FlowLPS: Langevin-Proximal Sampling for Measurement-Feedback-Guided Generative Inference

Anonymous Author(s)
Affiliation
Address
email

Figure 1: **Qualitative results of FlowLPS**. (a) Motion deblurring, (b) Gaussian deblurring, (c) Super-resolution ($\times 12$) and (d) random inpainting results before and after FlowLPS. (e) and (f) are results for box inpainting.

Abstract

World feedback offers measurable, non-human signals for aligning learning systems with real-world consequences. In imaging inverse problems, such feedback is available in a particularly explicit form: a known measurement process defines a physical consistency signal that evaluates whether a candidate reconstruction explains the observed measurement. This turns degradation-aware inverse solving into an inference-time alignment problem: generate samples that remain on the pretrained data prior while maximizing measurement-derived feedback. We propose **FlowLPS**, a training-free latent flow solver that uses this feedback under a finite compute budget. FlowLPS combines posterior-oriented Langevin exploration with local proximal feedback refinement: Langevin updates produce feedback-aware posterior-oriented candidates around the model trajectory, while proximal refinement rapidly improves measurement feedback from these candidates. A controlled pCN-style re-noising step stabilizes the reverse trajectory while preserving stochasticity. Experiments on FFHQ and DIV2K across five inverse problems show that FlowLPS achieves a strong balance between feedback satisfaction, measured by reconstruction fidelity, and perceptual realism, demonstrating how explicit world feedback can guide pretrained generative models without human labels or task-specific training.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

1 Introduction

Reinforcement learning from human feedback has shown that feedback signals can align powerful models with desired behavior, but many real-world tasks provide more direct non-human feedback, such as physical measurements, system metrics, safety constraints, or latency costs. Such world feedback is often objective and grounded in the consequences of an action or generated output.

Imaging inverse problems provide a clean setting for studying world-feedback-guided generative inference. The goal is to recover an underlying signal $\mathbf{x}_0$ from a noisy measurement

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

with known operator $\mathcal{A}$. Because inverse problems are ill-posed, many candidates can explain the same measurement, so a desirable reconstruction should satisfy the measurement while remaining plausible under pretrained diffusion or flow priors [1–5]. From the world-feedback perspective, the known measurement process defines an explicit feedback signal for any candidate reconstruction $\mathbf{x}$:

$$R_{\mathbf{y}}(\mathbf{x}) = -\frac{1}{2\sigma_n^2}\|\mathbf{y} - \mathcal{A}(\mathbf{x})\|^2. \tag{2}$$

This physically grounded signal measures whether the generated output explains the observed world measurement, allowing degradation-aware inverse solving to be viewed as a constrained feedback-alignment problem:

$$\text{Find} \operatorname*{arg\,max}_{\mathbf{x}\sim p_{data}} R_{\mathbf{y}}(\mathbf{x}), \tag{3}$$

or, equivalently, as posterior inference combining a pretrained data prior with a measurement-derived likelihood. Existing generative inverse solvers can be broadly divided into posterior-guided methods, which approximate measurement-conditioned posteriors [6–14], and optimization-based methods, which enforce measurement feedback through variational objectives, direct optimization, or proximal corrections [15–21].

However, using measurement feedback efficiently at inference time remains challenging in latent generative models. In models such as Stable Diffusion, the VAE decoder induces a nonlinear measurement landscape, making a single feedback correction at each reverse step often insufficient [9, 22]. Recent methods therefore strengthen measurement enforcement through hard data-consistency optimization [22], decomposition-based correction [14], or decoupled exact-consistency optimization [19]. These refinement-dominant methods can rapidly improve the feedback score, but their behavior can be initialization-sensitive and may perturb the prior-supported trajectory, leading to artifacts or degraded realism [23, 24]. Conversely, sampling-dominant methods, such as Langevin-based posterior samplers [11], better preserve stochastic exploration and prior fidelity but may require many iterations before producing sharp, feedback-satisfying reconstructions.

We propose **FlowLPS** (Langevin-Proximal Sampling), a training-free latent flow solver for inference-time alignment with measurement feedback. FlowLPS uses the world-derived feedback in two complementary ways. First, a few Langevin updates perturb the model-predicted clean estimate in posterior-oriented directions, producing stochastic candidate initializations that remain close to the generative trajectory. Second, a local MAP-style proximal refinement step rapidly improves measurement feedback from the Langevin-updated candidate. Finally, a controlled pCN-style re-noising step [25] stabilizes the reverse trajectory while retaining useful stochasticity. Together, these components provide a practical exploration–refinement balance for using explicit measurement feedback under a fixed computational budget.

Our contributions are:

- **World-feedback formulation.** We formulate degradation-aware inverse solving as inference-time alignment of generative priors using explicit measurement feedback.
- **FlowLPS.** We propose a training-free latent flow solver that balances stochastic exploration and proximal feedback refinement under a finite compute budget.
- **Finite-step analysis and ablations.** We provide a local analysis explaining why Langevin exploration and proximal refinement play complementary roles, and validate this through update-allocation ablations.
- **Empirical validation.** Experiments on FFHQ and DIV2K show that FlowLPS improves measurement-feedback satisfaction while preserving perceptual realism across diverse inverse problems.

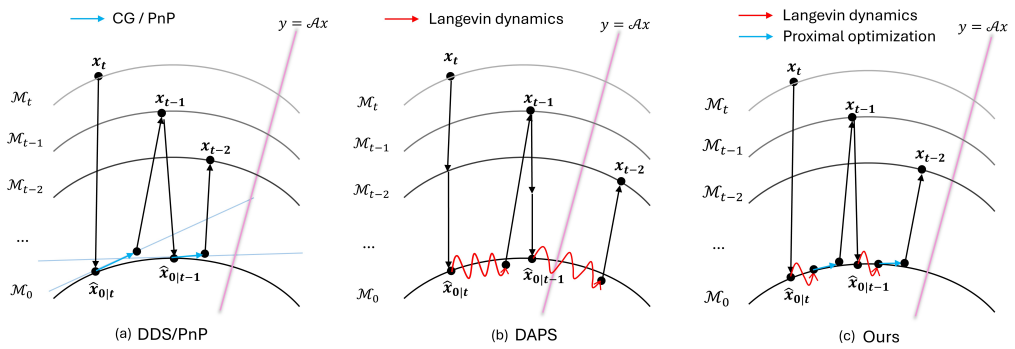

Figure 2: **Using measurement feedback under a finite inference budget.** (a) Refinement-dominant solvers repeatedly optimize the world-feedback signal $R_{\mathbf{y}}(\mathbf{x})$ but may become initialization-sensitive in nonlinear latent spaces. (b) Sampling-dominant solvers preserve stochastic exploration but can require many iterations to satisfy the feedback. (c) FlowLPS combines both: Langevin updates provide stochastic posterior-oriented candidate initializations, and proximal refinement rapidly improves measurement feedback.

2 Related Work

Generative inverse solvers. Generative inverse solvers are often discussed through two broad lines: posterior-guided inference and optimization-based correction. Posterior-guided methods steer the generative trajectory using approximations to $\nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t|\mathbf{y}) = \nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t) + \nabla_{\mathbf{z}_t} \log p(\mathbf{y}|\mathbf{z}_t)$, where $\mathbf{z}_t$ denotes an intermediate state. Since the likelihood term is generally intractable, DPS [6] and related methods [7, 9, 12] approximate this term using denoised estimates or Gaussian assumptions on $p(\mathbf{z}_0|\mathbf{z}_t)$. Later methods reduce direct guidance cost by updating the denoised estimate $\hat{\mathbf{z}}_{0|t}$, which has become a practical surrogate for posterior guidance [8, 10, 11, 13, 14]. Optimization-based methods enforce measurement feedback through explicit optimization, variational objectives, or proximal corrections [15–21]. These methods can rapidly reduce measurement error, especially with multiple refinement steps per reverse step. However, in latent-space inverse problems, decoder-induced nonlinearity makes the refinement landscape highly nonconvex, so aggressive local correction can be initialization-sensitive and may degrade perceptual realism.

Measurement-feedback regimes in latent inverse solvers. From the world-feedback perspective, latent inverse solvers differ in how they use the measurement-derived feedback signal under a finite inference budget.

