\mathbf{x})$ and Jacobian $J_t = J_A J_F$ from (14).

(A1) Local linearization and penalized Gauss–Newton (standard). Using the first-order residual model

$$\mathbf{r}_t(\mathbf{x}_t + \Delta) \approx \mathbf{r}_t + J_t \Delta, \tag{16}$$

substituting into (11) yields the quadratic approximation

$$\mathcal{E}_t(\mathbf{x}_t + \Delta) \;\approx\; \frac{1}{2\sigma_n^2}\|\mathbf{r}_t + J_t\Delta\|_2^2 + \frac{\alpha_t}{2}\Delta^\top \mathbf{H}_t \Delta. \tag{17}$$

Its stationary point solves the ideal penalized Gauss–Newton system

$$\left(\frac{1}{\sigma_n^2} J_t^\top J_t + \alpha_t \mathbf{H}_t\right)\Delta = -\frac{1}{\sigma_n^2} J_t^\top \mathbf{r}_t. \tag{18}$$

With $J_t = J_A J_F$, the curvature term is $J_t^\top J_t = J_F^\top J_A^\top J_A J_F$, which requires forward denoiser-Jacobian products of the form $J_F\mathbf{v}$ inside curvature matvecs.

(A2) One-sided curvature approximation (CLAMP design choice). To avoid forward denoiser-Jacobian products inside the curvature, CLAMP adopts a *one-sided* Gauss–Newton/Fisher model that approximates the forward denoiser perturbation as $J_F\mathbf{v} \approx \mathbf{v}$ *only when forming the curvature operator*, while retaining the pullback $J_F^\top$ in the adjoint pass so that update directions remain coordinate-correct under anisotropic J_F (Prop. 3.2). This approximation is motivated by the local translation heuristic $F_\theta(\mathbf{x}_t + \Delta; \sigma_t) \approx F_\theta(\mathbf{x}_t; \sigma_t) + \Delta$ for small Δ.

A useful sanity check is that the diffusion-state gradient of the stepwise likelihood is exact under this construction:

$$\nabla_{\mathbf{x}} \Phi_t(\mathbf{x}_t) = \frac{1}{\sigma_n^2} J_F^\top J_A^\top \mathbf{r}_t,$$

whereas only the curvature is approximated. In particular, if we define the *exact* GN/Fisher curvature operator $\mathbf{B}_t^{\text{exact}}(\mathbf{v}) := \sigma_n^{-2} J_F^\top J_A^\top J_A J_F \mathbf{v}$, then the one-sided operator $\mathbf{B}_t$ defined below satisfies

$$\big(\mathbf{B}_t^{\text{exact}} - \mathbf{B}_t\big)(\mathbf{v}) = \frac{1}{\sigma_n^2} J_F^\top J_A^\top J_A (J_F - I)\mathbf{v},$$

so the curvature discrepancy is controlled by the deviation of the forward action from identity (i.e., $\|J_F - I\|$), while eliminating $J_F\mathbf{v}$ products in curvature matvecs.

Asymmetric Jacobian actions. Concretely, we use the asymmetric Jacobian actions

$$\tilde{J}_t(\mathbf{v}) := J_A \mathbf{v}, \qquad \tilde{J}_t^\top(\mathbf{u}) := J_F^\top J_A^\top \mathbf{u}, \tag{19}$$

which induce the matrix-free operators

$$\mathbf{B}_t(\mathbf{v}) := \frac{1}{\sigma_n^2}\tilde{J}_t^\top\big(\tilde{J}_t(\mathbf{v})\big) = \frac{1}{\sigma_n^2} J_F^\top J_A^\top\big(J_A\mathbf{v}\big), \tag{20}$$

$$\mathbf{c}_t := \frac{1}{\sigma_n^2}\tilde{J}_t^\top \mathbf{r}_t = \frac{1}{\sigma_n^2} J_F^\top J_A^\top \mathbf{r}_t. \tag{21}$$

Compared to the exact GN/Fisher curvature $\sigma_n^{-2} J_F^\top J_A^\top J_A J_F$, $\mathbf{B}_t$ drops the rightmost J_F factor. Since $\tilde{J}_t(\cdot)$ and $\tilde{J}_t^\top(\cdot)$ are not adjoints of a single linear map, $\mathbf{B}_t$ (and thus $\mathbf{M}_t$ below) need not be self-adjoint, motivating a nonsymmetric Krylov solver (GMRES).

Linear system to solve. The correction Δ_t is defined as the solution of the damped system

$$\mathbf{M}_t(\Delta_t) = -\mathbf{c}_t, \qquad \mathbf{M}_t(\mathbf{v}) := \mathbf{B}_t(\mathbf{v}) + \alpha_t \mathbf{H}_t(\mathbf{v}), \tag{22}$$

which we solve with GMRES (Sec. 3.7) using only matrix–vector products.

3.6. Prior-aligned anisotropic damping and diffusion-calibrated regularization

Rank-one anisotropic metric. Define the denoiser residual and its unit direction

$$\mathbf{d}_t := \mathbf{x}_t - \hat{\mathbf{x}}_{0|t}, \; \mathbf{u}_t := \frac{\mathbf{d}_t}{\|\mathbf{d}_t\|_2}. \tag{23}$$

We set

$$\mathbf{H}_t := \lambda_{\text{id}}\mathbf{I} + \mathbf{u}_t\mathbf{u}_t^\top, \; \mathbf{H}_t\mathbf{v} = \lambda_{\text{id}}\mathbf{v} + \mathbf{u}_t(\mathbf{u}_t^\top\mathbf{v}), \tag{24}$$

where $\lambda_{\text{id}} \geq 0$ provides baseline isotropic damping and prevents degeneracy. This metric penalizes motion along the denoiser residual direction $\mathbf{u}_t$ more strongly than orthogonal directions, yielding explicit step control aligned with a diffusion-prior-indicated direction.

Diffusion-calibrated regularization strength. Define the (unscaled) backprojected residual in diffusion coordinates

$$\mathbf{a}_t := \sigma_n^2\, \mathbf{c}_t = J_F^\top J_A^\top \mathbf{r}_t. \tag{25}$$

We set

$$\alpha_t := \frac{\|\mathbf{d}_t\|_2}{\|\mathbf{a}_t\|_2} \cdot \frac{1}{\sigma_{\text{prior}}^2}. \tag{26}$$

where $\sigma_{\text{prior}} = \max(\sigma_{t-1}, \sigma_{\min}^+)$. The ratio $\|\mathbf{d}_t\|_2/\|\mathbf{a}_t\|_2$ provides an automatic scale match between the prior-indicated magnitude (the denoiser residual $\mathbf{d}_t = \mathbf{x}_t - \hat{\mathbf{x}}_{0|t}$) and the likelihood signal seen in diffusion coordinates (the backprojection $\mathbf{a}_t$): when the likelihood pullback is strong, α_t decreases to avoid over-damping; when it is weak, α_t increases to keep corrections conservative. The factor $\sigma_{\text{prior}}^{-2}$ ties the damping to the remaining diffusion noise level, yielding comparable step control across t. Finally, defining $\mathbf{a}_t = \sigma_n^2 \mathbf{c}_t$ removes explicit σ_n^{-2} dependence from α_t, so higher measurement noise naturally weakens likelihood corrections through $\mathbf{B}_t$ and $\mathbf{c}_t$ without retuning.

3.7. Matrix-free GMRES and Jacobian-action interface

We never form J_A, J_F, $\mathbf{B}_t$, or $\mathbf{H}_t$ explicitly. Instead, we assume access to matrix-free Jacobian actions: a JVP through A to compute $J_A\mathbf{v}$ (for linear A, this equals $A\mathbf{v}$ exactly), a VJP through A (or an explicit adjoint when available) to compute $J_A^\top \mathbf{u}$, and a VJP through the denoiser to compute $J_F^\top \mathbf{u}$. With these primitives, we evaluate

$$\mathbf{B}_t(\mathbf{v}) = \sigma_n^{-2}\, J_F^\top\Big(J_A^\top\big(J_A\mathbf{v}\big)\Big), \tag{27}$$

$$\mathbf{c}_t = \sigma_n^{-2}\, J_F^\top\big(J_A^\top \mathbf{r}_t\big),\ \mathbf{H}_t(\mathbf{v}) = \lambda_{\text{id}}\mathbf{v} + \mathbf{u}_t(\mathbf{u}_t^\top \mathbf{v}). \tag{28}$$

For differentiable nonlinear A, these Jacobian actions can be obtained via automatic differentiation (Baydin et al., 2018); if automatic differentiation is unavailable, one may approximate $J_A\mathbf{v}$ using standard numerical directional-derivative schemes (e.g., finite differences) (Knoll & Keyes, 2004).

We solve (22) using GMRES (Saad & Schultz, 1986), which applies to nonsymmetric operators and is fully matrix-free (see Algorithm 2 in Appendix A for details). We run GMRES for a fixed iteration budget K (Dembo et al., 1982; Knoll & Keyes, 2004). A fixed Krylov budget is appropriate because our operator is derived from a local linearization and a one-sided curvature approximation: over-solving can amplify model mismatch, whereas moderate K yields stable updates.

Per-step computational cost. Each GMRES iteration requires one application of $\mathbf{M}_t(\cdot)$, i.e., one JVP through A, one VJP through A, one VJP through F_θ, and one rank-one metric application.

3.8. Variance-preserving Langevin propagation

After computing Δ_t, we propagate from σ_t to σ_{t-1}. A deterministic EDM-style probability-flow update would be

$$\mathbf{x}_{t-1}^{\text{ODE}} = \mathbf{x}_t + \Delta_t + (\sigma_{t-1} - \sigma_t)\,\epsilon_\theta(\mathbf{x}_t, \sigma_t). \tag{29}$$

In posterior sampling, the additional drift Δ_t can make purely deterministic propagation sensitive to discretization and model mismatch. We instead use a stochastic transition that preserves the schedule variance while maintaining stochastic exploration:

$$\mathbf{x}_{t-1} = \mathbf{x}_t + \Delta_t + (\sigma_{\text{ret},t} - \sigma_t)\,\epsilon_\theta(\mathbf{x}_t, \sigma_t) + \sigma_{\text{inj},t}\,\boldsymbol{\xi}, \tag{30}$$

where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We enforce variance matching with the schedule:

$$\sigma_{t-1}^2 = \sigma_{\text{ret},t}^2 + \sigma_{\text{inj},t}^2. \tag{31}$$

Using $\epsilon_\theta = (\mathbf{x}_t - \hat{\mathbf{x}}_{0|t})/\sigma_t$, the drift can be rewritten as

$$(\sigma_{\text{ret},t} - \sigma_t)\,\epsilon_\theta = \lambda_t\big(F_\theta(\mathbf{x}_t; \sigma_t) - \mathbf{x}_t\big), \tag{32}$$

$$\lambda_t := 1 - \frac{\sigma_{\text{ret},t}}{\sigma_t}, \tag{33}$$

so (30) is a single annealed-Langevin (unadjusted Langevin algorithm; ULA) step (Durmus & Moulines, 2017): move a fraction λ_t toward the denoiser output and inject Gaussian noise.

Many stochastic diffusion samplers introduce an explicit stochasticity parameter to set the injected-noise term (with the complementary coefficient fixed by the schedule constraint), e.g., η in DDIM (Song et al., 2021a) and (η, η_b) in DDRM (Kawar et al., 2022). CLAMP instead determines $(\sigma_{\text{ret},t}, \sigma_{\text{inj},t})$ from the schedule by imposing a fluctuation–dissipation coupling at the geometric reference $\sigma_{\text{ref}}^2 := \sigma_t\sigma_{\text{ret},t}$, yielding a closed form.

Proposition 3.3 (Drift/noise split under variance preservation)**.** *Assume (i) variance matching* (31) *and (ii) a Langevin fluctuation–dissipation coupling at* $\sigma_{\text{ref}}^2 := \sigma_t\sigma_{\text{ret},t}$*, i.e.* $\sigma_{\text{inj},t}^2 = 2\lambda_t\sigma_{\text{ref}}^2$ *with* λ_t *from* (32)*. Then* $(\sigma_{\text{ret},t}, \sigma_{\text{inj},t})$ *is uniquely determined:*

$$\sigma_{\text{ret},t} = \sigma_t - \sqrt{\sigma_t^2 - \sigma_{t-1}^2},\ \sigma_{\text{inj},t} = \sqrt{\sigma_{t-1}^2 - \sigma_{\text{ret},t}^2}. \tag{34}$$

3.9. Latent CLAMP

To use a latent diffusion prior, we run CLAMP on latent states $\mathbf{z}_t$ with a differentiable decoder $D : \mathbb{R}^{d_z} \to \mathbb{R}^d$ that maps latents to pixels. Since measurements are defined in pixel space, it is natural to write the forward model through the composition

$$\tilde{A} := A \circ D, \qquad \mathbf{y} = \tilde{A}(\mathbf{z}_0) + \mathbf{n},\ \ \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I}). \tag{35}$$

Table 1. Quantitative comparison on FFHQ and ImageNet across inverse-problem tasks in pixel setting. We report average PSNR/SSIM (higher is better), LPIPS (lower is better), and run-time (lower is better) over 100 validation images. Best and second-best scores are highlighted in **bold** and underlined, respectively.

Task	Method	FFHQ				ImageNet			
		PSNR ↑	SSIM ↑	LPIPS ↓	Run-time (s)	PSNR ↑	SSIM ↑	LPIPS ↓	Run-time (s)
Super resolution 4×	Ours	**29.515**	**0.841**	**0.219**	6.743	**26.981**	**0.742**	**0.290**	23.587
	DAPS	28.619	0.764	0.262	28.257	25.512	0.636	0.374	65.580
	SITCOM	29.153	0.826	0.231	15.755	26.519	0.716	0.309	60.778
	DMPlug	28.637	0.797	0.253	118.255	25.091	0.660	0.347	238.433
	DCDP	27.611	0.785	0.225	**4.719**	24.358	0.640	0.369	**9.190**
Inpaint (box)	Ours	**25.495**	**0.882**	**0.125**	**6.175**	**21.547**	**0.825**	**0.188**	23.427
	DAPS	24.285	0.746	0.222	22.352	21.234	0.720	0.270	57.108
	SITCOM	25.275	0.831	0.179	17.697	20.698	0.726	0.241	66.709
	DMPlug	23.526	0.790	0.281	103.76	19.455	0.667	0.384	253.946
	DCDP	21.721	0.766	0.219	8.147	17.741	0.662	0.301	**17.214**
Inpaint (random)	Ours	**32.955**	**0.918**	**0.140**	**6.171**	**30.215**	**0.866**	**0.159**	23.419
	DAPS	30.078	0.791	0.214	21.576	27.350	0.721	0.251	56.932
	SITCOM	31.431	0.871	0.173	25.382	29.008	0.821	0.183	95.962
	DMPlug	30.825	0.865	0.231	122.143	27.023	0.739	0.317	248.146
	DCDP	27.240	0.788	0.229	8.130	23.284	0.606	0.355	**17.518**
Gaussian deblurring	Ours	29.010	**0.827**	**0.234**	6.343	26.388	0.710	0.333	23.924
	DAPS	28.762	0.767	0.254	37.730	25.932	0.655	0.355	74.984
	SITCOM	**29.359**	0.826	0.236	25.491	**26.741**	**0.719**	**0.315**	95.161
	DMPlug	27.95	0.770	0.278	181.695	24.082	0.613	0.389	313.680
	DCDP	27.553	0.761	0.245	**4.934**	23.499	0.542	0.449	**9.573**
Motion deblurring	Ours	**31.641**	**0.880**	0.183	6.322	29.141	**0.812**	0.243	23.805
	DAPS	30.954	0.823	0.202	38.606	28.623	0.764	0.246	78.000
	SITCOM	31.501	0.866	**0.167**	26.184	**29.321**	0.811	**0.213**	95.529
	DMPlug	29.272	0.818	0.260	152.699	25.211	0.660	0.365	284.413
	DCDP	25.468	0.566	0.365	**5.007**	17.771	0.248	0.574	**9.488**
Phase retrieval	Ours	**30.233**	**0.854**	**0.192**	28.660	19.680	0.459	0.478	104.628
	DAPS	30.103	0.796	0.208	98.054	**22.354**	**0.519**	**0.402**	241.113
	SITCOM	28.782	0.791	0.239	**24.584**	18.481	0.383	0.524	**94.187**
	DCDP	24.036	0.677	0.310	102.386	15.953	0.283	0.595	195.193
Nonlinear deblur	Ours	**29.961**	**0.856**	**0.166**	33.946	28.036	**0.788**	**0.212**	**82.427**
	DAPS	28.868	0.780	0.223	755.725	27.537	0.734	0.266	884.119
	SITCOM	29.519	0.812	0.207	**27.167**	**28.097**	0.774	0.221	97.029
	DMPlug	28.298	0.811	0.249	291.221	25.086	0.679	0.317	668.641
	DCDP	27.879	0.795	0.204	269.786	25.726	0.664	0.280	367.695
High dynamic range	Ours	**29.488**	**0.891**	**0.133**	**26.610**	**28.804**	**0.886**	**0.136**	**104.38**
	DAPS	27.149	0.834	0.196	76.724	26.568	0.819	0.198	216.701
	SITCOM	27.205	0.777	0.226	33.887	26.449	0.774	0.222	129.681
	DMPlug	25.507	0.777	0.264	221.418	23.244	0.715	0.308	564.501

At noise level σ_t, the latent denoiser produces $\hat{\mathbf{z}}_{0|t} = F_\theta(\mathbf{z}_t; \sigma_t)$ and the residual is evaluated as $\mathbf{r}_t = \tilde{A}(\hat{\mathbf{z}}_{0|t}) - \mathbf{y}$. The latent-space algorithm is otherwise identical to the pixel-space derivation: replace A by $\tilde{A}$ and $\mathbf{x}_t$ by $\mathbf{z}_t$ everywhere, retaining the denoiser pullback, anisotropic damping, matrix-free GMRES likelihood solve, and variance-preserving Langevin propagation. Operationally, this substitution only changes the required Jacobian actions: each $\mathrm{JVP}_A/\mathrm{VJP}_A$ call becomes a $\mathrm{JVP}_{\tilde{A}}/\mathrm{VJP}_{\tilde{A}}$ call (and thus traverses D), while the denoiser pullback still uses VJP_F. Algorithm 3 in Appendix A summarizes the resulting procedure, with final output $\mathbf{x}_0 = D(\mathbf{z}_0)$.

4. Experiments

4.1. Experimental setup

We largely follow the experimental setup of DAPS (Zhang et al., 2025) and evaluate on eight task settings, comprising five linear and three nonlinear inverse problems. Implementation details and hyperparameter configurations for all baselines and our method, as well as other experimental details are provided in Appendix C.

4.2. Main Results

Tables 1 and 3 compare our method to baselines on FFHQ and ImageNet across eight inverse-problem tasks in both pixel and latent settings. Overall, we match or improve quantitative metrics, and Figure 1 shows reconstructions with fewer artifacts across representative tasks.

Nonlinear tasks are challenging because a nonlinear forward model $A(\cdot)$ makes the data term nonconvex and can induce rapidly varying local curvature along the sampling trajectory, so likelihood-based corrections become highly sensitive to step size and direction. This is especially evident for nonlinear deblurring: small mis-calibration of the correction can lead to instability or pronounced artifacts. In contrast, our method remains stable in both pixel and latent settings. We attribute this behavior to combining (i) curvature-aware scaling of the likelihood correction with (ii) prior-aligned regularization that constrains updates in sensitive directions, which together stabilizes the conditioning step.

Beyond reconstruction quality, Table 1, 3 and Figure 3 also highlight the computational efficiency of our approach: we match or improve metrics while significantly reducing runtime relative to competing methods. For instance, SITCOM is a strong and fast baseline, yet on pixel-space FFHQ motion deblurring we achieve up to a $4.14\times$ speedup while attaining higher PSNR. In latent space, we are approximately $2.4\times$ faster than Latent DAPS and about $9\times$ faster than ReSample, while also delivering better metrics overall.

We provide additional experiments in Appendix D, including higher-resolution evaluations, latent-space experiments, sample results across a wider range of datasets and tasks, hyperparameter analyses, sample diversity under severe degradations, experiments with various measurement-noise settings, and normalized residual checks.

4.3. Accelerated MRI reconstruction results

To assess applicability beyond natural images, we evaluate on multi-coil accelerated MRI reconstruction using SKM-TEA dataset (Desai et al., 2021). Table 2 reports PSNR/SSIM for $\times 4$ and $\times 8$ acceleration. Our method achieves the best PSNR/SSIM among the compared diffusion-based baselines. Appendix E provides experimental details, including the baselines and qualitative results.

Table 2. Quantitative comparison for MRI reconstruction under Poisson-disc undersampling at acceleration factors $\times 4$ and $\times 8$. We report average PSNR/SSIM over 100 test slices. Best scores are in **bold** and second-best are underlined.

Method	$\times 4$		$\times 8$	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DPS	25.46	0.531	25.40	0.527
DAPS	30.20	0.687	28.19	0.582
DDS	32.07	0.763	30.03	0.669
Score-Med	<u>32.21</u>	<u>0.776</u>	<u>30.29</u>	<u>0.694</u>
Ours	**34.05**	**0.834**	**32.27**	**0.766**

4.4. Ablation studies

We ablate four components under matched compute (same diffusion steps and linear-solver budgets): noise-level transition, prior-aligned damping, curvature operator, and Krylov solver. Table 7 shows that (i) replacing the variance-preserving transition with an ODE-style update and (ii) removing damping both degrade quality, especially for nonlinear operators. In the curvature model, dropping the pullback $J_F^\top$ speeds up inference but reduces accuracy, whereas using the full curvature (including forward J_F) substantially increases runtime with little or no gain over the one-sided approximation, supporting our choice as the best accuracy–efficiency trade-off. Finally, since the resulting system is generally nonsymmetric, GMRES is more stable than CG in practice; additional results are in Appendix D.

5. Conclusion

CLAMP provides a training-free posterior sampler for inverse problems that works with pixel and latent diffusion priors and supports both linear and nonlinear forward models. Its core update replaces heuristic scalar guidance with a denoiser-pullback Gauss–Newton correction and a prior-aligned anisotropic damping metric, producing noise-level–dependent matrix-scaled steps. Although each correction uses additional Jacobian actions (JVP/VJP) beyond a single backprojected gradient, it is computed with a fixed-budget matrix-free solve and yields favorable end-to-end runtime in practice. Apart from compute-budget choices (diffusion steps and Krylov iterations), CLAMP uses a single damping hyperparameter (λ_{id}) that is kept nearly unchanged across tasks. Empirically, it matches or improves reconstruction quality over the compared baselines while reducing inference time on FFHQ/ImageNet inverse problems and MRI reconstruction. Limitations and future directions are discussed in Appendix F.

Impact Statement

CLAMP makes diffusion posterior sampling for inverse problems more reliable and faster by replacing hand-tuned scalar guidance with a noise-level–specific, geometry-aware correction. It computes coordinate-correct likelihood updates by pulling measurement sensitivity back through the denoiser, then stabilizes them with manifold-aligned anisotropic damping—yielding direction-wise step control without per-task schedules. A matrix-free GMRES implementation and a variance-preserving Langevin transition deliver a strong quality–runtime trade-off across diverse linear/nonlinear operators, including medical imaging (accelerated MRI).

References

Alkhouri, I., Liang, S., Huang, C.-H., Dai, J., Qu, Q., Ravishankar, S., and Wang, R. SITCOM: Step-wise triple-consistent diffusion sampling for inverse problems. In *International Conference on Machine Learning*, 2025. 2, 19, 20

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153): 1–43, 2018. 3, 6

Bertero, M. and Boccacci, P. *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, Bristol, UK, 1998. 1

Calvetti, D. and Somersalo, E. *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*, volume 2 of *Surveys and Tutorials in the Applied Mathematical Sciences*. Springer, New York, NY, 2007. 1

Chung, H., Sim, B., Ryu, D., and Ye, J. C. Improving diffusion models for inverse problems using manifold constraints. In *Advances in Neural Information Processing Systems*, 2022. 2

Chung, H., Kim, J., McCann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023a. 1, 19, 23, 26

Chung, H., Kim, J., and Ye, J. C. Direct diffusion bridge using data consistency for inverse problems. In *Advances in Neural Information Processing Systems*, 2023b. 2

Chung, H., Lee, S., and Ye, J.-C. Decomposed diffusion sampler for accelerating large-scale inverse problems. In *International Conference on Learning Representations*, 2024. 26

Conn, A. R., Gould, N. I. M., and Toint, P. L. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000. 2, 3, 4

Dashti, M. and Stuart, A. M. The bayesian approach to inverse problems. In *Handbook of uncertainty quantification*, pp. 1–118. Springer, 2015. 1

Dembo, R. S., Eisenstat, S. C., and Steihaug, T. Inexact newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982. 3, 6

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Li, F.-F. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 19

Desai, A. D., Schmidt, A. M., Rubin, E. B., Sandino, C. M., Black, M. S., Mazzoli, V., Stevens, K. J., Boutin, R., Ré, C., Gold, G. E., Hargreaves, B. A., and Chaudhari, A. S. SKM-TEA: A dataset for accelerated MRI reconstruction with dense image labels for quantitative clinical evaluation. In *NeurIPS Track on Datasets and Benchmarks*, 2021. 8, 26

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021. 19, 22

Durmus, A. and Moulines, É. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, 27(3):1551–1587, 2017. 6

Feng, B. T., Smith, J., Rubinstein, M., Chang, H., Bouman, K. L., and Freeman, W. T. Score-based diffusion models as principled priors for inverse imaging. In *International Conference on Computer Vision (ICCV)*, 2023. 1

Garber, T. and Tirer, T. Image restoration by denoising diffusion models with iteratively preconditioned guidance. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

Graikos, A., Jojic, N., and Samaras, D. Fast constrained sampling in pre-trained diffusion models. In *Advances in Neural Information Processing Systems*, 2025. 2

Hamidi, S. M., Yang, E., and Liang, B. Coupled data and measurement space dynamics for enhanced diffusion posterior sampling. In *Advances in Neural Information Processing Systems*, 2025. 2

Hanke, M. A regularizing Levenberg–Marquardt scheme, with applications to inverse groundwater filtration problems. *Inverse Problems*, 13(1):79–95, 1997. 3

Ho, J., Jain, A. N., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 1, 3

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 19

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022. 1, 3, 19

Kawar, B., Elad, M., Ermon, S., and Song, J. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022. 6

Knoll, D. A. and Keyes, D. E. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004. 3, 6

Li, X., Kwon, S. M., Alkhouri, I. R., Ravishankar, S., and Qu, Q. Decoupled data consistency with diffusion purification for image restoration. *arXiv*, 2024. 19, 20

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020. 26

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, 2 edition, 2006. 2, 3, 4

Pearlmutter, B. A. Fast exact multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994. 3

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 19

Rout, L., Raoof, N., Daras, G., Caramanis, C., Dimakis, A. G., and Shakkottai, S. Solving linear inverse problems provably via posterior sampling with latent diffusion models. In *Advances in Neural Information Processing Systems*, 2023. 19, 20

Saad, Y. and Schultz, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986. 3, 6

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 1

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015. 1

Song, B., Kwon, S. M., Zhang, Z., Hu, X., Qu, Q., and Shen, L. Solving inverse problems with latent diffusion models via hard data consistency. In *International Conference on Learning Representations*, 2024. 19, 20

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. 6

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. 1

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. 1, 3

Song, Y., Shen, L., Xing, L., and Ermon, S. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022. 26

Steihaug, T. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983. 3

Stuart, A. M. Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559, 2010. 1

Wang, H., Li, T., Zhuang, Z., Chen, T., Liang, H., and Sun, J. Early stopping for deep image prior. *Transactions on Machine Learning Research*, 2023. 20

Wang, H., Zhang, X., Li, T., Wan, Y., Chen, T., and Sun, J. DMPlug: A plug-in method for solving inverse problems with diffusion models. In *Advances in Neural Information Processing Systems*, 2024. 19, 20

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 19

Yismaw, N., Kamilov, U. S., and Asif, M. S. Gaussian is all you need: A unified framework for solving inverse problems via diffusion posterior sampling. *IEEE Transactions on Computational Imaging*, 11:1020–1030, 2025. 2

Zhang, B., Chu, W., Berner, J., Meng, C., Anandkumar, A., and Song, Y. Improving diffusion inverse problem solving with decoupled noise annealing. In *Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 8, 19, 20, 26

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 19

A. Algorithms

Algorithm 1 CLAMP

Require: Measurement $\mathbf{y}$; forward model A; noise std σ_n.
Denoiser $F_\theta(\cdot;\sigma)$; schedule $\{\sigma_t\}_{t=T}^{0}$.
$\lambda_{\mathrm{id}} \geq 0$; GMRES iterations K.
▷ All Jacobian actions are matrix-free: use JVP for $J_A\mathbf{v}$ and VJP for $J_A^\top\mathbf{u}$, $J_F^\top\mathbf{u}$.