Refinement-dominant solvers prioritize repeated feedback optimization through gradient refinement, Krylov/CG-style correction, or proximal updates. For example, DDS [10] performs multi-step refinement, while FLAIR [19] and Flower [21] use strong consistency or proximal updates. Such methods can quickly reduce measurement mismatch, but deterministic refinement may be initialization-dependent in nonlinear latent spaces.

In contrast, *sampling-dominant* solvers prioritize stochastic posterior exploration. A representative example is DAPS [11], which uses Langevin dynamics to sample from $p(\mathbf{z}_0|\mathbf{z}_t, \mathbf{y})$. This regime better preserves stochasticity and prior fidelity, but often requires many iterations to concentrate samples into sharp, feedback-satisfying reconstructions.

FlowLPS is motivated by this finite-step feedback-use trade-off. It uses a few Langevin updates to obtain posterior-oriented initializations, followed by local proximal refinement to improve measurement feedback. This places FlowLPS between the two regimes as an exploration–refinement hybrid.

3 Background: Measurement Feedback for Generative Inference

Latent flow priors. Flow models generate samples by transporting a source distribution p_1 to a target data distribution p_0 through a time-dependent path. For rectified flow [4], the interpolation is

$$\mathbf{z}_t = t\mathbf{z}_1 + (1-t)\mathbf{z}_0, \tag{4}$$

where $\mathbf{z}_1 \sim \mathcal{N}(0, I)$ and $\mathbf{z}_0$ is a clean latent. Given the vector field $\mathbf{v}_t$, the clean and noise estimates are

$$\hat{\mathbf{z}}_{0|t} = \mathbf{z}_t - t\mathbf{v}_t(\mathbf{z}_t), \qquad \hat{\mathbf{z}}_{1|t} = \mathbf{z}_t + (1-t)\mathbf{v}_t(\mathbf{z}_t). \tag{5}$$

The reverse update can be written as

$$\mathbf{z}_{t-\Delta t} = \big(1 - (t - \Delta t)\big)\hat{\mathbf{z}}_{0|t} + (t - \Delta t)\hat{\mathbf{z}}_{1|t}. \tag{6}$$

Measurement feedback as likelihood. Under Gaussian measurement noise, the feedback in Eq.(2) is exactly the log-likelihood up to a constant:

$$\log p(\mathbf{y} \mid \mathbf{x}) = -\frac{1}{2\sigma_n^2}\|\mathbf{y} - \mathcal{A}(\mathbf{x})\|^2 + C. \tag{7}$$

Thus, posterior inference combines a pretrained data prior with world-derived measurement feedback:

$$p(\mathbf{x} \mid \mathbf{y}) \propto p_{\text{data}}(\mathbf{x}) \exp(R_{\mathbf{y}}(\mathbf{x})). \tag{8}$$

FlowLPS uses this feedback at inference time; it does not require human preference labels, task-specific model training, or policy optimization.

Langevin–correction viewpoint. Many posterior samplers combine stochastic exploration with deterministic correction steps. A useful abstraction is

$$\tilde{\mathbf{z}}^{k+1} = \mathbf{z}^k - \gamma\nabla f(\mathbf{z}^k) + \sqrt{2\gamma}\,\epsilon^k, \qquad \epsilon^k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{9}$$

$$\mathbf{z}^{k+1} = \mathcal{C}_\gamma(\tilde{\mathbf{z}}^{k+1}). \tag{10}$$

Different choices of $\mathcal{C}_\gamma$ recover different methods: $\mathcal{C}_\gamma = \mathrm{Id}$ gives ULA, $\mathcal{C}_\gamma = \Pi_{\mathcal{C}}$ gives projected Langevin algorithms [26, 27], and $\mathcal{C}_\gamma = \mathrm{prox}_{\gamma g}$ gives proximal-Langevin-type updates [28–30].

FlowLPS adopts this operational principle for finite-budget feedback use: Langevin updates produce posterior-oriented candidate initializations, while proximal refinement locally improves the measurement feedback from those candidates. Unlike classical proximal Langevin methods, our goal is not to prove asymptotic posterior convergence, but to obtain a practical exploration–refinement balance for latent generative inference under a fixed compute budget.

4 FlowLPS: Feedback-Guided Langevin-Proximal Sampling

We present FlowLPS as a practical finite-step latent flow solver that aligns pretrained generative samples with world-derived measurement feedback. Motivated by the Langevin+correction viewpoint, FlowLPS adapts an exploration–refinement structure to latent flow inference.

At each reverse-time step, FlowLPS performs three operations. First, a few Langevin updates produce stochastic, posterior-oriented candidate initializations using the measurement feedback and the local generative prior. Second, a local MAP-style proximal refinement step improves the measurement-feedback value from this initialization. Third, a pCN-style re-noising step refreshes the noise component while retaining trajectory coherence. Together, these components provide a practical balance between feedback-aware exploration and local feedback exploitation under a fixed compute budget.

4.1 Phase 1: Posterior-Oriented Langevin Exploration

At timestep t, we adopt a local Gaussian approximation of the conditional clean distribution, following prior work [7, 11, 12]:

$$p(\mathbf{z}_0|\mathbf{z}_t) \approx \mathcal{N}(\hat{\mathbf{z}}_{0|t}, s_t^2\mathbf{I}), \tag{11}$$

where s_t^2 is specified heuristically. Under this approximation, the likelihood gradient is the measurement-feedback gradient and the resulting conditional posterior gradient can be evaluated as

$$\nabla_{\mathbf{z}_0} \log p(\mathbf{z}_0|\mathbf{z}_t, \mathbf{y}) = \nabla_{\mathbf{z}_0} \log p(\mathbf{y}|\mathbf{z}_0) + \nabla_{\mathbf{z}_0} \log p(\mathbf{z}_0|\mathbf{z}_t). \tag{12}$$

Starting from the clean image estimate $\hat{\mathbf{z}}_{0|t}^{(0)} = \hat{\mathbf{z}}_{0|t}$, we run N_L Langevin updates:

$$\hat{\mathbf{z}}_{0|t}^{(j+1)} = \hat{\mathbf{z}}_{0|t}^{(j)} + \eta_t \nabla_{\hat{\mathbf{z}}_{0|t}^{(j)}} \log p(\hat{\mathbf{z}}_{0|t}^{(j)}|\mathbf{z}_t, \mathbf{y}) + \sqrt{2\eta_t}\,\boldsymbol{\epsilon}_j, \quad \boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{13}$$

for $j = 0, \cdots, N_L - 1$. This step injects stochastic posterior-oriented exploration around the model-induced trajectory, producing diverse, posterior-informed initializations for the subsequent deterministic refinement.

4.2 Phase 2: Local MAP-Style Proximal Feedback Refinement

Finite-step Langevin updates provide posterior-oriented exploration, but may not yield a measurement-consistent estimate within a limited number of steps. We therefore refine the Langevin-updated point via a local MAP problem.

Proposition 1. *Assume $p(\mathbf{z}_0|\mathbf{z}_t) = \mathcal{N}(\hat{\mathbf{z}}_{0|t}, s_t^2 I)$ and Gaussian measurement noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$. Then, the maximizer is given by*

$$\mathbf{z}_{0|t}^* = \arg\max_{\mathbf{z}_0} \log p(\mathbf{z}_0|\mathbf{z}_t, \mathbf{y}) = \arg\min_{\mathbf{z}_0} \|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}_0))\|^2 + \frac{\sigma_n^2}{s_t^2}\|\mathbf{z}_0 - \hat{\mathbf{z}}_{0|t}\|^2. \tag{14}$$

Proposition 1 shows that maximizing the local feedback-regularized posterior is equivalent to a proximal measurement-feedback refinement objective. The first term corresponds to maximizing the world-derived measurement feedback, while the second term keeps the refinement local to the posterior-oriented Langevin candidate. This stochastic initialization can change the attraction region from which refinement is performed, while the proximal objective improves measurement consistency locally.

FlowLPS therefore interpolates between two regimes: removing the proximal step ($N_P = 0$) yields a sampling-dominant solver similar to DAPS [11], while removing the Langevin step ($N_L = 0$) yields a refinement-dominant solver similar to Flower [21].