$\sigma_{\min}^{+} \leftarrow \min\{\sigma_t : \sigma_t > 0\}$.
Draw $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I})$.
for $t = T, \ldots, 1$ **do**
  $\hat{\mathbf{x}}_{0|t} \leftarrow F_\theta(\mathbf{x}_t;\sigma_t)$
  $\mathbf{r}_t \leftarrow A(\hat{\mathbf{x}}_{0|t}) - \mathbf{y}$
  $\mathbf{d}_t \leftarrow \mathbf{x}_t - \hat{\mathbf{x}}_{0|t}$; $\hat{\boldsymbol{\epsilon}}_\theta \leftarrow \mathbf{d}_t/\sigma_t$
  $\sigma_{\mathrm{prior}} \leftarrow \max(\sigma_{t-1}, \sigma_{\min}^{+})$
  $\mathbf{u}_t \leftarrow \mathbf{d}_t/\|\mathbf{d}_t\|_2$
  $\mathbf{a}_t \leftarrow \mathrm{VJP}_F\big(\mathrm{VJP}_A(\mathbf{r}_t)\big)$
  $\mathbf{c}_t \leftarrow \mathbf{a}_t/\sigma_n^2$
  $\alpha_t \leftarrow (\|\mathbf{d}_t\|_2/\|\mathbf{a}_t\|_2)\cdot\sigma_{\mathrm{prior}}^{-2}$
  Define $\mathbf{H}_t(\mathbf{v}) \leftarrow \lambda_{\mathrm{id}}\mathbf{v} + \mathbf{u}_t(\mathbf{u}_t^\top\mathbf{v})$
  Define $\mathbf{B}_t(\mathbf{v}) \leftarrow \sigma_n^{-2}\,\mathrm{VJP}_F\big(\mathrm{VJP}_A(\mathrm{JVP}_A(\mathbf{v}))\big)$
  Define $\mathbf{M}_t(\mathbf{v}) \leftarrow \mathbf{B}_t(\mathbf{v}) + \alpha_t\mathbf{H}_t(\mathbf{v})$
  $\Delta_t \leftarrow \mathrm{GMRES}(\mathbf{M}_t,\ -\mathbf{c}_t;\ K)$ (Alg. 2)
  $\sigma_{\mathrm{ret},t} \leftarrow \sigma_t - \sqrt{\sigma_t^2 - \sigma_{t-1}^2}$
  $\sigma_{\mathrm{inj},t} \leftarrow \sqrt{\sigma_{t-1}^2 - \sigma_{\mathrm{ret},t}^2}$
  $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_{t-1} \leftarrow \mathbf{x}_t + \Delta_t + (\sigma_{\mathrm{ret},t} - \sigma_t)\hat{\boldsymbol{\epsilon}}_\theta + \sigma_{\mathrm{inj},t}\boldsymbol{\xi}$
end for
return $\mathbf{x}_0$

Algorithm 2 GMRES for $(\mathbf{B}_t + \alpha_t \mathbf{H}_t)\Delta = -\mathbf{c}_t$

Require: Matvec $\mathbf{M}(\mathbf{v}) := \mathbf{B}_t(\mathbf{v}) + \alpha_t \mathbf{H}_t(\mathbf{v})$; RHS $\mathbf{b} := -\mathbf{c}_t$; Krylov budget K; breakdown tolerance τ (e.g., 10^{-12}).
Ensure: Approximate solution Δ.

$\Delta_0 \leftarrow \mathbf{0}$
$\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{M}(\Delta_0)$
$\beta \leftarrow \|\mathbf{r}_0\|_2$
if $\beta < \tau$ **then**
    return $\mathbf{0}$
end if
$\mathbf{v}_1 \leftarrow \mathbf{r}_0/\beta$
$m \leftarrow K$
for $j = 1, 2, \ldots, K$ **do**
    $\mathbf{w} \leftarrow \mathbf{M}(\mathbf{v}_j)$
    for $\ell = 1, 2, \ldots, j$ **do**
        $h_{\ell j} \leftarrow \mathbf{v}_\ell^\top \mathbf{w}$
        $\mathbf{w} \leftarrow \mathbf{w} - h_{\ell j}\mathbf{v}_\ell$
    end for
    $h_{j+1,j} \leftarrow \|\mathbf{w}\|_2$
    if $h_{j+1,j} < \tau$ **then**
        $m \leftarrow j$
        break
    end if
    $\mathbf{v}_{j+1} \leftarrow \mathbf{w}/h_{j+1,j}$
end for
Form the (upper) Hessenberg matrix $\bar{\mathcal{H}}_m \in \mathbb{R}^{(m+1)\times m}$ from $\{h_{\ell j}\}_{1\le\ell\le j+1,\ 1\le j\le m}$.
Set $\mathbf{e}_1 \leftarrow [\beta, 0, \ldots, 0]^\top \in \mathbb{R}^{m+1}$
Solve $\mathbf{y} = \arg\min_{\mathbf{y}\in\mathbb{R}^m} \|\bar{\mathcal{H}}_m \mathbf{y} - \mathbf{e}_1\|_2$ (least squares; e.g., QR/SVD)
$\Delta \leftarrow \sum_{j=1}^{m} y_j \mathbf{v}_j$
return Δ

Algorithm 3 Latent-CLAMP

Require: Measurement $\mathbf{y}$; forward model A; noise std σ_n.
Latent denoiser $F_\theta(\cdot;\sigma)$; decoder $D(\cdot)$; schedule $\{\sigma_t\}_{t=T}^{0}$.
$\lambda_{\mathrm{id}} \geq 0$; GMRES iterations K.
▷ All Jacobian actions are matrix-free: use JVP/VJP for $A\circ D$ and VJP for F.

$\sigma^{+}_{\min} \leftarrow \min\{\sigma_t : \sigma_t > 0\}$.
Draw $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \mathbf{I})$.
for $t = T, \ldots, 1$ **do**
  $\hat{\mathbf{z}}_{0|t} \leftarrow F_\theta(\mathbf{z}_t; \sigma_t)$
  $\mathbf{r}_t \leftarrow A\big(D(\hat{\mathbf{z}}_{0|t})\big) - \mathbf{y}$
  $\mathbf{d}_t \leftarrow \mathbf{z}_t - \hat{\mathbf{z}}_{0|t}$; $\hat{\boldsymbol{\epsilon}}_\theta \leftarrow \mathbf{d}_t/\sigma_t$
  $\sigma_{\mathrm{prior}} \leftarrow \max(\sigma_{t-1}, \sigma^{+}_{\min})$
  $\mathbf{u}_t \leftarrow \mathbf{d}_t / \|\mathbf{d}_t\|_2$
  $\mathbf{a}_t \leftarrow \mathrm{VJP}_F\Big(\mathrm{VJP}_{A\circ D}(\mathbf{r}_t)\Big)$
  $\mathbf{c}_t \leftarrow \mathbf{a}_t/\sigma_n^2$
  $\alpha_t \leftarrow (\|\mathbf{d}_t\|_2/\|\mathbf{a}_t\|_2)\cdot \sigma_{\mathrm{prior}}^{-2}$
  Define $\mathbf{H}_t(\mathbf{v}) \leftarrow \lambda_{\mathrm{id}}\mathbf{v} + \mathbf{u}_t(\mathbf{u}_t^\top \mathbf{v})$
  Define $\mathbf{B}_t(\mathbf{v}) \leftarrow \sigma_n^{-2}\,\mathrm{VJP}_F\Big(\mathrm{VJP}_{A\circ D}(\mathrm{JVP}_{A\circ D}(\mathbf{v}))\Big)$
  Define $\mathbf{M}_t(\mathbf{v}) \leftarrow \mathbf{B}_t(\mathbf{v}) + \alpha_t \mathbf{H}_t(\mathbf{v})$
  $\Delta_t \leftarrow \mathrm{GMRES}(\mathbf{M}_t, -\mathbf{c}_t; K)$ (Alg. 2)
  $\sigma_{\mathrm{ret},t} \leftarrow \sigma_t - \sqrt{\sigma_t^2 - \sigma_{t-1}^2}$
  $\sigma_{\mathrm{inj},t} \leftarrow \sqrt{\sigma_{t-1}^2 - \sigma_{\mathrm{ret},t}^2}$
  $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{z}_{t-1} \leftarrow \mathbf{z}_t + \Delta_t + (\sigma_{\mathrm{ret},t} - \sigma_t)\hat{\boldsymbol{\epsilon}}_\theta + \sigma_{\mathrm{inj},t}\boldsymbol{\xi}$
end for
return $\mathbf{x}_0 \leftarrow D(\mathbf{z}_0)$

B. Theoretical details

This appendix collects proofs for the propositions stated in Sec. 3. We include intermediate steps explicitly so the derivations can be checked mechanically. We use the Euclidean inner product $\langle \mathbf{a}, \mathbf{b}\rangle := \mathbf{a}^\top \mathbf{b}$ and norm $\|\mathbf{a}\|_2 := \sqrt{\mathbf{a}^\top \mathbf{a}}$. For a symmetric positive definite matrix $\mathbf{Q}$, we denote the largest and smallest eigenvalues by $\lambda_{\max}(\mathbf{Q})$ and $\lambda_{\min}(\mathbf{Q})$, and the condition number by $\kappa(\mathbf{Q}) := \lambda_{\max}(\mathbf{Q})/\lambda_{\min}(\mathbf{Q})$.

B.1. Proof of Proposition 3.1

Proposition 3.1. Fix t and consider a local quadratic model in the increment Δ:

$$\mathcal{J}(\Delta) = \mathcal{J}(\mathbf{0}) + \langle \mathbf{g}_t, \Delta\rangle + \frac{1}{2}\Delta^\top \mathbf{Q}_t \Delta, \qquad \mathbf{Q}_t \succ 0. \tag{36}$$

Consider the scalar-guided update $\Delta_{\mathrm{sc}} = -\eta \mathbf{g}_t$ with $\eta \geq 0$. Then $\mathcal{J}(\Delta_{\mathrm{sc}}) \leq \mathcal{J}(\mathbf{0})$ for all $\mathbf{g}_t$ if and only if $\eta \leq 2/\lambda_{\max}(\mathbf{Q}_t)$. Let $\Delta^\star = -\mathbf{Q}_t^{-1}\mathbf{g}_t$ be the minimizer of (36). In an eigenbasis $\mathbf{Q}_t\mathbf{v}_i = \lambda_i \mathbf{v}_i$, for each index i with $\langle \mathbf{v}_i, \mathbf{g}_t\rangle \neq 0$,

$$\frac{|\langle \mathbf{v}_i, \Delta_{\mathrm{sc}}\rangle|}{|\langle \mathbf{v}_i, \Delta^\star\rangle|} = \eta\lambda_i \leq \frac{2\lambda_i}{\lambda_{\max}(\mathbf{Q}_t)}, \tag{37}$$

and in particular for $\lambda_i = \lambda_{\min}(\mathbf{Q}_t)$, $\frac{|\langle \mathbf{v}_{\min}, \Delta_{\mathrm{sc}}\rangle|}{|\langle \mathbf{v}_{\min}, \Delta^\star\rangle|} \leq \frac{2}{\kappa(\mathbf{Q}_t)}$.

Proof. We prove three claims: (i) the objective change formula under Δ_{sc}; (ii) the (necessary and sufficient) descent condition $\eta \leq 2/\lambda_{\max}$; (iii) the eigen-coordinate ratio (37).

1) Objective change under a scalar step. Substitute $\Delta_{\mathrm{sc}} = -\eta \mathbf{g}_t$ into (36):

$$\begin{aligned}\mathcal{J}(\Delta_{\mathrm{sc}}) - \mathcal{J}(\mathbf{0}) &= \langle \mathbf{g}_t, -\eta \mathbf{g}_t \rangle + \frac{1}{2}(-\eta \mathbf{g}_t)^\top \mathbf{Q}_t (-\eta \mathbf{g}_t) \\ &= -\eta\, \mathbf{g}_t^\top \mathbf{g}_t + \frac{\eta^2}{2}\, \mathbf{g}_t^\top \mathbf{Q}_t \mathbf{g}_t \\ &= -\eta\, \|\mathbf{g}_t\|_2^2 + \frac{\eta^2}{2}\, \mathbf{g}_t^\top \mathbf{Q}_t \mathbf{g}_t. \end{aligned} \tag{38}$$

2) Rayleigh quotient bound. Because $\mathbf{Q}_t$ is symmetric positive definite, there exists an orthonormal eigenbasis $\{\mathbf{v}_i\}_{i=1}^d$ and eigenvalues $\{\lambda_i\}_{i=1}^d$ with $\lambda_i > 0$ and $\mathbf{Q}_t \mathbf{v}_i = \lambda_i \mathbf{v}_i$. Expand $\mathbf{g}_t$ as

$$\mathbf{g}_t = \sum_{i=1}^d a_i \mathbf{v}_i, \qquad a_i := \langle \mathbf{v}_i, \mathbf{g}_t \rangle.$$

Then, using orthonormality $\mathbf{v}_i^\top \mathbf{v}_j = \delta_{ij}$,

$$\|\mathbf{g}_t\|_2^2 = \mathbf{g}_t^\top \mathbf{g}_t = \sum_{i=1}^d a_i^2,$$

and

$$\mathbf{g}_t^\top \mathbf{Q}_t \mathbf{g}_t = \left(\sum_i a_i \mathbf{v}_i \right)^\top \left(\sum_j a_j \mathbf{Q}_t \mathbf{v}_j \right) = \sum_{j=1}^d \lambda_j a_j^2.$$

Since $\lambda_j \le \lambda_{\max}(\mathbf{Q}_t)$ for all j, we obtain

$$\mathbf{g}_t^\top \mathbf{Q}_t \mathbf{g}_t = \sum_j \lambda_j a_j^2 \le \lambda_{\max}(\mathbf{Q}_t) \sum_j a_j^2 = \lambda_{\max}(\mathbf{Q}_t) \|\mathbf{g}_t\|_2^2. \tag{39}$$

3) Sufficient condition for descent. Combine (38) and (39):

$$\begin{aligned}\mathcal{J}(\Delta_{\mathrm{sc}}) - \mathcal{J}(\mathbf{0}) &\le -\eta\, \|\mathbf{g}_t\|_2^2 + \frac{\eta^2}{2} \lambda_{\max}(\mathbf{Q}_t) \|\mathbf{g}_t\|_2^2 \\ &= -\eta\, \|\mathbf{g}_t\|_2^2 \left(1 - \frac{\eta}{2} \lambda_{\max}(\mathbf{Q}_t) \right). \end{aligned} \tag{40}$$

If $0 < \eta \le 2/\lambda_{\max}(\mathbf{Q}_t)$, then $1 - \frac{\eta}{2}\lambda_{\max}(\mathbf{Q}_t) \ge 0$ and therefore $\mathcal{J}(\Delta_{\mathrm{sc}}) - \mathcal{J}(\mathbf{0}) \le 0$. Hence $\mathcal{J}(\Delta_{\mathrm{sc}}) \le \mathcal{J}(\mathbf{0})$.

4) Necessity of $\eta \le 2/\lambda_{\max}(\mathbf{Q}_t)$ for uniform descent over all $\mathbf{g}_t$. Assume $\eta > 2/\lambda_{\max}(\mathbf{Q}_t)$. Let $\mathbf{v}_{\max}$ be a unit eigenvector associated with $\lambda_{\max}(\mathbf{Q}_t)$, and choose $\mathbf{g}_t = \mathbf{v}_{\max}$. Then $\|\mathbf{g}_t\|_2^2 = 1$ and $\mathbf{g}_t^\top \mathbf{Q}_t \mathbf{g}_t = \lambda_{\max}(\mathbf{Q}_t)$. From (38),

$$\mathcal{J}(\Delta_{\mathrm{sc}}) - \mathcal{J}(\mathbf{0}) = -\eta + \frac{\eta^2}{2} \lambda_{\max}(\mathbf{Q}_t) = \eta \left(\frac{\eta}{2} \lambda_{\max}(\mathbf{Q}_t) - 1 \right) > 0,$$

since $\frac{\eta}{2}\lambda_{\max}(\mathbf{Q}_t) > 1$ by assumption. Thus the objective increases for this choice of $\mathbf{g}_t$, so $\eta > 2/\lambda_{\max}$ cannot guarantee uniform descent. This shows the condition is also necessary for a guarantee that holds for all $\mathbf{g}_t$.

5) Closed form of the quadratic minimizer $\Delta^\star$. Differentiate (36) with respect to Δ. The derivative of $\langle \mathbf{g}_t, \Delta \rangle = \mathbf{g}_t^\top \Delta$ is $\mathbf{g}_t$. For symmetric $\mathbf{Q}_t$, the derivative of $\frac{1}{2}\Delta^\top \mathbf{Q}_t \Delta$ is $\mathbf{Q}_t \Delta$. Therefore,

$$\nabla_\Delta \mathcal{J}(\Delta) = \mathbf{g}_t + \mathbf{Q}_t \Delta.$$

Setting $\nabla_\Delta \mathcal{J}(\Delta^\star) = \mathbf{0}$ gives $\mathbf{Q}_t \Delta^\star = -\mathbf{g}_t$, hence

$$\Delta^\star = -\mathbf{Q}_t^{-1} \mathbf{g}_t.$$

6) Eigen-coordinate ratio. Using the eigen-expansion $\mathbf{g}_t = \sum_i a_i \mathbf{v}_i$,

$$\Delta_{\mathrm{sc}} = -\eta \mathbf{g}_t = \sum_i (-\eta a_i)\mathbf{v}_i.$$

Also, because $\mathbf{Q}_t^{-1}\mathbf{v}_i = \frac{1}{\lambda_i}\mathbf{v}_i$,

$$\Delta^\star = -\mathbf{Q}_t^{-1}\mathbf{g}_t = -\sum_i a_i\, \mathbf{Q}_t^{-1}\mathbf{v}_i = -\sum_i \frac{a_i}{\lambda_i}\mathbf{v}_i.$$

Therefore

$$\langle \mathbf{v}_i, \Delta_{\mathrm{sc}}\rangle = -\eta a_i, \qquad \langle \mathbf{v}_i, \Delta^\star\rangle = -\frac{a_i}{\lambda_i}.$$

If $a_i = \langle \mathbf{v}_i, \mathbf{g}_t\rangle = 0$, then both components are zero and the ratio is not informative. If $a_i \neq 0$, then

$$\frac{|\langle \mathbf{v}_i, \Delta_{\mathrm{sc}}\rangle|}{|\langle \mathbf{v}_i, \Delta^\star\rangle|} = \frac{|\eta a_i|}{|a_i|/\lambda_i} = \eta\lambda_i.$$

Under the stability condition $\eta \le 2/\lambda_{\max}(\mathbf{Q}_t)$, this yields

$$\eta\lambda_i \le \frac{2\lambda_i}{\lambda_{\max}(\mathbf{Q}_t)}.$$

For $\lambda_i = \lambda_{\min}(\mathbf{Q}_t)$ the bound becomes $\eta\lambda_{\min}(\mathbf{Q}_t) \le 2/\kappa(\mathbf{Q}_t)$. □

B.2. Proof of Proposition 3.2

Proposition 3.2. Let

$$\tilde{\Phi}(\mathbf{z}) := \frac{1}{2\sigma_n^2}\|A(\mathbf{z}) - \mathbf{y}\|_2^2, \qquad \Phi_t(\mathbf{x}) := \tilde{\Phi}(F_\theta(\mathbf{x};\sigma_t)).$$

Let $\mathbf{z}_t := F_\theta(\mathbf{x}_t;\sigma_t)$, $\mathbf{g}_z := \nabla_{\mathbf{z}}\tilde{\Phi}(\mathbf{z})|_{\mathbf{z}=\mathbf{z}_t}$, and $J_F := \nabla_{\mathbf{x}} F_\theta(\mathbf{x};\sigma_t)|_{\mathbf{x}=\mathbf{x}_t}$. Then, for any increment Δ,

$$\Phi_t(\mathbf{x}_t + \Delta) = \Phi_t(\mathbf{x}_t) + \langle J_F^\top \mathbf{g}_z, \Delta\rangle + o(\|\Delta\|_2).$$

Consequently, among all increments with $\|\Delta\|_2 \le \rho$, the direction that maximizes first-order decrease of Φ_t is proportional to $-J_F^\top \mathbf{g}_z$. Under first-order residual linearization, the Gauss–Newton/Fisher curvature of Φ_t in diffusion-state coordinates is $\frac{1}{\sigma_n^2} J_F^\top J_A^\top J_A J_F$, where $J_A := \nabla_{\mathbf{z}} A(\mathbf{z})|_{\mathbf{z}=\mathbf{z}_t}$.

Proof. We prove the first-order expansion, then the steepest descent direction, and finally the GN curvature form.

1) First-order expansion of $\Phi_t(\mathbf{x}_t + \Delta)$.

Step 1: Expand the denoiser. Because $F_\theta(\cdot;\sigma_t)$ is differentiable at $\mathbf{x}_t$, by the definition of the Jacobian there exists a remainder term $\mathbf{r}_F(\Delta)$ such that

$$F_\theta(\mathbf{x}_t + \Delta;\sigma_t) = F_\theta(\mathbf{x}_t;\sigma_t) + J_F\Delta + \mathbf{r}_F(\Delta), \tag{41}$$

where $\|\mathbf{r}_F(\Delta)\|_2 = o(\|\Delta\|_2)$ as $\Delta \to \mathbf{0}$. Define $\mathbf{z}_t := F_\theta(\mathbf{x}_t;\sigma_t)$, so (41) becomes

$$F_\theta(\mathbf{x}_t + \Delta;\sigma_t) = \mathbf{z}_t + J_F\Delta + o(\|\Delta\|_2).$$

Step 2: Expand the clean-space likelihood. Because $\tilde{\Phi}$ is differentiable at $\mathbf{z}_t$, there exists a remainder $r_{\tilde{\Phi}}(\mathbf{u})$ such that

$$\tilde{\Phi}(\mathbf{z}_t + \mathbf{u}) = \tilde{\Phi}(\mathbf{z}_t) + \langle \mathbf{g}_z, \mathbf{u}\rangle + r_{\tilde{\Phi}}(\mathbf{u}), \tag{42}$$

where $\mathbf{g}_z := \nabla_{\mathbf{z}}\tilde{\Phi}(\mathbf{z})|_{\mathbf{z}=\mathbf{z}_t}$ and $r_{\tilde{\Phi}}(\mathbf{u}) = o(\|\mathbf{u}\|_2)$ as $\mathbf{u} \to \mathbf{0}$.

Step 3: Compose the expansions. By definition $\Phi_t(\mathbf{x}) = \tilde{\Phi}(F_\theta(\mathbf{x};\sigma_t))$. Let

$$\mathbf{u}(\Delta) := F_\theta(\mathbf{x}_t + \Delta;\sigma_t) - \mathbf{z}_t.$$

From Step 1, $\mathbf{u}(\Delta) = J_F\Delta + o(\|\Delta\|_2)$, hence $\|\mathbf{u}(\Delta)\|_2 = O(\|\Delta\|_2)$. Apply (42) with $\mathbf{u} = \mathbf{u}(\Delta)$:

$$\begin{aligned}\Phi_t(\mathbf{x}_t + \Delta) &= \tilde{\Phi}\big(\mathbf{z}_t + \mathbf{u}(\Delta)\big) \\ &= \tilde{\Phi}(\mathbf{z}_t) + \langle \mathbf{g}_z, \mathbf{u}(\Delta)\rangle + o(\|\mathbf{u}(\Delta)\|_2).\end{aligned}$$

Substitute $\mathbf{u}(\Delta) = J_F\Delta + o(\|\Delta\|_2)$:

$$\Phi_t(\mathbf{x}_t + \Delta) = \tilde{\Phi}(\mathbf{z}_t) + \langle \mathbf{g}_z, J_F\Delta\rangle + o(\|\Delta\|_2),$$

because $\langle \mathbf{g}_z, o(\|\Delta\|_2)\rangle = o(\|\Delta\|_2)$ and $o(\|\mathbf{u}(\Delta)\|_2) = o(\|\Delta\|_2)$. Finally, use $\langle \mathbf{g}_z, J_F\Delta\rangle = \langle J_F^\top \mathbf{g}_z, \Delta\rangle$ and note $\tilde{\Phi}(\mathbf{z}_t) = \Phi_t(\mathbf{x}_t)$ to obtain

$$\Phi_t(\mathbf{x}_t + \Delta) = \Phi_t(\mathbf{x}_t) + \langle J_F^\top \mathbf{g}_z, \Delta\rangle + o(\|\Delta\|_2).$$

2) Steepest first-order decrease under $\|\Delta\|_2 \le \rho$. Let $\mathbf{w} := J_F^\top \mathbf{g}_z$. For small Δ, the first-order change is governed by $\langle \mathbf{w}, \Delta\rangle$. To maximize first-order *decrease* we minimize $\langle \mathbf{w}, \Delta\rangle$ subject to $\|\Delta\|_2 \le \rho$.

By Cauchy–Schwarz,

$$\langle \mathbf{w}, \Delta\rangle \ge -\|\mathbf{w}\|_2\,\|\Delta\|_2 \ge -\|\mathbf{w}\|_2\,\rho.$$

Equality is achieved by choosing Δ colinear with $-\mathbf{w}$ and with maximal norm:

$$\Delta^\star = -\rho\,\frac{\mathbf{w}}{\|\mathbf{w}\|_2} = -\rho\,\frac{J_F^\top \mathbf{g}_z}{\|J_F^\top \mathbf{g}_z\|_2}, \qquad (\mathbf{w} \neq \mathbf{0}).$$

Thus the steepest descent direction is proportional to $-J_F^\top \mathbf{g}_z$.