4.3 Local Finite-Step Feedback-Use Analysis

We provide a local, interpretive analysis of the exploration–refinement mechanism. This analysis focuses on within-basin concentration after the Langevin phase provides a useful local initialization, rather than on global basin selection. Specifically, we consider a neighborhood where the negative log conditional posterior is well approximated by a smooth strongly convex objective. Under this local basin model, deterministic refinement contracts toward the local minimizer, whereas continued Langevin updates retain an additive stochastic variance term. This explains why proximal refinement can concentrate faster than continuing stochastic Langevin updates once a useful initialization is obtained, and motivates the N_L ablations in Sec. 5.3.1.

Define $\Phi(\mathbf{z}_0) := -\log p(\mathbf{z}_0|\mathbf{z}_t, \mathbf{y})$. Under the local Gaussian approximation in Sec. 4.1, Φ is equivalent to the proximal objective in Eq.(14) up to additive constants and a positive scaling factor. Thus, both deterministic refinement and ULA can be analyzed locally through the same smooth objective Φ around a local minimizer $\mathbf{z}_B^\star$.

Proposition 2. *Let Φ be twice differentiable on a convex neighborhood Ω_B of a local minimizer $\mathbf{z}_B^\star$, and assume*

$$\mu\mathbf{I} \preceq \nabla^2\Phi(\mathbf{z}) \preceq L\mathbf{I}, \qquad \forall \mathbf{z} \in \Omega_B, \tag{15}$$

for some $0 < \mu \leq L$. Let R_N denote N gradient-refinement steps with step size $0 < \tau \leq \frac{1}{L}$, and assume the iterates remain in Ω_B. Then, with $q = 1 - \tau\mu < 1$,

$$\|R_N(\mathbf{z}) - \mathbf{z}_B^\star\| \leq q^N \|\mathbf{z} - \mathbf{z}_B^\star\|. \tag{16}$$

Consequently, for any $B_r = \{\mathbf{z} : \|\mathbf{z} - \mathbf{z}_B^\star\| \leq r\}$ with $B_{r/q^N} \subset \Omega_B$, we have

$$B_{r/q^N} \subseteq R_N^{-1}(B_r). \tag{17}$$

Thus, refinement enlarges the effective pre-refinement region from which a small neighborhood of the local minimizer can be reached.

Lemma 1. *Under the assumptions of Proposition 2, consider the ULA update*

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \tau\nabla\Phi(\mathbf{z}_k) + \sqrt{2\tau}\boldsymbol{\xi}_k, \quad \boldsymbol{\xi}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \tag{18}$$

and assume the iterates remain in Ω_B. Then

$$\mathbb{E}\|\mathbf{z}_N - \mathbf{z}_B^\star\|^2 \leq q^{2N}\|\mathbf{z}_0 - \mathbf{z}_B^\star\|^2 + 2\tau d\,\frac{1 - q^{2N}}{1 - q^2}, \qquad q = 1 - \tau\mu. \tag{19}$$

Algorithm 1 Algorithm of FlowLPS

Require: $\mathbf{y}$, $\mathcal{A}$, VAE decoder $\mathcal{D}$, $\mathbf{v}_t^\theta$, N_L
Initialize: $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t : 1 \to 0$ **do**
  $\mathbf{v}_t \leftarrow \mathbf{v}_t^\theta(\mathbf{z}_t)$
  $\hat{\mathbf{z}}_{0|t}^{(0)} \leftarrow \mathbf{z}_t - t\mathbf{v}_t,\ \hat{\mathbf{z}}_{1|t} \leftarrow \mathbf{z}_t + (1-t)\mathbf{v}_t$
  $\hat{\boldsymbol{\epsilon}}_{1|t} \leftarrow \sqrt{1-(t-\Delta t)}\hat{\mathbf{z}}_{1|t} + \sqrt{t-\Delta t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ pCN update
  for $j = 0, 1, \cdots, N_L - 1$ **do**
    $\hat{\mathbf{z}}_{0|t}^{(j+1)} \leftarrow \hat{\mathbf{z}}_{0|t}^{(j)} + \eta_t \nabla_{\hat{\mathbf{z}}_{0|t}^{(j)}} \log p(\hat{\mathbf{z}}_{0|t}^{(j)} | \mathbf{z}_t, \mathbf{y}) + \sqrt{2\eta_t} Z_j, \quad Z_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Langevin dynamics
  end for
  $\mathbf{z} \leftarrow \hat{\mathbf{z}}_{0|t}^{(N_L)}$
  $\mathbf{z}_{0|t}^* \leftarrow \underset{\mathbf{z}}{\text{argmin}}\ ||\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}))||^2 + \frac{\sigma_n^2}{s_t^2}||\mathbf{z} - \hat{\mathbf{z}}_{0|t}^{(N_L)}||^2$ ▷ Proximal optimization
  $\mathbf{z}_{t-\Delta t} \leftarrow \big(1-(t-\Delta t)\big)\mathbf{z}_{0|t}^* + (t-\Delta t)\hat{\boldsymbol{\epsilon}}_{1|t}$
end for
return $\mathcal{D}(\mathbf{z}_{0|t}^*)$

Together, Proposition 2 and Lemma 1 clarify the complementary local behavior of deterministic refinement and Langevin exploration under this local basin model. Deterministic refinement contracts geometrically toward the local minimizer and enlarges the effective pre-refinement region by a factor q^{-N}. In contrast, ULA has the same contraction in its drift, but the injected Gaussian noise introduces an additive variance term in the finite-step bound. Thus, Langevin updates maintain stochastic exploration but can remain diffuse over a limited number of steps, whereas proximal refinement rapidly concentrates once initialized in a useful local region.

This local view suggests a qualitative update-allocation trade-off controlled by N_L. Too few Langevin updates make refinement strongly tied to the model-predicted initialization, while too many Langevin updates leave insufficient update budget for local measurement-consistent refinement. We evaluate this trade-off through the N_L ablations in Sec. 5.3.1 and Appendix C.1.

4.4 Trajectory Stabilization via pCN-Style Re-Noising

After obtaining the refined clean estimate $\mathbf{z}_{0|t}^*$, a standard way to move to the next noise level is

$$\mathbf{z}_{t-\Delta t} = (1-(t-\Delta t))\mathbf{z}_{0|t}^* + (t-\Delta t)\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{20}$$

Rather than drawing this noise component at each step, FlowLPS uses a pCN-style refresh and substitutes the refreshed noise into the re-noising equation. Given $\hat{\mathbf{z}}_{1|t}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the pCN update is

$$\hat{\mathbf{z}}_{1|t}^{(i+1)} = \rho_t \hat{\mathbf{z}}_{1|t}^{(i)} + \sqrt{1-\rho_t^2}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{21}$$

This form is inspired by the Gaussian-preserving property of pCN [25]. In practice, FlowLPS initializes the refresh from the model-predicted noise $\hat{\mathbf{z}}_{1|t}^{(0)} = \mathbf{z}_t + (1-t)\mathbf{v}_t(\mathbf{z}_t)$, trading exact marginal preservation for trajectory coherence. We use a single pCN-style re-noising step with mixing coefficient $\rho_t = \sqrt{1-t}$, which adaptively balances fresh stochasticity with preservation of the model-predicted noise direction. Ablations on the mixing schedule and number of pCN steps are provided in Sec. 5.3.2 and Appendix C.3.

Diversity. Although proximal refinement locally concentrates each trajectory, FlowLPS remains stochastic through both Langevin updates and pCN-style re-noising. We evaluate the resulting diversity–validity trade-off in Appendix D.2.

The complete FlowLPS algorithm is summarized in Algorithm 1.

5 Experimental Results

5.1 Experimental Setup

Datasets, tasks, and metrics. We evaluate FlowLPS on 1k images from FFHQ [31] and 0.8k images from DIV2K [32], resized to 768×768 resolution. We consider five measurement-feedback

Table 1: **Quantitative Results on FFHQ 1k and DIV2K 0.8k datasets** with $\sigma_n = 0.03$. **Bold**: best. Underline: second best.