3) Gauss–Newton/Fisher curvature in diffusion-state coordinates. Define the clean residual $\mathbf{r}(\mathbf{z}) := A(\mathbf{z}) - \mathbf{y}$, so $\tilde{\Phi}(\mathbf{z}) = \frac{1}{2\sigma_n^2}\|\mathbf{r}(\mathbf{z})\|_2^2$. Let $\mathbf{r}_t := \mathbf{r}(\mathbf{z}_t) = A(\mathbf{z}_t) - \mathbf{y}$ and $J_A := \nabla_{\mathbf{z}} A(\mathbf{z})|_{\mathbf{z}=\mathbf{z}_t}$.

First-order linearization of A around $\mathbf{z}_t$ gives

$$A(\mathbf{z}_t + \mathbf{u}) \approx A(\mathbf{z}_t) + J_A\mathbf{u} \quad \Longrightarrow \quad \mathbf{r}(\mathbf{z}_t + \mathbf{u}) \approx \mathbf{r}_t + J_A\mathbf{u}.$$

From Step 1, the clean-space perturbation induced by Δ is $\mathbf{u} \approx J_F\Delta$ (up to higher-order terms). Therefore

$$\mathbf{r}(F_\theta(\mathbf{x}_t + \Delta; \sigma_t)) = \mathbf{r}(\mathbf{z}_t + \mathbf{u}) \approx \mathbf{r}_t + J_A(J_F\Delta).$$

Substitute into $\Phi_t(\mathbf{x}_t + \Delta) = \frac{1}{2\sigma_n^2}\|\cdot\|_2^2$:

$$\begin{aligned}\Phi_t(\mathbf{x}_t + \Delta) &\approx \frac{1}{2\sigma_n^2}\,\|\mathbf{r}_t + J_A J_F\Delta\|_2^2 \\ &= \frac{1}{2\sigma_n^2}\Big(\|\mathbf{r}_t\|_2^2 + 2\langle \mathbf{r}_t,\ J_A J_F\Delta\rangle + \|J_A J_F\Delta\|_2^2\Big).\end{aligned}$$

Rewrite the cross term:

$$\langle \mathbf{r}_t,\ J_A J_F\Delta\rangle = \langle J_F^\top J_A^\top \mathbf{r}_t,\ \Delta\rangle.$$

Rewrite the quadratic term:

$$\|J_A J_F\Delta\|_2^2 = (J_A J_F\Delta)^\top (J_A J_F\Delta) = \Delta^\top J_F^\top J_A^\top J_A J_F\,\Delta.$$

Hence the quadratic form in Δ is $\frac{1}{2}\Delta^\top \left(\frac{1}{\sigma_n^2} J_F^\top J_A^\top J_A J_F\right)\Delta$, which is precisely the Gauss–Newton/Fisher curvature $\frac{1}{\sigma_n^2} J^\top J$ for $J = J_A J_F$. □

B.3. Proof of Proposition 3.3

Proposition 3.3. Consider the stochastic prior propagation step

$$\mathbf{x}_{t-1} = \mathbf{x}_t + \Delta_t + (\sigma_{\mathrm{ret},t} - \sigma_t)\,\epsilon_\theta(\mathbf{x}_t, \sigma_t) + \sigma_{\mathrm{inj},t}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{43}$$

and impose schedule variance matching

$$\sigma_{t-1}^2 = \sigma_{\mathrm{ret},t}^2 + \sigma_{\mathrm{inj},t}^2. \tag{44}$$

Define

$$\lambda_t := 1 - \frac{\sigma_{\mathrm{ret},t}}{\sigma_t}, \qquad \sigma_{\mathrm{ref}}^2 := \sigma_t \sigma_{\mathrm{ret},t},$$

and assume the fluctuation–dissipation coupling

$$\sigma_{\mathrm{inj},t}^2 = 2\lambda_t\,\sigma_{\mathrm{ref}}^2. \tag{45}$$

Then, under a decreasing nonnegative schedule $0 \le \sigma_{t-1} \le \sigma_t$ with $\sigma_t > 0$, the pair $(\sigma_{\mathrm{ret},t}, \sigma_{\mathrm{inj},t})$ is uniquely determined by

$$\sigma_{\mathrm{ret},t} = \sigma_t - \sqrt{\sigma_t^2 - \sigma_{t-1}^2}, \qquad \sigma_{\mathrm{inj},t} = \sqrt{\sigma_{t-1}^2 - \sigma_{\mathrm{ret},t}^2}.$$

Proof. The proof is algebraic: we eliminate $\sigma_{\mathrm{inj},t}$ using (44) and (45), then solve for $\sigma_{\mathrm{ret},t}$.

1) Expand the coupling (45). Substitute the definitions of λ_t and σ_{ref}^2 into (45):

$$\begin{aligned}
\sigma_{\mathrm{inj},t}^2 &= 2\left(1 - \frac{\sigma_{\mathrm{ret},t}}{\sigma_t}\right)(\sigma_t \sigma_{\mathrm{ret},t}) \\
&= 2\left(\sigma_t \sigma_{\mathrm{ret},t} - \sigma_{\mathrm{ret},t}^2\right) \\
&= 2(\sigma_t - \sigma_{\mathrm{ret},t})\,\sigma_{\mathrm{ret},t}.
\end{aligned} \tag{46}$$

2) Eliminate $\sigma_{\mathrm{inj},t}$ using variance matching. From (44),

$$\sigma_{t-1}^2 = \sigma_{\mathrm{ret},t}^2 + \sigma_{\mathrm{inj},t}^2.$$

Substitute (46):

$$\begin{aligned}
\sigma_{t-1}^2 &= \sigma_{\mathrm{ret},t}^2 + 2(\sigma_t - \sigma_{\mathrm{ret},t})\,\sigma_{\mathrm{ret},t} \\
&= \sigma_{\mathrm{ret},t}^2 + 2\sigma_t \sigma_{\mathrm{ret},t} - 2\sigma_{\mathrm{ret},t}^2 \\
&= 2\sigma_t \sigma_{\mathrm{ret},t} - \sigma_{\mathrm{ret},t}^2.
\end{aligned} \tag{47}$$

Rearrange to obtain a quadratic equation in $\sigma_{\mathrm{ret},t}$:

$$\sigma_{\mathrm{ret},t}^2 - 2\sigma_t \sigma_{\mathrm{ret},t} + \sigma_{t-1}^2 = 0. \tag{48}$$

3) Solve the quadratic. Apply the quadratic formula to (48):

$$\sigma_{\mathrm{ret},t} = \frac{2\sigma_t \pm \sqrt{(2\sigma_t)^2 - 4\sigma_{t-1}^2}}{2} = \sigma_t \pm \sqrt{\sigma_t^2 - \sigma_{t-1}^2}.$$

Thus the two algebraic candidates are

$$\sigma_{\mathrm{ret},t}^{(\pm)} = \sigma_t \pm \sqrt{\sigma_t^2 - \sigma_{t-1}^2}. \tag{49}$$

The square root is real because $\sigma_t^2 - \sigma_{t-1}^2 \ge 0$ under $\sigma_{t-1} \le \sigma_t$.

4) Select the admissible root. Since $\sigma_{\mathrm{ret},t}$ and $\sigma_{\mathrm{inj},t}$ are standard deviations, they must satisfy

$$\sigma_{\mathrm{ret},t} \geq 0, \qquad \sigma_{\mathrm{inj},t}^2 = \sigma_{t-1}^2 - \sigma_{\mathrm{ret},t}^2 \geq 0 \quad \Longrightarrow \quad \sigma_{\mathrm{ret},t} \leq \sigma_{t-1}.$$

We check the two roots in (49).

(a) The "plus" root is inadmissible when $\sigma_{t-1} < \sigma_t$. If $\sigma_{t-1} < \sigma_t$, then $\sqrt{\sigma_t^2 - \sigma_{t-1}^2} > 0$ and

$$\sigma_{\mathrm{ret},t}^{(+)} = \sigma_t + \sqrt{\sigma_t^2 - \sigma_{t-1}^2} > \sigma_t > \sigma_{t-1},$$

so it violates $\sigma_{\mathrm{ret},t} \leq \sigma_{t-1}$ and cannot satisfy variance matching with a real $\sigma_{\mathrm{inj},t}$.

(b) The "minus" root is admissible. Define

$$\sigma_{\mathrm{ret},t} := \sigma_{\mathrm{ret},t}^{(-)} = \sigma_t - \sqrt{\sigma_t^2 - \sigma_{t-1}^2}. \tag{50}$$

Nonnegativity. Because $\sigma_t^2 - \sigma_{t-1}^2 \leq \sigma_t^2$ and $\sigma_t \geq 0$, we have $\sqrt{\sigma_t^2 - \sigma_{t-1}^2} \leq \sigma_t$, hence $\sigma_{\mathrm{ret},t} \geq 0$.

Upper bound $\sigma_{\mathrm{ret},t} \leq \sigma_{t-1}$. The inequality $\sigma_{\mathrm{ret},t} \leq \sigma_{t-1}$ is equivalent to

$$\sigma_t - \sqrt{\sigma_t^2 - \sigma_{t-1}^2} \leq \sigma_{t-1} \quad \Longleftrightarrow \quad \sqrt{\sigma_t^2 - \sigma_{t-1}^2} \geq \sigma_t - \sigma_{t-1}.$$

Both sides are nonnegative because $\sigma_t \geq \sigma_{t-1} \geq 0$, so we can square without changing the inequality:

$$\begin{aligned} \sigma_t^2 - \sigma_{t-1}^2 &\geq (\sigma_t - \sigma_{t-1})^2 \\ &= \sigma_t^2 - 2\sigma_t\sigma_{t-1} + \sigma_{t-1}^2. \end{aligned}$$

Cancel σ_t^2 on both sides and rearrange:

$$-\sigma_{t-1}^2 \geq -2\sigma_t\sigma_{t-1} + \sigma_{t-1}^2 \quad \Longleftrightarrow \quad 2\sigma_t\sigma_{t-1} \geq 2\sigma_{t-1}^2 \quad \Longleftrightarrow \quad \sigma_{t-1}(\sigma_t - \sigma_{t-1}) \geq 0,$$

which holds because $\sigma_{t-1} \geq 0$ and $\sigma_t - \sigma_{t-1} \geq 0$. Therefore $\sigma_{\mathrm{ret},t} \leq \sigma_{t-1}$.

Thus the minus root is admissible. Under a strict decrease $\sigma_{t-1} < \sigma_t$, it is the only admissible root, hence $\sigma_{\mathrm{ret},t}$ is uniquely determined.

5) Recover $\sigma_{\mathrm{inj},t}$. Once $\sigma_{\mathrm{ret},t}$ is fixed, variance matching (44) gives

$$\sigma_{\mathrm{inj},t}^2 = \sigma_{t-1}^2 - \sigma_{\mathrm{ret},t}^2.$$

The right-hand side is nonnegative because $\sigma_{\mathrm{ret},t} \leq \sigma_{t-1}$, so define

$$\sigma_{\mathrm{inj},t} := \sqrt{\sigma_{t-1}^2 - \sigma_{\mathrm{ret},t}^2},$$

taking the principal (nonnegative) square root. □

Table 3. Quantitative comparison on FFHQ and ImageNet across inverse-problem tasks in latent setting. We report average PSNR/SSIM (higher is better), LPIPS (lower is better), and run-time (lower is better) over 100 validation images. Best and second-best scores are highlighted in **bold** and underlined, respectively.

Task	Method	FFHQ				ImageNet			
		PSNR ↑	SSIM ↑	LPIPS ↓	Run-time (s)	PSNR ↑	SSIM ↑	LPIPS ↓	Run-time (s)
Super resolution 4×	Ours	28.933	**0.829**	**0.233**	**34.255**	**26.465**	**0.726**	**0.335**	**37.353**
	PSLD	24.627	0.628	0.377	69.290	16.656	0.291	0.541	102.757
	ReSample	23.317	0.456	0.507	300.061	22.152	0.423	0.470	269.078
	LatentDAPS	**29.204**	0.825	0.272	84.460	26.189	0.702	0.388	86.315
Inpaint (box)	Ours	**25.151**	0.837	0.236	**35.369**	20.836	0.731	0.342	**37.393**
	PSLD	23.461	**0.842**	**0.118**	68.346	20.906	**0.802**	**0.188**	96.722
	ReSample	19.978	0.796	0.247	296.326	18.087	0.713	0.309	265.139
	LatentDAPS	23.474	0.742	0.369	85.823	**22.818**	0.561	0.543	88.725
Inpaint (random)	Ours	**31.433**	**0.894**	**0.193**	**35.263**	**28.181**	**0.806**	0.262	**37.396**
	PSLD	25.652	0.697	0.368	67.614	21.977	0.574	0.473	100.200
	ReSample	29.950	0.842	0.201	307.874	26.916	0.756	**0.255**	323.283
	LatentDAPS	26.036	0.743	0.385	85.840	19.630	0.588	0.522	86.092
Gaussian deblurring	Ours	**28.269**	**0.801**	**0.272**	**35.011**	**25.443**	**0.662**	**0.411**	**38.110**
	PSLD	22.015	0.503	0.444	70.098	19.591	0.329	0.555	109.263
	ReSample	26.357	0.662	0.329	355.885	23.530	0.497	0.439	338.281
	LatentDAPS	25.717	0.732	0.384	87.407	22.695	0.567	0.549	86.016
Motion deblurring	Ours	**29.959**	**0.840**	**0.244**	**36.185**	**27.119**	**0.730**	**0.348**	**38.118**
	PSLD	24.416	0.603	0.346	70.041	20.761	0.376	0.518	98.400
	ReSample	28.744	0.754	0.262	347.756	24.845	0.579	0.404	347.756
	LatentDAPS	26.646	0.757	0.361	87.409	23.557	0.592	0.513	89.050
Phase retrieval	Ours	**28.194**	**0.802**	**0.271**	**133.941**	**20.133**	**0.483**	**0.458**	**119.078**
	ReSample	24.676	0.606	0.412	320.911	16.913	0.320	0.608	319.601
	LatentDAPS	23.199	0.692	0.421	177.730	17.067	0.446	0.624	192.787
Nonlinear deblur	Ours	**29.243**	**0.836**	0.243	**146.718**	25.889	**0.716**	0.325	**158.452**
	ReSample	28.748	0.797	**0.236**	843.212	**26.047**	0.697	**0.301**	686.128
	LatentDAPS	25.152	0.726	0.387	194.014	22.516	0.568	0.530	198.074
High dynamic range	Ours	**26.245**	0.816	0.279	**134.060**	**28.885**	**0.887**	**0.130**	**117.544**
	ReSample	25.038	**0.822**	**0.239**	291.372	24.950	0.783	0.257	273.695
	LatentDAPS	20.789	0.630	0.512	174.976	19.394	0.469	0.641	180.929

C. Experimental Details.

C.1. Implementation details.

We use natural-image pre-trained diffusion priors in both pixel and latent spaces. Pixel-space priors are a FFHQ model (Chung et al., 2023a) and an ImageNet model (Dhariwal & Nichol, 2021). Latent-space priors use an unconditional LDM-VQ4 model (Rombach et al., 2022) for both FFHQ and ImageNet. For a fair comparison, we apply the same priors to all baselines. We follow (Karras et al., 2022) for the time-step discretization and noise schedule.

Datasets and metrics. We evaluate on FFHQ (256×256) (Karras et al., 2019) and ImageNet (256×256) (Deng et al., 2009), using 100 images from the validation set for each dataset. We report standard distortion and perceptual metrics, including PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), along with qualitative reconstructions.

Baselines. We evaluate our method against representative state-of-the-art baselines capable of handling noisy measurement settings in both pixel and latent domains. For pixel-space, we compare our approach with DAPS (Zhang et al., 2025), SITCOM (Alkhouri et al., 2025), DMPlug (Wang et al., 2024), and DCDP (Li et al., 2024). For latent-space, we benchmark against Latent DAPS, ReSample (Song et al., 2024), PSLD (Rout et al., 2023).

Table 4. Diffusion steps (T) and GMRES iterations (K) used for each task in pixel and latent spaces.

	Parameter	Linear tasks	Phase retrieval	Nonlinear deblur	High dynamic range
Pixel	T	50	250	50	250
	K	5	4	20	4
Latent	T	25	250	75	250
	K	20	5	20	5

All experiments were conducted on a single NVIDIA RTX 6000 Ada Generation GPU with batch size 1. Unless stated otherwise, we set the noise standard deviation to $\sigma_n = 0.01$ and the identity-damping parameter to $\lambda_{\mathrm{id}} = 2.0$ for all tasks. Following the DAPS setting, we assume that the true measurement noise level is unknown at test time; therefore, while measurements are generated with a larger noise level (e.g., $\beta_{\mathbf{y}} = 0.05$), we use a smaller assumed noise standard deviation $\sigma_n = 0.01$ in our method. We set the number of diffusion steps T and the number of GMRES iterations K as summarized in Table 4.

C.2. Baseline Details.

DAPS (Zhang et al., 2025) We used the provided code in the official DAPS repository (code). For both `DAPS` and `LatentDAPS`, we followed the DAPS paper setting and performed reconstruction via Langevin Dynamics.

SITCOM (Alkhouri et al., 2025) We used the default configuration provided in the official SITCOM repository (code).

DMPlug (Wang et al., 2024) We used the default configuration of DMPlug (code). Specifically, we employed DDIM with $T = 3$ reverse steps and $\eta = 0$ for all tasks, and optimized the latent seed using Adam with a learning rate of 0.01. The maximum number of optimization iterations was set to 5000 for linear tasks and 10000 for nonlinear tasks. For early stopping, we adopted the same ES-WMV (Wang et al., 2023) criterion as DMPlug, using a window size of $W = 10$ and patience $P = 100$ for linear tasks, and $W = 50$ and $P = 300$ for nonlinear tasks.

DCDP (Li et al., 2024). We used the official implementation (code). Following the DCDP paper, linear tasks were evaluated in the noiseless setting. For noisy linear experiments, we followed the paper's recommendation to reduce the learning rate by a factor of 10 and to increase the ending timestep t_k. For inpainting, keeping the original learning rate performed better; under noise level $\beta_{\mathbf{y}} = 0.05$, $t_k = 100$ gave the best performance and was used. For nonlinear tasks, we used the default configuration.

Table 5. **DCDP hyperparameters for noisy linear tasks.** SR: super-resolution, BIP: box inpainting, RIP: random inpainting, GB: Gaussian blur, MB: motion blur.

Param (DCDP)	**SR**	**BIP**	**RIP**	**GB**	**MB**
t_k	100	100	100	100	100
α	10^2	10^3	10^3	10^5	10^5

ReSample (Song et al., 2024) We used the default configuration from the official ReSample repository (code).

PSLD (Rout et al., 2023) We used the official PSLD implementation (code). However, the provided configuration was unstable in our setup; therefore, we tuned (η, γ) based on FFHQ and applied the task-specific settings below.

Table 6. **PSLD hyperparameters used in our experiments (FFHQ-based tuning).** SR: super-resolution, BIP: box inpainting, RIP: random inpainting, GB: Gaussian blur, MB: motion blur.

Param (PSLD)	**SR**	**BIP**	**RIP**	**GB**	**MB**
η	1.0	0.1	0.1	0.1	0.1
γ	0.1	10^{-3}	10^{-2}	10^{-3}	10^{-2}

Table 7. We evaluate four design choices—noise-level transition (variance-preserving Langevin vs. ODE-style update), prior-aligned damping, curvature operator (one-sided, without $J_F^\top$, and full curvature with J_F), and linear solver (GMRES vs. BiCG/CG)—on super-resolution and nonlinear deblurring. Compute is matched across variants by fixing the number of diffusion steps and the number of Krylov/solver iterations.

Variant	Super resolution				Nonlinear deblur			
	PSNR	SSIM	LPIPS	Time	PSNR	SSIM	LPIPS	Time
Vanilla	22.982	0.390	0.496	1.696	22.644	0.532	0.364	18.294
w/ODE sampling	28.767	0.813	0.230	6.346	28.997	0.818	0.194	30.106
w/o damping	26.391	0.628	0.414	6.291	29.976	0.823	0.177	31.053
w/o $J_F^\top$	28.501	0.758	0.334	1.669	24.848	0.643	0.339	18.127
w/ J_F	29.798	0.841	0.227	31.985	30.420	0.843	0.170	132.596
GMRES→BiCG	29.806	0.841	0.228	11.107	26.796	0.740	0.284	74.102
Proposed	29.807	0.841	0.229	6.370	30.524	0.859	0.170	30.286

D. Additional Experiments

Latent-space experiments. Table 3 summarizes latent-space inverse-problem results on FFHQ and ImageNet. Overall, our method consistently achieves strong reconstruction quality while being markedly faster than prior latent-space baselines, indicating that the proposed acceleration does not sacrifice fidelity. Notably, we remain robust on blur families and also handle the notoriously ill-posed phase retrieval task reliably in the latent setting. We additionally observe particularly large gains on high dynamic range reconstruction, where our method shows a clear margin over competing approaches across datasets. These results suggest that our latent-space solver generalizes well across diverse degradations, including highly ill-conditioned regimes, while maintaining practical runtimes.

Ablation Studies. To justify our design choices, we conduct a targeted study of four components: (i) how we transition to the next noise level, (ii) whether prior-aligned damping is applied, (iii) the effect of the one-sided curvature approximation, and (iv) the choice of linear solver. For the noise-level transition, we compare our default variance-preserving Langevin propagation against an EDM-style ODE update, where no additional stochastic noise is injected and the iterate is propagated solely using the model prediction. To evaluate the one-sided curvature approximation, we consider three curvature operators. Our default uses the one-sided form $J_F^\top J_A^\top(J_A\,\cdot)$. Removing the one-sided approximation yields $J_A^\top(J_A\,\cdot)$, while the full form corresponds to $J_F^\top J_A^\top(J_A(J_F\,\cdot))$. We compare these three variants in a controlled setting. For solving the resulting linear systems, our default employs GMRES. As a variant, we compare against BiCG, a Krylov alternative for nonsymmetric systems—relevant here because the one-sided approximation makes the system matrix B nonsymmetric. We additionally include a plain CG solver as a further baseline. Finally, we compare against a "vanilla" configuration that uses CG with $J_A^\top(J_A\,\cdot)$, removes damping, and adopts the ODE update. For a fair comparison, we equalize compute across methods by matching both the number of solver steps and the number of diffusion steps.

Across a representative linear task (super-resolution) and a nonlinear task (nonlinear deblurring), we observe consistent trends. First, replacing the proposed sampling transition with an ODE-style update degrades performance. Second, removing prior-aligned damping underperforms in both settings, indicating that the damping is important beyond the specific task type. Third, dropping the $J_F^\top$ factor reduces runtime by approximately $3.8\times$ on super-resolution, but incurs a clear performance trade-off, most notably on nonlinear deblurring. Conversely, using the exact curvature by including J_F increases runtime by about $5.0\times$ on super-resolution and $4.4\times$ on nonlinear deblurring, while providing nearly identical or only marginally different performance. These results indicate that the one-sided approximation achieves a more favorable accuracy–efficiency trade-off, preserving performance at significantly lower computational cost.

Regarding the linear solver choice, CG is fundamentally mismatched to our setting since it assumes a symmetric (typically SPD) system, whereas the one-sided curvature leads to a nonsymmetric matrix B. This motivates BiCG as a more appropriate Krylov alternative for nonsymmetric systems. However, BiCG (and related CG-style variants) can still exhibit numerical instability in practice when the effective system becomes poorly conditioned, especially on challenging nonlinear tasks such as nonlinear deblurring. In particular, increasing the number of iterations may amplify nonnormal effects and round-off errors, causing the iterates to become unstable and the reconstruction quality to deteriorate. These observations further

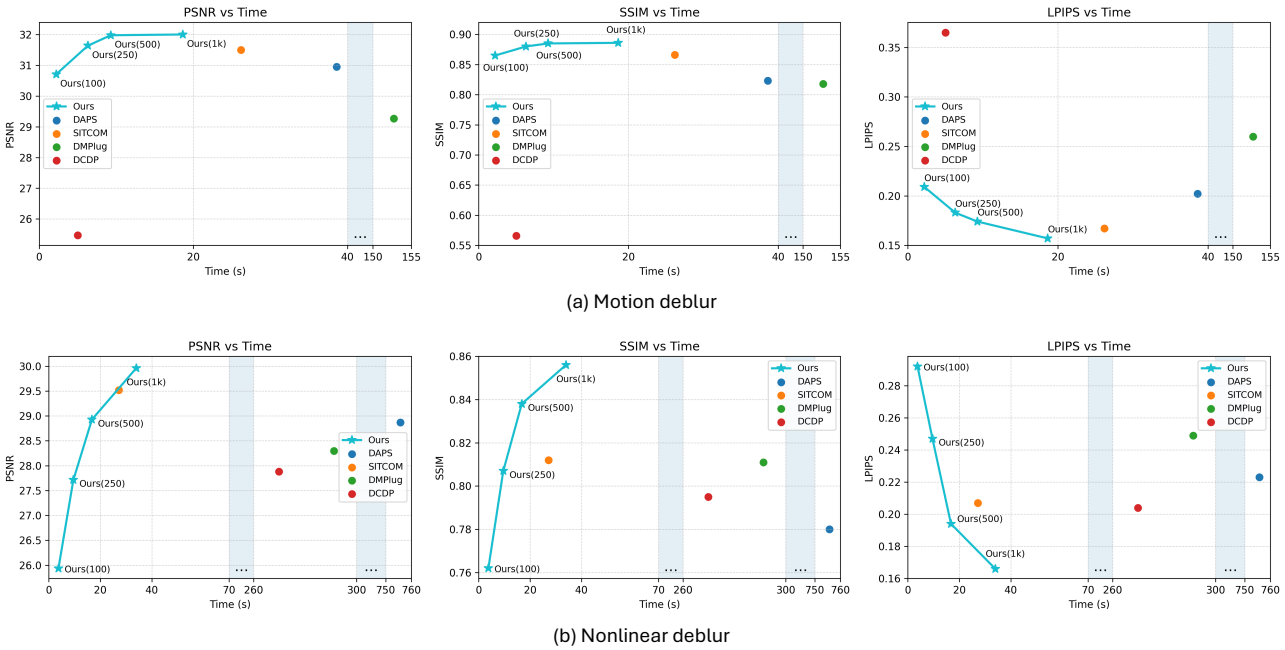

Figure 3. PSNR/SSIM/LPIPS versus wall-clock time for (a) motion deblur and (b) nonlinear deblur. Our method traces a Pareto curve as we vary the compute budget, where the budget for ours is measured by the product of the number of diffusion time steps (T) and the number of GMRES iterations (K); the evaluated (T, K) configurations are listed in Table 8.

Table 8. In Figure 3, we report the diffusion step count T and the number of GMRES iterations per step K used for CLAMP at four compute budgets (Steps $= T \times K \in \{100, 250, 500, 1\text{k}\}$) on motion deblur and nonlinear deblur.