Task	Method	FFHQ 1k				DIV2K 0.8k			
		PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)
Inpainting (Box)	DAPS	21.78	0.842	24.75	0.114	23.72	0.841	19.96	0.080
	RLSD	16.32	0.519	191.6	0.538	16.25	0.417	164.1	0.535
	FlowDPS	18.81	0.771	58.49	0.204	21.36	0.756	36.13	0.142
	FlowChef	17.72	0.701	74.07	0.265	21.02	0.688	59.44	0.160
	FLAIR	**21.84**	0.836	22.63	0.122	23.90	0.845	20.76	0.076
	Flower	21.63	**0.876**	48.91	0.119	**24.41**	**0.868**	22.38	0.081
	Ours	21.68	0.849	**21.90**	**0.107**	23.73	0.862	**15.50**	**0.065**
Inpainting (Random)	DAPS	33.41	0.857	13.28	0.034	27.97	0.799	15.37	0.039
	RLSD	17.04	0.539	123.6	0.441	15.63	0.392	137.7	0.480
	FlowDPS	30.36	0.808	32.03	0.109	25.44	0.713	27.96	0.114
	FlowChef	32.07	0.849	63.23	0.077	25.35	0.688	57.82	0.139
	FLAIR	14.09	0.502	131.9	0.443	13.78	0.355	116.9	0.408
	Flower	31.37	0.826	15.76	0.048	28.81	**0.843**	9.824	0.026
	Ours	**34.34**	**0.886**	**7.460**	**0.019**	**28.85**	0.841	**9.360**	**0.025**
Gaussian Deblurring	DAPS	27.21	0.715	41.72	0.171	21.42	0.498	81.81	0.294
	RLSD	21.42	0.626	110.3	0.426	17.90	0.424	154.9	0.571
	FlowDPS	26.87	0.715	45.19	0.201	21.17	0.519	78.79	0.322
	FlowChef	24.87	0.690	46.61	0.260	21.19	0.542	102.6	0.363
	FLAIR	26.84	0.704	**29.03**	0.165	21.27	0.509	63.97	0.276
	Flower	**28.67**	**0.796**	94.34	0.167	21.84	**0.561**	88.65	0.312
	Ours	27.76	0.737	31.22	**0.130**	**21.97**	0.549	**63.62**	**0.242**
Motion Deblurring	DAPS	28.96	0.754	31.71	0.116	23.13	0.572	55.48	0.187
	RLSD	21.67	0.631	99.00	0.396	18.17	0.431	136.3	0.538
	FlowDPS	25.98	0.704	48.88	0.229	20.91	0.514	81.88	0.325
	FlowChef	20.52	0.626	69.13	0.388	19.84	0.528	128.0	0.431
	FLAIR	28.50	0.744	22.62	0.113	22.95	0.579	43.70	0.174
	Flower	**31.00**	**0.834**	67.56	0.087	23.51	0.624	68.60	0.207
	Ours	29.81	0.774	**21.15**	**0.067**	**24.29**	**0.642**	**36.02**	**0.110**
Super-Resolution ($\times 12$)	DAPS	24.54	0.535	81.93	0.309	19.13	0.347	106.9	0.379
	RLSD	19.53	0.664	125.0	0.424	17.04	0.453	172.0	0.553
	FlowDPS	26.68	0.725	36.07	0.171	21.77	0.543	67.01	0.272
	FlowChef	26.59	0.721	58.39	0.208	21.86	0.563	96.32	0.305
	FLAIR	26.82	0.739	**31.33**	0.181	20.42	0.475	67.91	0.285
	Flower	**28.10**	**0.769**	49.55	0.179	**22.16**	**0.567**	75.64	0.280
	Ours	27.12	0.716	34.54	**0.158**	21.16	0.505	**65.52**	**0.267**

settings: Gaussian deblurring, motion deblurring, $12\times$ super-resolution, box inpainting, and random inpainting. All tasks use Gaussian measurement noise with $\sigma_n = 0.03$. For FFHQ, we use the text prompt "a high quality photo of a face," and for DIV2K, we use prompts generated by DAPE [33].

We report PSNR and SSIM as reconstruction-fidelity metrics that reflect how well the output satisfies the measurement-induced constraint, and FID and LPIPS as perceptual-realism metrics. This dual evaluation is important because inverse problems are ambiguous: aggressively optimizing the measurement feedback can improve fidelity while degrading perceptual realism.

Baselines. We compare FlowLPS with diffusion-based solvers, RLSD [17] and DAPS [11], and flow-based solvers, FlowChef [13], FlowDPS [14], FLAIR [19], and Flower [21]. To isolate the effect of the feedback-use strategy, we adapt all baselines to the same SD3 backbone used by FlowLPS. Unless otherwise stated, all methods are evaluated under a fixed budget of 40 NFEs. For methods with repeated measurement updates, we allow up to 15 inner updates per timestep for a matched optimization budget. For DAPS, we use 20 outer steps with 2 inner steps to match the 40-NFE budget.

5.2 Main Results: Measurement Feedback and Perceptual Realism

Table 1 summarizes the quantitative comparison across five inverse problems. Overall, FlowLPS achieves a strong balance between measurement-feedback satisfaction and perceptual realism. This balance is central to world-feedback-guided generative inference: the feedback signal $\|\mathbf{y} - \mathcal{A}(\mathbf{x})\|^2$ should be satisfied, but the solution should also remain on the pretrained data prior.

Refinement-dominant solvers, such as Flower, often achieve competitive distortion metrics by aggressively optimizing the measurement residual. However, consistent with the perception–distortion trade-off [34, 35], this can come at the cost of perceptual fidelity, producing overly smooth or less natural textures in ambiguous regions. Conversely, sampling-dominant solvers such as DAPS preserve stochastic posterior exploration, but under matched inference budgets they often fail to concentrate into sharp, feedback-satisfying reconstructions.

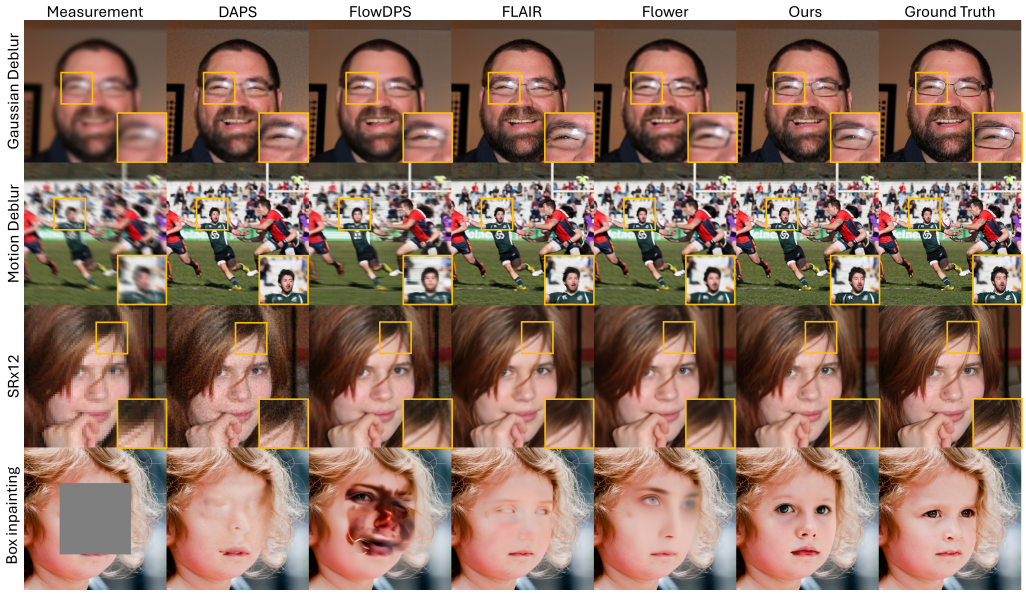

Figure 3: **Qualitative comparison on FFHQ and DIV2K datasets.**

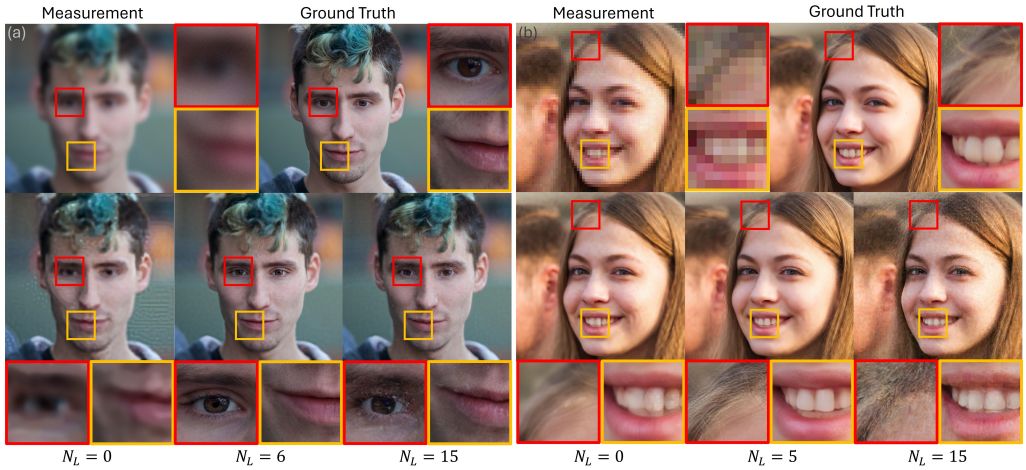

Figure 4: **Qualitative effect of Langevin steps (N_L).** (a) Gaussian deblurring and (b) $\times 12$ super-resolution. With no Langevin updates ($N_L = 0$), FlowLPS behaves as a refinement-dominant solver and tends to produce overly smooth reconstructions. With excessive Langevin updates ($N_L = 15$), too much budget is spent on stochastic exploration, leaving insufficient refinement and causing unstable details. Moderate values ($N_L = 5$–7) provide the best exploration–refinement balance.