Task	**Step**	100	250	500	1k
Motion deblur	T	20	50	50	100
	K	5	5	10	10
Nonlinear deblur	T	10	25	25	50
	K	10	10	20	20

support GMRES as a robust default for our nonsymmetric curvature operator.

Evaluation on runtime and quality. In Figure 3, we evaluate the *compute–quality* trade-off on (a) motion deblur and (b) nonlinear deblur by plotting PSNR/SSIM/LPIPS against wall-clock time. Overall, CLAMP exhibits a favorable Pareto frontier, achieving improvements across all metrics(PSNR/SSIM/LPIPS) at comparable or lower runtime.

High-Resolution Experiments (512×512). To assess the scalability of our method to higher resolutions, we conduct additional experiments at 512×512 using the publicly available ImageNet class-conditional diffusion prior (Dhariwal & Nichol, 2021). Since this prior requires a class label, we obtain a label for each validation image by running a fixed pretrained ImageNet classifier once offline and treating the predicted label as a constant condition thereafter. Importantly, the classifier is not used during sampling (i.e., we do not employ classifier guidance); the sampling procedure only queries the class-conditional diffusion model with the precomputed label for each image. Under this protocol, we compare our method against DAPS and SITCOM on one representative linear task and one representative nonlinear task, using the same 512×512 diffusion prior and the same precomputed labels for all methods. Table 9 shows that our method remains competitive in reconstruction quality at higher resolution while providing substantial speedups over baselines: it is 4.57× faster than SITCOM on random inpainting and 2.65× faster than DAPS on HDR, while maintaining higher PSNR/SSIM and lower LPIPS compared to competing methods. Reconstruction samples are provided in Figure 4, demonstrating that our method reconstructs images close to the originals while preserving fine details and naturalness even at higher resolutions.

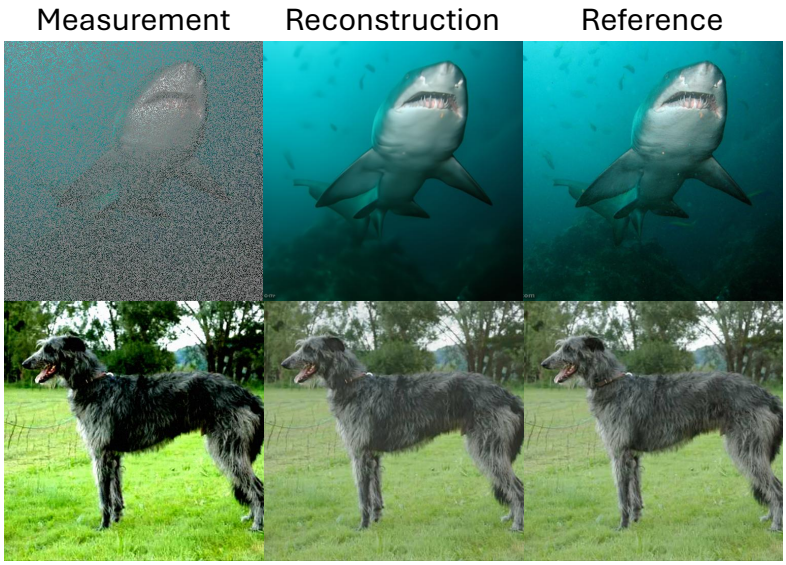

Figure 4. Visualizations at 512×512 resolution on ImageNet. **Top:** super-resolution. **Bottom:** high dynamic range (HDR).

Table 9. 512×512 ImageNet results with a class-conditional diffusion prior (offline predicted labels; no classifier guidance). Random Inpainting (linear) and HDR (nonlinear): PSNR/SSIM/LPIPS and run-time. Best/second-best are **bold**/underlined (10 samples, batch size 1).

Method	Random Inpainting				HDR			
	PSNR ↑	SSIM ↑	LPIPS ↓	Run-time (s)	PSNR ↑	SSIM ↑	LPIPS ↓	Run-time (s)
Ours	**35.646**	**0.920**	**0.212**	**40.897**	**28.845**	**0.858**	**0.192**	**176.777**
DAPS	31.736	0.761	0.313	119.487	28.003	0.815	0.255	469.332
SITCOM	34.047	0.868	0.230	187.028	25.361	0.712	0.322	254.260

Hyperparameter analysis. We analyze the identity-damping parameter λ_{id} in the prior-aligned metric $\mathbf{H}_t = \lambda_{\text{id}}\mathbf{I} + \mathbf{u}_t\mathbf{u}_t^\top$ used by the correction system $(\mathbf{B}_t + \alpha_t\mathbf{H}_t)\boldsymbol{\Delta}_t = -\mathbf{c}_t$. Here $\mathbf{B}_t$ denotes the one-sided GN/Fisher curvature operator and α_t is the diffusion-calibrated scaling that adjusts the overall damping strength across noise levels. The isotropic term $\lambda_{\text{id}}\mathbf{I}$ provides baseline regularization in all directions (and is the only regularization on $\mathbf{u}_t^\perp$, where the rank-one term vanishes), while $\mathbf{u}_t\mathbf{u}_t^\top$ adds additional damping along the denoiser-residual direction.

Table 10 shows that the effect of λ_{id} is task dependent. For random inpainting, which is intrinsically one-to-many, smaller λ_{id} yields substantially better perceptual quality (lower LPIPS), consistent with allowing more freedom in $u_t^\perp$ while maintaining measurement consistency; larger λ_{id} progressively over-regularizes the correction and degrades all metrics. For nonlinear blur, moderate λ_{id} improves stability and achieves the best overall metrics in this sweep (we use $\lambda_{\text{id}} = 2$), whereas very large λ_{id} again over-regularizes the update and leads to under-correction and worse performance.

Severe degradation settings. In Figure 5, we visualize sample diversity under severe degradation on FFHQ 256×256. We consider two highly ill-posed settings: $16\times$ super-resolution (top two rows) and 192×192 box inpainting (bottom two rows). For each measurement, we draw three independent runs (Sample1–3). Despite the measurement being heavily corrupted or largely missing, our method produces natural-looking, measurement-consistent reconstructions while exhibiting meaningful diversity across runs (e.g., facial attributes and expressions), suggesting effective sampling of a multi-modal solution space.

Various measurement noise levels. We further test robustness by varying only the measurement noise model/level while keeping the forward operator and compute budget fixed. For Gaussian corruption, we add i.i.d. additive noise in the measurement domain with $\sigma \in \{0.01, 0.05, 0.1\}$, where $\sigma = 0.05$ is used as our default setting. For Poisson corruption, we follow the Poisson simulation protocol of DPS (Chung et al., 2023a): measurements are interpreted as photon counts proportional to discretized intensity, noisy counts are sampled from a Poisson distribution, and the result is rescaled back to

Table 10. Effect of the identity-damping parameter λ_{id}: we sweep λ_{id} and report PSNR/SSIM/LPIPS on random inpainting and nonlinear blur.

λ_{id}	Random Inpainting			Nonlinear Blur		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
0	33.081	0.897	0.144	30.012	0.825	0.175
0.1	33.165	0.899	0.144	30.048	0.828	0.173
2	33.249	0.901	0.202	30.524	0.859	0.170
10	31.873	0.863	0.285	29.654	0.842	0.218
100	27.482	0.754	0.415	26.765	0.776	0.294

Figure 5. **Sample diversity under severe degradation.** *Top two rows:* 16× superresolution. *Bottom two rows:* 192×192 box inpainting. Our method yields diverse, measurement-consistent reconstructions by effectively sampling from a multi-modal posterior.

the normalized range. Here, the reported Poisson level $\lambda \in \{3.0,\ 1.0,\ 0.5\}$ acts as a scale factor (photon-rate / gain) that modulates the expected counts, producing signal-dependent shot noise.

Across both noise families, our method remains stable without qualitative degradation and shows graceful performance decay as the noise increases, indicating that the proposed update is not overly sensitive to the assumed noise level and is reliable under a broad range of measurement conditions.

Normalized residual check. A core challenge in training-free diffusion posterior sampling is to ensure that the generated samples are not only visually plausible but also *statistically consistent* with the assumed measurement model. To this end, we evaluate a simple yet informative calibration diagnostic in measurement space.

We define the noise-normalized residual statistic as

$$r_s \ := \ \frac{1}{m} \frac{\|y - A(\mathbf{x})\|_2^2}{\beta_{\mathbf{y}}^2} \tag{51}$$

where m is the number of measurement entries. It measures the average squared misfit in units of the assumed measurement noise (assuming i.i.d. Gaussian noise with standard deviation $\beta_{\mathbf{y}}$ across measurements).

If samples are well-calibrated with respect to the likelihood under the measurement noise level $\beta_{\mathbf{y}}$, then $\|\mathbf{y} - A(\mathbf{x})\|_2^2/\beta_{\mathbf{y}}^2$ is on the order of a χ_m^2 random variable, implying that r_s should concentrate around 1. Deviations reveal systematic

Table 11. **Effect of measurement noise.** We evaluate robustness to different measurement noise levels on motion deblurring in pixel space and nonlinear deblurring in latent space. For **Gaussian** noise, we add i.i.d. noise to the measurement, $\mathbf{y} = A(\mathbf{x}) + \mathbf{n}$ with $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$, sweeping $\sigma \in \{0.01, 0.05, 0.1\}$. For **Poisson** noise, the Poisson level λ controls the effective photon budget (larger λ indicates higher expected counts and thus higher SNR).

Gaussian (σ)	**Motion deblur (pixel)**			**Nonlinear deblur (latent)**			**Poisson** (λ)	**Motion deblur (pixel)**			**Nonlinear deblur (latent)**		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
0.01	32.217	0.885	0.200	30.201	0.848	0.259	3.0	31.945	0.881	0.195	29.737	0.841	0.254
0.05 (base)	31.913	0.880	0.194	29.793	0.842	0.252	1.0	31.210	0.867	0.193	28.820	0.822	0.250
0.1	30.647	0.853	0.196	28.108	0.799	0.251	0.5	29.724	0.823	0.227	27.241	0.766	0.295

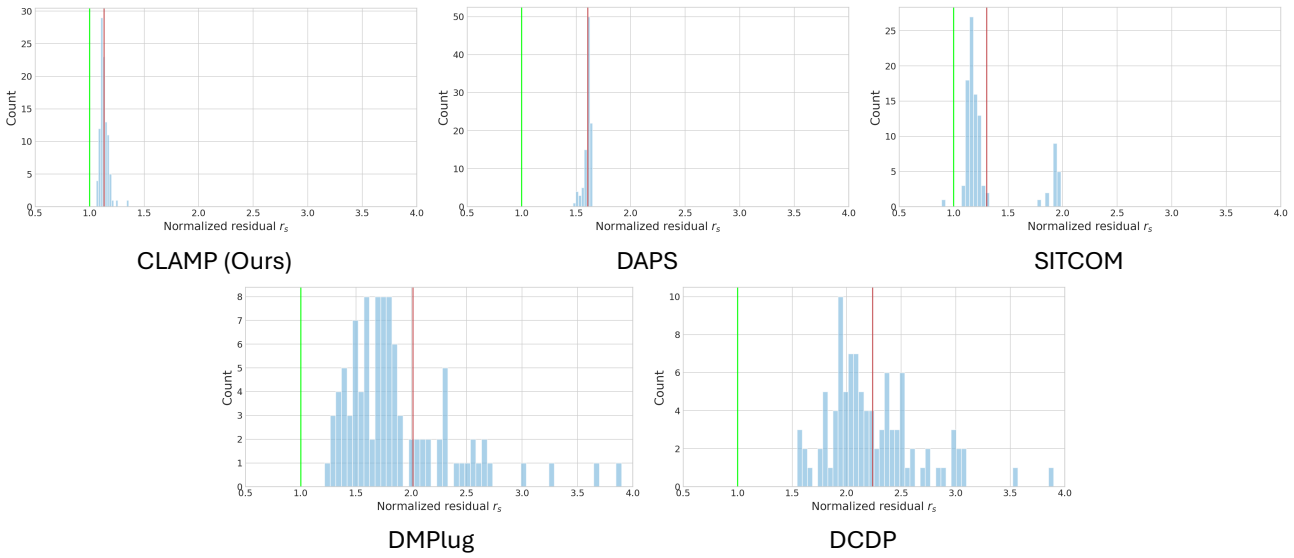

Figure 6. **Measurement-space calibration on Box Inpainting.** Histogram of the normalized measurement residual $r_s := \|\mathbf{y} - A(\mathbf{x})\|_2^2/(\beta_{\mathbf{y}}^2\, m)$ for the *box inpainting* operator, where y is the masked observation, $A(\cdot)$ applies the same box mask, $\beta_{\mathbf{y}}$ is the measurement noise standard deviation , and m is the number of measurement entries. The red line denotes the empirical mean and the green line indicates the ideal value $r_s = 1$.

mis-calibration: $r_s \gg 1$ indicates under-enforcing measurement consistency (insufficient likelihood correction), whereas $r_s \ll 1$ indicates over-enforcing data consistency (overconfident and potentially over-smoothed samples).

Figure 6 reports the empirical distribution of r_s for the box inpainting operator, and Table 12 summarizes the mean r_s across tasks on FFHQ dataset. CLAMP yields residual distributions most tightly centered near the ideal value 1 and achieves mean r_s closest to 1 on the majority of tasks, providing evidence that CLAMP produces samples that are better calibrated in measurement space under the same assumed noise level. While this diagnostic does not fully characterize posterior uncertainty, it serves as a practical sanity check that complements reconstruction metrics by directly probing sample-level consistency with the observation model.