FlowLPS targets the intermediate regime. The Langevin phase produces posterior-oriented candidate initializations, while the proximal phase locally improves the measurement-feedback value from those candidates. Across tasks, FlowLPS maintains competitive PSNR/SSIM while improving FID/LPIPS in many settings, indicating a better feedback-use trade-off. The qualitative comparisons in Fig. 3 show the same trend: FlowLPS preserves sharper and more natural local details while maintaining measurement consistency. Additional qualitative results are provided in Appendix D.3. We further evaluate FlowLPS beyond the main latent-space linear setting, including pixel-space inverse problems and phase retrieval, in Appendix B.4.

5.3 Feedback-Use Ablations

5.3.1 Update Allocation: Langevin Exploration vs. Proximal Refinement

We vary the number of Langevin steps N_L on 200 FFHQ images while keeping the total number of Langevin–proximal update steps fixed at $N_T = 15$. Increasing N_L shifts computation from local

proximal refinement to posterior-oriented Langevin exploration. This tests how a finite compute budget should be allocated between feedback-aware exploration and local feedback exploitation.

Figure 5 shows a clear intermediate optimum. When $N_L = 0$, FlowLPS behaves as a refinement-dominant solver initialized from the model prediction, often producing overly smooth reconstructions. Moderate Langevin steps ($N_L = 5$–7) improve perceptual metrics while maintaining competitive fidelity, suggesting that posterior-oriented stochastic initialization provides a better starting point for proximal feedback refinement. When N_L is too large, the method shifts toward a sampling-dominant regime, leaving insufficient update budget to concentrate the estimate and satisfy the measurement feedback.

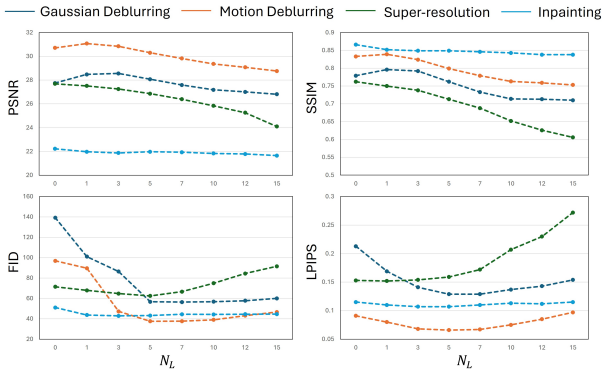

Figure 5: **Quantitative impact of Langevin dynamics steps (N_L).**

These results support the intended exploration–refinement mechanism of FlowLPS: Langevin updates improve the candidate initialization, while the proximal phase concentrates the estimate into a feedback-satisfying reconstruction under the fixed update budget. Figure 4 visualizes the same behavior, and additional results are provided in Appendix C.1.

5.3.2 Trajectory Stabilization via pCN Re-Noising

We ablate two design choices in the pCN-style re-noising step on 200 FFHQ images: the mixing schedule ρ_t and the noise initialization. Although pCN re-noising is not the primary mechanism for optimizing measurement feedback, it stabilizes the reverse trajectory after feedback refinement and preserves useful stochasticity. Table 2 evaluates different schedules on box inpainting. Adaptive schedules outperform fixed ones, with $\rho_t = \sqrt{1-t}$ achieving the best FID and LPIPS. This suggests that stochastic refresh should depend on the remaining reverse-time uncertainty rather than being fixed across timesteps. Using this schedule, Table 3 shows that initializing the refresh from the model-predicted noise improves perceptual metrics over independent Gaussian initialization. This supports our design choice of retaining trajectory coherence while injecting controlled stochasticity. Additional ablations on the number of pCN steps, decaying-N_L schedules, and component-wise effects are provided in Appendix C.3, Appendix C.2, and Appendix C.4 respectively.

Table 2: **pCN mixing schedule ablation.**

ρ_t	PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)
0	**22.29**	0.843	54.21	0.116
0.5	21.47	0.844	53.79	0.111
$1-t$	21.79	0.842	43.56	0.108
$\sqrt{1-t}$	21.89	**0.854**	**37.40**	**0.101**

Table 3: **Noise initialization ablation.**

Task	Initialization	PSNR (↑)	FID (↓)	LPIPS (↓)
Super-Resolution (×12)	$\mathcal{N}(\mathbf{0}, \mathbf{I})$	**27.21**	73.43	0.181
	Model prediction	27.13	**63.08**	**0.155**
Inpainting (Box)	$\mathcal{N}(\mathbf{0}, \mathbf{I})$	**22.42**	48.69	0.106
	Model prediction	21.89	**37.40**	**0.101**

6 Conclusion

FlowLPS illustrates a simple form of world-feedback-guided generative inference. The feedback signal is not a human preference score, but a physical consistency value induced by a known measurement process. Unlike RLHF-style pipelines, FlowLPS does not train a policy or reward model; instead, it aligns pretrained generative samples at inference time using explicit measurement feedback. This makes inverse problems a useful testbed for studying how foundation generative models can leverage objective, non-human feedback signals under finite compute budgets.

Limitations. The current formulation assumes access to a known measurement operator and focuses on feedback available at inference time. More general world-feedback settings may involve delayed, noisy, partial, or learned feedback signals. FlowLPS also depends on the pretrained backbone prior, so highly out-of-distribution inputs may produce artifacts or hallucinated details. Extending Langevin–proximal feedback use to broader world-feedback settings, including safety-critical or delayed-feedback environments, is an interesting direction for future work.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[2] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR*, 2021.

[3] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://arxiv.org/abs/2210.02747`.

[4] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://arxiv.org/abs/2209.03003`.

[5] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[6] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=OnD9zGAGT0k`.

[7] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=9_gsMA8MRKQ`.

[8] Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=o3BxOLoxm1`.

[9] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[10] Hyungjin Chung, Suhyeon Lee, and Jong Chul Ye. Decomposed diffusion sampler for accelerating large-scale inverse problems. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=DsEhqQtfAG`.

[11] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. *arXiv preprint arXiv:2407.01521*, 2024.

[12] Ashwini Pokle, Matthew J. Muckley, Ricky T. Q. Chen, and Brian Karrer. Training-free linear image inverses via flows. In *Transactions on Machine Learning Research*, 2024. URL `https://arxiv.org/abs/2310.04432`.

[13] Maitreya Patel, Song Wen, Dimitris N Metaxas, and Yezhou Yang. Steering rectified flow models in the vector field for controlled image generation. *arXiv preprint arXiv:2412.00100*, 2024.

[14] Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. Flowdps: Flow-driven posterior sampling for inverse problems. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025. URL `https://arxiv.org/abs/2503.08136`.

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

[16] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1229, 2023.

[17] Nicolas Zilberstein, Morteza Mardani, and Santiago Segarra. Repulsive latent score distillation for solving inverse problems. In *International Conference on Learning Representations*, 2024. URL `https://arxiv.org/abs/2406.16683`.

[18] Ségolène Tiffany Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. Pnp-flow: Plug-and-play image restoration with flow matching. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://arxiv.org/abs/2410.02423`.

[19] Julius Erbach, Dominik Narnhofer, Andreas Dombos, Bernt Schiele, Jan Eric Lenssen, and Konrad Schindler. Solving inverse problems with flair. In *Advances in Neural Information Processing Systems*, 2025. URL `https://arxiv.org/abs/2506.02680`.

[20] Xiang Li, Soo Min Kwon, Shijun Liang, Ismail R. Alkhouri, Saiprasad Ravishankar, and Qing Qu. Decoupled data consistency with diffusion purification for image restoration, 2025. URL `https://arxiv.org/abs/2403.06054`.

[21] Mehrsa Pourya, Bassam El Rawas, and Michael Unser. Flower: A flow-matching solver for inverse problems. In *International Conference on Learning Representations*, 2025. URL `https://arxiv.org/abs/2509.26287`.

[22] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=j8hdRqOUhN`.

[23] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=nJJjv0JDJju`.

[24] Rayhan Zirvi, Bahareh Tolooshams, and Anima Anandkumar. Diffusion state-guided projected gradient for inverse problems. In *International Conference on Learning Representations*, 2025. URL `https://arxiv.org/abs/2410.03463`.

[25] Alexandros Beskos, Mark Girolami, Shiwei Lan, Patrick E Farrell, and Andrew M Stuart. Geometric MCMC for infinite-dimensional inverse problems. *Journal of Computational Physics*, 335:327–351, 2017.

[26] Sébastien Bubeck, Ronen Eldan, and Joseph Lehec. Sampling from a log-concave distribution with projected langevin monte carlo. *Discrete & Computational Geometry*, 59(4):757–783, 2018.

[27] Andrew Lamperski. Projected stochastic gradient langevin algorithms for constrained sampling and non-convex learning. In *Conference on Learning Theory*, pages 2891–2937. PMLR, 2021.

[28] Adil Salim and Peter Richtarik. Primal dual interpretation of the proximal stochastic gradient langevin algorithm. *Advances in Neural Information Processing Systems*, 33:3786–3796, 2020.

[29] Matthias J Ehrhardt, Lorenz Kuger, and Carola-Bibiane Schönlieb. Proximal langevin sampling with inexact proximal mapping. *SIAM journal on imaging sciences*, 17(3):1729–1760, 2024.

[30] Marien Renaud, Valentin De Bortoli, Arthur Leclaire, and Nicolas Papadakis. From stability of langevin diffusion to convergence of proximal mcmc for non-log-concave sampling. *Advances in Neural Information Processing Systems*, 2025.

[31] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[32] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017.

[33] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25456–25467, 2024.

[34] Christian Ledig, Lucas Theis, Ferenc Husz'ar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[35] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[36] Suraj Patil, Pedro Cuenca, Nathan Lambert, and Patrick von Platen. Stable diffusion with diffusers. *Hugging Face Blog*, 2022. https://huggingface.co/blog/stable_diffusion.

A Proofs

Proposition 1. *Assume $p(\mathbf{z}_0|\mathbf{z}_t) = \mathcal{N}(\hat{\mathbf{z}}_{0|t}, s_t^2 I)$ and Gaussian measurement noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$. Then, the maximizer is given by*

$$\mathbf{z}_{0|t}^* = \arg\max_{\mathbf{z}_0} \log p(\mathbf{z}_0|\mathbf{z}_t, \mathbf{y}) = \arg\min_{\mathbf{z}_0} \|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}_0))\|^2 + \frac{\sigma_n^2}{s_t^2}\|\mathbf{z}_0 - \hat{\mathbf{z}}_{0|t}\|^2. \tag{14}$$

Proof. By Bayes' rule and the Markov property,

$$\begin{aligned} p(\mathbf{y}|\mathbf{z}_0, \mathbf{z}_t) &= p(\mathbf{y}|\mathbf{z}_0), \\ p(\mathbf{z}_0|\mathbf{z}_t, \mathbf{y}) &= \frac{p(\mathbf{y}|\mathbf{z}_0, \mathbf{z}_t)p(\mathbf{z}_0|\mathbf{z}_t)}{p(\mathbf{y}|\mathbf{z}_t)} \\ &\propto p(\mathbf{y}|\mathbf{z}_0)p(\mathbf{z}_0|\mathbf{z}_t). \end{aligned}$$

Substituting the Gaussian forms gives

$$\begin{aligned} \max_{\mathbf{z}_0} \log p(\mathbf{z}_0|\mathbf{z}_t, \mathbf{y}) &= \max_{\mathbf{z}_0} \log p(\mathbf{y}|\mathbf{z}_0) + \log p(\mathbf{z}_0|\mathbf{z}_t) \\ &= \max_{\mathbf{z}_0} -\frac{\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}_0))\|^2}{2\sigma_n^2} - \frac{\|\mathbf{z}_0 - \hat{\mathbf{z}}_{0|t}\|^2}{2s_t^2} \\ &= \min_{\mathbf{z}_0} \|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}_0))\|^2 + \frac{\sigma_n^2}{s_t^2}\|\mathbf{z}_0 - \hat{\mathbf{z}}_{0|t}\|^2 \end{aligned}$$

□

Proposition 2. *Let Φ be twice differentiable on a convex neighborhood Ω_B of a local minimizer $\mathbf{z}_B^\star$, and assume*

$$\mu\mathbf{I} \preceq \nabla^2\Phi(\mathbf{z}) \preceq L\mathbf{I}, \qquad \forall \mathbf{z} \in \Omega_B, \tag{15}$$

for some $0 < \mu \leq L$. Let R_N denote N gradient-refinement steps with step size $0 < \tau \leq \frac{1}{L}$, and assume the iterates remain in Ω_B. Then, with $q = 1 - \tau\mu < 1$,

$$\|R_N(\mathbf{z}) - \mathbf{z}_B^\star\| \leq q^N \|\mathbf{z} - \mathbf{z}_B^\star\|. \tag{16}$$

Proof. Let $T(\mathbf{z}) = \mathbf{z} - \tau\nabla\Phi(\mathbf{z})$. Since Φ is twice differentiable and satisfies

$$\mu\mathbf{I} \preceq \nabla^2\Phi(\mathbf{z}) \preceq L\mathbf{I} \tag{22}$$

on Ω_B, the gradient map

$$T(\mathbf{z}) = \mathbf{z} - \tau\nabla\Phi(\mathbf{z}) \tag{23}$$

is contractive on Ω_B for any $0 < \tau \leq 1/L$. Indeed, for any $\mathbf{z} \in \Omega_B$, using $\nabla\Phi(\mathbf{z}_B^\star) = 0$, we have

$$T(\mathbf{z}) - \mathbf{z}_B^\star = \mathbf{z} - \mathbf{z}_B^\star - \tau\left(\nabla\Phi(\mathbf{z}) - \nabla\Phi(\mathbf{z}_B^\star)\right) \tag{24}$$

$$= \left[\mathbf{I} - \tau\int_0^1 \nabla^2\Phi\big(\mathbf{z}_B^\star + s(\mathbf{z} - \mathbf{z}_B^\star)\big)\,ds\right](\mathbf{z} - \mathbf{z}_B^\star). \tag{25}$$

The eigenvalues of the averaged Hessian lie in $[\mu, L]$. Therefore, since $0 < \tau \leq 1/L$, the operator norm of the bracketed matrix is at most

$$q = 1 - \tau\mu. \tag{26}$$

Thus,

$$\|T(\mathbf{z}) - \mathbf{z}_B^\star\| \leq q\|\mathbf{z} - \mathbf{z}_B^\star\|. \tag{27}$$

Applying this contraction recursively for N refinement steps gives

$$\|R_N(\mathbf{z}) - \mathbf{z}_B^\star\| \leq q^N \|\mathbf{z} - \mathbf{z}_B^\star\|. \tag{28}$$

This proves the claim. □

Lemma 1. *Under the assumptions of Proposition 2, consider the ULA update*

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \tau\nabla\Phi(\mathbf{z}_k) + \sqrt{2\tau}\boldsymbol{\xi}_k, \quad \boldsymbol{\xi}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \tag{18}$$

and assume the iterates remain in Ω_B. Then

$$\mathbb{E}\|\mathbf{z}_N - \mathbf{z}_B^\star\|^2 \le q^{2N}\|\mathbf{z}_0 - \mathbf{z}_B^\star\|^2 + 2\tau d\,\frac{1-q^{2N}}{1-q^2}, \qquad q = 1-\tau\mu. \tag{19}$$