Table 12. **Measurement-space calibration (normalized residual).** We report the mean measurement residual normalized by the measurement noise level for posterior samples across inverse problems. Values closer to 1 indicate better calibration/coverage in measurement space; systematic deviations imply over- or under-enforcing data consistency. Columns correspond to different operators/tasks (SR, BIP, RIP, GB, MB, PR, NB, HDR).

Method	**SR**	**BIP**	**RIP**	**GB**	**MB**	**PR**	**NB**	**HDR**
Ours	<u>1.272</u>	**1.131**	**1.078**	**1.030**	**1.052**	**1.045**	**1.118**	**1.412**
DAPS	1.361	1.605	1.285	<u>1.031</u>	1.071	<u>1.054</u>	1.290	2.391
SITCOM	1.364	<u>1.302</u>	<u>1.114</u>	1.041	1.059	1.056	1.243	<u>1.528</u>
DMPlug	**1.117**	2.014	1.344	1.072	1.216	-	<u>1.190</u>	3.651
DCDP	1.676	2.237	1.955	1.174	1.760	1.082	1.528	-

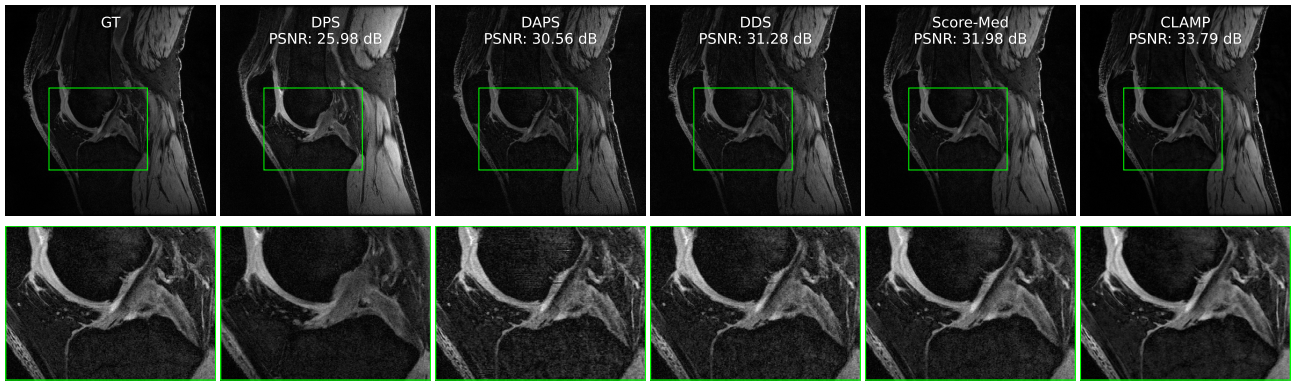

Figure 7. Reconstructions from $\times 8$ undersampled measurements are shown for various diffusion-based baselines. The top row displays the full magnitude images with their respective PSNR(dB) values. The bottom row provides zoomed-in crops of the region highlighted by the green box. Our method CLAMP demonstrates superior restoration of fine anatomical details and sharper edges compared to other baselines.

E. Qualitative MRI Results and Implementation Details

This section provides supplementary implementation details and qualitative comparisons for accelerated multi-coil MRI reconstruction. We adopt the standard linear forward model

$$\mathbf{y} = A\mathbf{x}_0 + \mathbf{n}, \qquad A = MFS,$$

where M is a k-space subsampling mask, F is the (2D) Fourier transform, and S denotes coil sensitivity maps (applied coil-wise, with measurements stacked across coils). For subsampling, we use *Poisson* (Poisson-disc) masks at the target acceleration factors.

We conduct experiments on the SKM-TEA dataset (Desai et al., 2021). We estimate sensitivity maps S using JSENSE and use SENSE for coil combination as well as to generate fully-sampled reference reconstructions from fully-sampled data. We compare against baselines, including DPS (Chung et al., 2023a), DAPS (Zhang et al., 2025), DDS (Chung et al., 2024), and Score-Med (Song et al., 2022).

Our diffusion-based prior is built upon a U-Net backbone consisting of approximately 167 million parameters, integrated within the EDM framework. The model was trained for 200,000 iterations using the RAdam (Liu et al., 2020) optimizer with a learning rate of 1×10^{-4} and a batch size of 4 on two NVIDIA RTX 4090 GPUs. The model processes complex-valued images as two-channel (real/imaginary) inputs at a resolution of 512×512.

For testing, we evaluate the model on the SKM-TEA test split, which comprises 39 3D volumes. From these volumes, we randomly sample 100 slices to conduct a robust comparative analysis of the reconstruction performance. During the inference, we set the number of diffusion steps to $T = 25$, the number of GMRES iterations to $K = 10$ and identity-damping parameter to $\lambda_{\text{id}} = 1.0$. To evaluate reconstruction fidelity, we measure PSNR and SSIM on magnitude images. These metrics are calculated per slice and averaged across the sampled test set to yield the quantitative scores. Table 2 presents the quantitative results and Figure 7 shows qualitative comparisons including reconstructed magnitude images and zoomed-in crops of anatomical structures.

F. Future work and Limitations

Our method relies on automatic differentiation to compute Jacobian actions of the forward operator, specifically VJPs and JVPs. As a result, the framework may be difficult to apply directly when the measurement process contains non-differentiable components, such as hard quantization, clipping, or codec pipelines (e.g., JPEG). While replacing such operators with differentiable surrogates provides a practical workaround, this can introduce modeling mismatch between the true measurement process and its surrogate, which may adversely affect reconstruction quality. Developing principled and robust mechanisms to handle non-differentiable or black-box forward operators remains an important direction for future research.

Furthermore, our current experimental validation focuses on 2D image data. Extending the proposed framework to

higher-dimensional modalities is a promising avenue for future work, including 3D reconstruction and video inverse problems. In these settings, scaling the curvature-guided corrections and associated linear solves to accommodate increased dimensionality—as well as exploiting spatiotemporal structure and temporal consistency—will be key challenges.

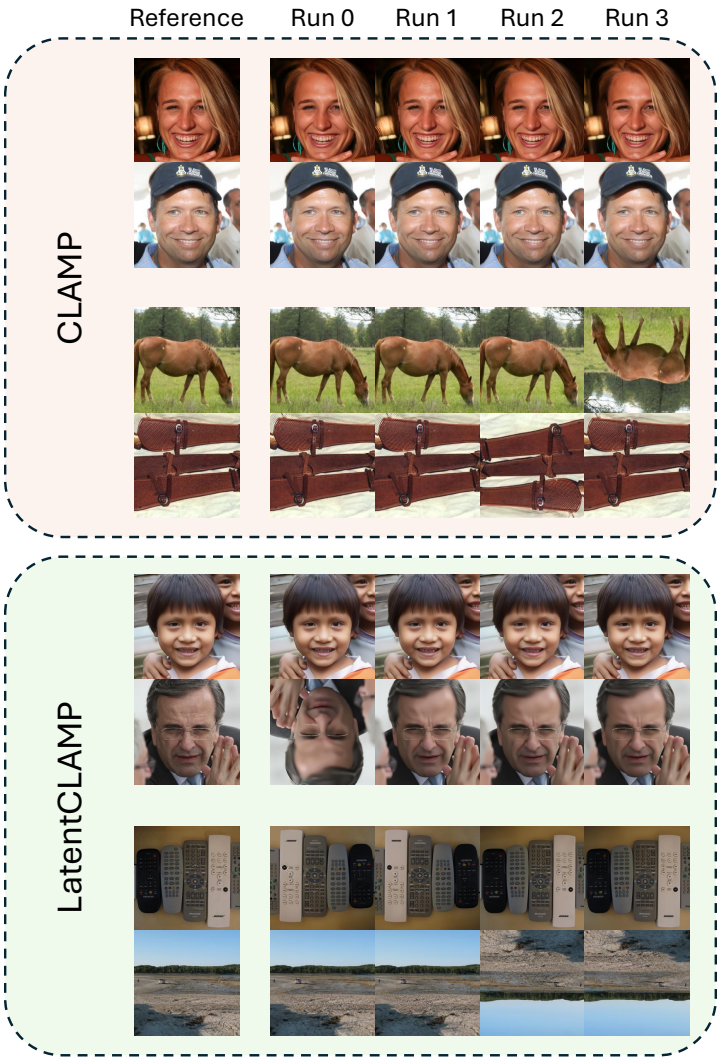

Figure 8. **Additional visualizations: phase retrieval (FFHQ and ImageNet).** Additional reconstructions on FFHQ and ImageNet for phase retrieval; for each input, we visualize all reconstructions obtained from four independent trials.

Figure 9. **Additional visualizations: super-resolution (FFHQ and ImageNet).** Additional reconstructions on FFHQ and ImageNet for super-resolution.

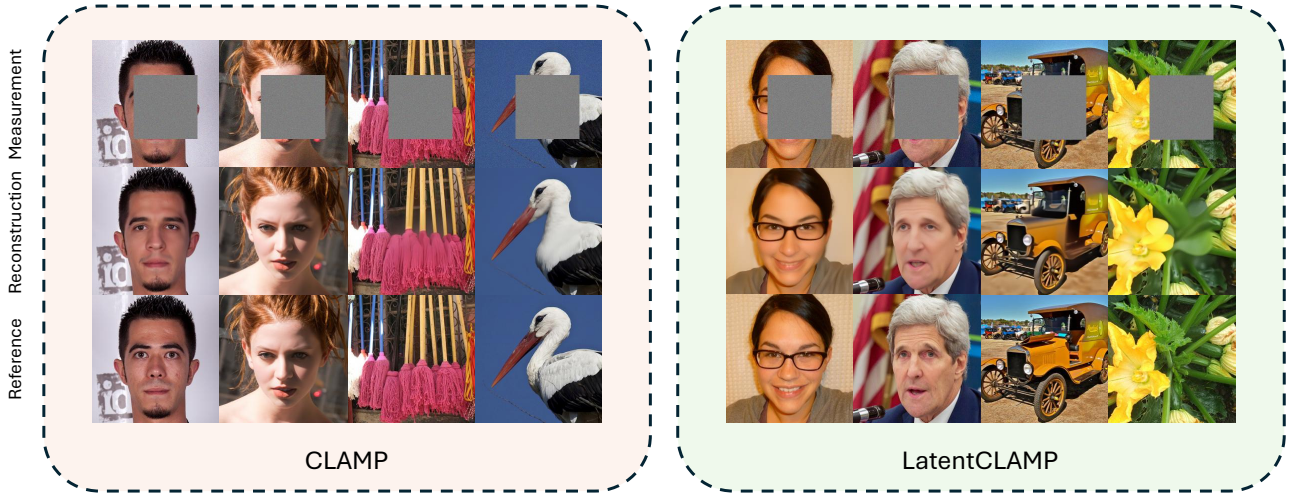

Figure 10. **Additional visualizations: box inpainting (FFHQ and ImageNet).** Additional reconstructions on FFHQ and ImageNet for box inpainting

Figure 11. **Additional visualizations: random inpainting (FFHQ and ImageNet).** Additional reconstructions on FFHQ and ImageNet for random inpainting.

Figure 12. **Additional visualizations: Gaussian deblurring (FFHQ and ImageNet).** Additional reconstructions on FFHQ and ImageNet for Gaussian deblurring.

Figure 13. **Additional visualizations: motion deblurring (FFHQ and ImageNet).** Additional reconstructions on FFHQ and ImageNet for motion deblurring.

Figure 14. **Additional visualizations: nonlinear deblurring (FFHQ and ImageNet).** Additional reconstructions on FFHQ and ImageNet for nonlinear deblurring.

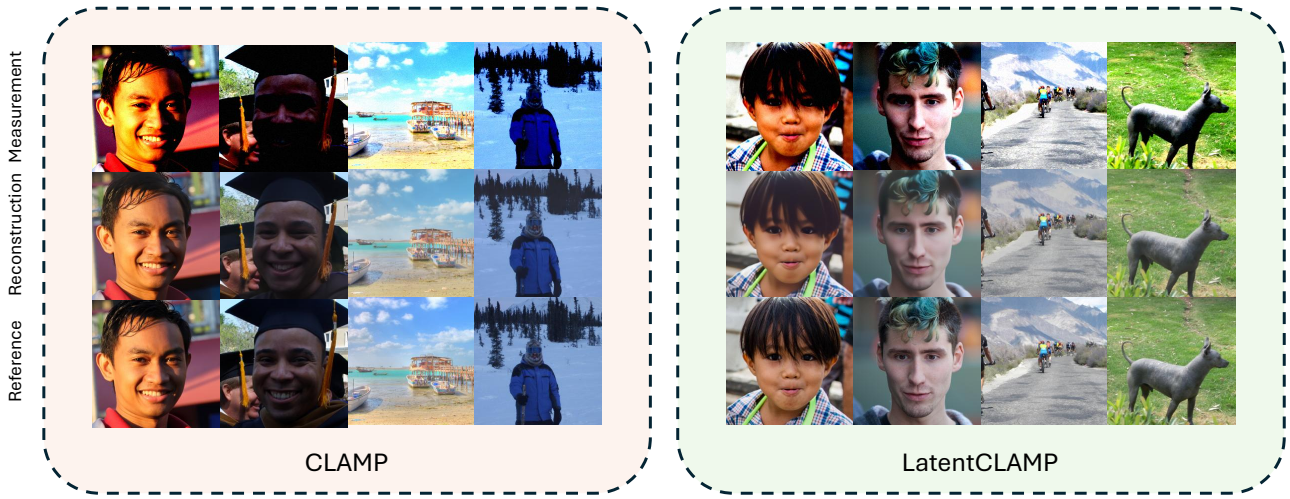

Figure 15. **Additional visualizations: high dynamic range (FFHQ and ImageNet).** Additional reconstructions on FFHQ and ImageNet for high dynamic range reconstruction.