Proof. Let

$$T(\mathbf{z}) = \mathbf{z} - \tau\nabla\Phi(\mathbf{z}). \tag{29}$$

From Proposition 2, T is locally contractive:

$$\|T(\mathbf{z}) - \mathbf{z}_B^\star\| \le q\|\mathbf{z} - \mathbf{z}_B^\star\|. \tag{30}$$

Conditioned on $\mathbf{z}_k$, the ULA update satisfies

$$\mathbf{z}_{k+1} - \mathbf{z}_B^\star = T(\mathbf{z}_k) - \mathbf{z}_B^\star + \sqrt{2\tau}\boldsymbol{\xi}_k. \tag{31}$$

Taking conditional expectation and using $\mathbb{E}[\boldsymbol{\xi}_k] = 0$ and $\mathbb{E}\|\boldsymbol{\xi}_k\|^2 = d$, we obtain

$$\mathbb{E}\left[\|\mathbf{z}_{k+1} - \mathbf{z}_B^\star\|^2 \mid \mathbf{z}_k\right] = \|T(\mathbf{z}_k) - \mathbf{z}_B^\star\|^2 + 2\tau d \tag{32}$$

$$\le q^2\|\mathbf{z}_k - \mathbf{z}_B^\star\|^2 + 2\tau d. \tag{33}$$

Taking total expectation gives the recursion

$$a_{k+1} \le q^2 a_k + 2\tau d, \qquad a_k = \mathbb{E}\|\mathbf{z}_k - \mathbf{z}_B^\star\|^2. \tag{34}$$

Unrolling the recursion yields

$$a_N \le q^{2N}a_0 + 2\tau d\sum_{\ell=0}^{N-1} q^{2\ell} = q^{2N}a_0 + 2\tau d\frac{1-q^{2N}}{1-q^2}. \tag{35}$$

This proves the result. □

B Implementation Details

In this section, we provide implementation details for FlowLPS and all baseline methods. Unless otherwise stated, the main experiments use Stable Diffusion 3.5-Medium [5] as the matched backbone for all baselines and FlowLPS. All experiments were run on a single NVIDIA RTX 3090 GPU. We additionally report original SD2.1 diffusion-backbone [36] results for DAPS and RLSD in Sec. B.3.

B.1 FlowLPS Configuration

We employed a fixed number of Langevin-Proximal update steps, $N_T = 15$ for our main experiments. The variance parameter $s_t^2 = t$ was used in the proximal objective. A constant step size $\zeta_t = 10^{-4}$ was used for Langevin dynamics across all timesteps. For the proximal optimization, we utilized the SGD optimizer. We used an initial learning rate of 0.5 for super-resolution and 0.1 for other tasks. To ensure stability during optimization, we applied a step-based learning rate decay schedule: for super-resolution, the learning rate is decayed by a factor of 0.85 every 5 iterations, and for inpainting, by 0.65 every 10 iterations. For the forward measurement operators, we adopted the implementations from DPS [6]. Based on empirical findings that negative prompts improve results, we used the negative prompt "blurry, noisy, a bad quality photo" for classifier-free guidance. All experiments were conducted with a Classifier-Free Guidance (CFG) scale of 2.0.

Time Schedule. Fig. 6 illustrates the Mean Squared Error (MSE) loss of the reconstructed clean image estimate $\mathcal{D}(\mathbf{z}_{0|t}^*)$ as a function of timestep t, scaled relative to the maximum value. The results are averaged over 100 FFHQ images, for both super-resolution and deblurring tasks, with the deblurring curve representing the average of both Gaussian and motion deblurring tasks. We observed that the MSE decreases as t decreases, reaching a minimum around $t \approx 0.2$ for deblurring and $t \approx 0.3$ for super-resolution, before beginning to rise as $t \to 0$. Based on this trend, we adopted a truncated time schedule to stop the reverse process before significant divergence occurs. Specifically, we set the sampling schedule start point to $40 + \alpha$ (using the first 40 steps). We set $\alpha = 5$ for super-resolution and $\alpha = 3$ for other tasks, which align with the optimal stopping points observed in our analysis. We found these settings empirically provide the best trade-off between reconstruction fidelity (MSE) and perceptual details.

A detailed summary of hyperparameters for all tasks is provided in Table 4.

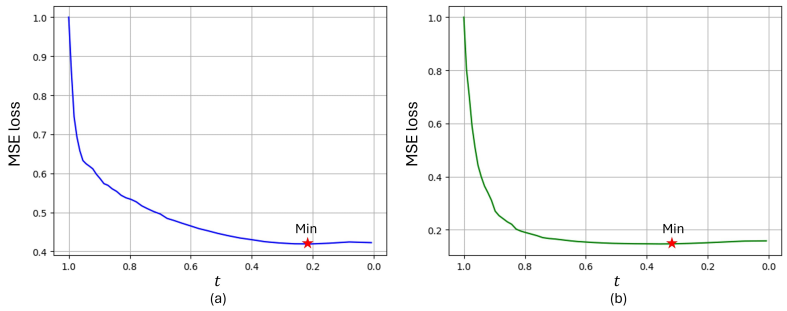

Figure 6: **Reconstruction MSE error vs. Timestep t.** (a) Deblurring, (b) Super-resolution. The plot shows the relative MSE of the estimated clean image $\mathcal{D}(\mathbf{z}^*_{0|t})$ for deblurring (averaged over Gaussian/Motion) and super-resolution. The error minimizes around $t \approx 0.2$ for deblurring and $t \approx 0.3$ for super-resolution, justifying the use of task-specific truncated time schedules.

Table 4: **Hyperparameters for FlowLPS experiments.** N_L denotes the number of Langevin steps, N_P denotes the number of Proximal optimization steps, N_T denotes the total number of Langevin-Proximal optimization steps, ζ_t denotes the step size of Langevin dynamics, and α denotes the sampling schedule offset parameter.

	Inpainting (Box)	Inpainting (Random)	Gaussian Deblurring	Motion Deblurring	Super-resolution ($\times 12$)
ζ_t	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
α	3	3	3	3	5
N_L	4	5	6	6	4
N_P	11	10	9	9	11
N_T	15	15	15	15	15

B.2 Baseline Details

DAPS [11]. We adapt DAPS to the SD3 backbone and use 20 annealing steps with 2 inner denoised-prediction ODE evaluations, resulting in 40 NFEs in total. We use a CFG scale of 2.0 for all tasks. For most tasks, we use 15 MCMC updates per annealing step with a learning rate of 10^{-4}. For super-resolution, we found that 15 MCMC updates produced degenerate reconstructions under the SD3 adaptation; therefore, we use 60 MCMC updates with a learning rate of 10^{-3} and decay the learning rate by a factor of 0.7 every 5 MCMC iterations. All hyperparameters were selected by grid search on a validation set of 200 images.

RLSD [17]. We adapt RLSD to the SD3 backbone and use 40 NFEs. We do not use pixel-space augmentation, as we found that it degraded performance, consistent with observations in prior latent-space inverse solvers [19]. We also disable the repulsive term, since it reduced reconstruction fidelity in our experiments, consistent with the original findings. We set $(lr_z, w_t) = (0.1, 1.5)$ for deblurring tasks, $(0.15, 1.5)$ for super-resolution, and $(0.2, 1.2t)$ for inpainting tasks.

FlowDPS [14]. We use 40 NFEs with 15 optimization steps per iteration and a CFG scale of 2.0. The step size is set to 8.0 for super-resolution, Gaussian deblurring, and motion deblurring, and 12.0 for inpainting tasks.

FlowChef [13]. We use 40 NFEs with a single optimization step per iteration. Although we attempted to increase the number of optimization steps to 15 for a matched inner-update budget, this did not improve performance. We therefore use the original single-step update with a step size of 1.5 across all tasks.

FLAIR [19]. We configure FLAIR with 40 NFEs and 15 optimization steps per iteration. We use a CFG scale of 2.0 and follow the default settings, except for super-resolution where the data-term optimizer learning rate is set to 1.0.

Table 5: **Additional comparison of original diffusion backbones and SD3 adaptations.** We report DAPS and RLSD using their original SD2.1 diffusion backbone and our SD3 adaptation, together with FlowLPS.

Task	Method	FFHQ 1k				DIV2K 0.8k			
		PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)
Inpainting (Box)	DAPS (SD2)	19.77	0.846	53.22	0.139	22.50	0.807	30.57	0.091
	DAPS (SD3)	21.78	0.842	24.75	0.114	23.72	0.841	19.96	0.080
	RLSD (SD2)	13.08	0.766	151.5	0.261	17.11	0.822	75.41	0.148
	RLSD (SD3)	16.32	0.519	191.6	0.538	16.25	0.417	164.1	0.535
	Ours	21.68	0.849	**21.90**	**0.107**	23.73	0.862	**15.50**	**0.065**
Inpainting (Random)	DAPS (SD2)	30.99	0.884	16.48	0.035	25.53	0.764	24.17	0.064
	DAPS (SD3)	33.41	0.857	13.28	0.034	27.97	0.799	15.37	0.039
	RLSD (SD2)	17.24	0.383	116.6	0.405	16.98	0.396	107.0	0.382
	RLSD (SD3)	17.04	0.539	123.6	0.441	15.63	0.392	137.7	0.480
	Ours	**34.34**	**0.886**	**7.460**	**0.019**	**28.85**	0.841	**9.360**	**0.025**
Gaussian Deblurring	DAPS (SD2)	28.20	0.772	49.83	0.174	21.94	**0.550**	75.34	0.286
	DAPS (SD3)	27.21	0.715	41.72	0.171	21.42	0.498	81.81	0.294
	RLSD (SD2)	25.50	0.674	88.23	0.227	21.86	0.538	104.1	**0.242**
	RLSD (SD3)	21.42	0.626	110.3	0.426	17.90	0.424	154.9	0.571
	Ours	27.76	0.737	31.22	**0.130**	**21.97**	0.549	**63.62**	**0.242**
Motion Deblurring	DAPS (SD2)	30.85	0.830	43.17	0.095	**24.56**	**0.661**	49.03	0.136
	DAPS (SD3)	28.96	0.754	31.71	0.116	23.13	0.572	55.48	0.187
	RLSD (SD2)	25.73	0.719	92.52	0.269	22.16	0.578	87.06	0.260
	RLSD (SD3)	21.67	0.631	99.00	0.396	18.17	0.431	136.3	0.538
	Ours	29.81	0.774	**21.15**	**0.067**	24.29	0.642	**36.02**	**0.110**
Super-Resolution ($\times 12$)	DAPS (SD2)	27.61	0.727	41.55	0.177	18.02	0.415	80.84	0.368
	DAPS (SD3)	24.54	0.535	81.93	0.309	19.13	0.347	106.9	0.379
	RLSD (SD2)	22.18	0.644	133.5	0.404	18.81	0.450	187.3	0.430
	RLSD (SD3)	19.53	0.664	125.0	0.424	17.04	0.453	172.0	0.553
	Ours	27.12	0.716	34.54	**0.158**	21.16	0.505	**65.52**	**0.267**

Flower [21]. We use 40 NFEs, 15 optimization steps per iteration, and a CFG scale of 2.0. Consistent with the authors' observations, we found that the optional uncertainty term destabilized restoration; thus, we set $\gamma = 0$.

B.3 Additional Baseline Comparisons

Original diffusion-backbone results. Although the main experiments adapt all baselines to the matched SD3 backbone, we additionally report DAPS and RLSD using their original SD2.1 diffusion backbone for reference. For the original DAPS setting, we use 5 ODE steps and 50 annealing steps, resulting in 250 NFEs. Hyperparameters were selected by grid search on a validation set of 200 images; for example, we use 50 MCMC steps with learning rate 10^{-6} and CFG scale 6.0 for Gaussian deblurring, and 50 MCMC steps with learning rate 10^{-5} and CFG scale 5.0 for super-resolution. For the original RLSD setting, we use 250 NFEs, disable the repulsive term, and set the CFG scale to 6.0 for all tasks. Table 5 reports DAPS and RLSD under their original SD2.1 diffusion-backbone protocols, together with our SD3 adaptations and FlowLPS. These results are provided for reference, since backbone choice and inference budget can substantially affect performance. Our main claims are based on the matched SD3 setting, while this table provides additional context on backbone-dependent performance variations.

Computational Costs. We report runtime comparisons for the inpainting task on an RTX 3090 GPU. For SD3-adapted baselines, we use the matched 40-NFE protocol from the main experiments. For the original SD2 diffusion baselines, we report their original high-budget 250-NFE protocols for reference.

RLSD (SD3) and FlowChef are faster because they use only a single measurement update per reverse step. In contrast, FlowDPS, FLAIR, Flower, and FlowLPS perform repeated measurement updates, leading to comparable runtimes. FlowLPS has a runtime similar to FLAIR and Flower, while achieving stronger reconstruction quality. DAPS (SD3) is faster under the 40-NFE setting because its decoupled design performs measurement updates over fewer outer reverse steps.

Higher-budget DAPS. Under the matched 40-NFE protocol, DAPS uses 20 outer reverse steps because each outer step requires 2 denoised-prediction ODE evaluations. Since measurement updates are performed at each outer step, this means that DAPS applies measurement correction at only half as many reverse timesteps as other 40-step baselines. To isolate the effect of this reduced number of

Table 6: **Runtime comparison on the inpainting task.** We report seconds per image on an RTX 3090 GPU. SD3-adapted baselines follow the matched 40-NFE protocol, while original SD2 baselines use 250 NFEs.

Method	DAPS (SD2)	DAPS (SD3)	RLSD (SD2)	RLSD (SD3)	FlowDPS	FlowChef	FLAIR	Flower	FlowLPS (Ours)
Runtime (s)	1005	107.6	332.9	27.46	207.1	23.06	288.7	270.8	264.5
NFE	250	40	250	40	40	40	40	40	40

Table 7: **Higher-budget DAPS experiments on FFHQ 1k dataset. Bold**: best. Underline: second best.

Task	Method	NFE	FFHQ 1k PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)
Inpainting (Box)	DAPS	40	21.78	0.842	24.75	0.114
	DAPS	120	**22.10**	0.848	24.57	**0.107**
	Ours	40	21.68	**0.849**	**21.90**	**0.107**
Inpainting (Random)	DAPS	40	33.41	0.857	13.28	0.034
	DAPS	120	**34.99**	**0.906**	9.811	0.025
	Ours	40	34.34	0.886	**7.460**	**0.019**
Gaussian Deblurring	DAPS	40	27.21	0.715	41.72	0.171
	DAPS	120	27.09	0.710	39.53	0.167
	Ours	40	**27.76**	**0.737**	**31.22**	**0.130**
Motion Deblurring	DAPS	40	28.96	0.754	31.71	0.116
	DAPS	120	28.80	0.747	29.85	0.112
	Ours	40	**29.81**	**0.774**	**21.15**	**0.067**
Super-Resolution ($\times 12$)	DAPS	40	24.54	0.535	81.93	0.309
	DAPS	120	24.25	0.546	71.26	0.283
	Ours	40	**27.12**	**0.716**	**34.54**	**0.158**

measurement-correction timesteps, we additionally evaluate a higher-budget DAPS variant with 40 outer reverse steps and 3 inner denoised-prediction ODE evaluations, resulting in 120 NFEs. The results are reported in Table 7.

As shown in Table 7, increasing the budget improves DAPS in several perceptual metrics, especially FID and LPIPS. However, even with 120 NFEs, DAPS remains consistently behind FlowLPS in perceptual quality and often in pixel-level fidelity as well. This supports our interpretation that, under practical inference budgets, Langevin-only posterior exploration struggles to consistently concentrate into sharp, measurement-consistent reconstructions.

B.4 Pixel-Space Inverse Problems and Phase Retrieval

Although FlowLPS is primarily designed for latent-space flow models, its Langevin-Proximal principle is not restricted to latent representations. We therefore evaluate FlowLPS with a pixel-space flow model [4] on CelebA-HQ 1k. In addition to linear inverse problems, we also evaluate phase retrieval as a representative nonlinear inverse problem. For phase retrieval, which is sensitive to initialization due to non-uniqueness, we generate 4 reconstructions for each method and report the sample with the best PSNR.

Table 8 shows the results. Consistent with the latent-space experiments, Flower achieves strong pixel-level metrics but weaker perceptual scores, often leading to blurry reconstructions. In contrast, FlowLPS achieves stronger perceptual quality while maintaining competitive pixel-level fidelity. On phase retrieval, FlowLPS achieves the best perceptual scores and competitive distortion metrics, suggesting that the proposed Langevin-Proximal principle is not limited to linear inverse problems.

C Additional Ablation Studies

C.1 Effectiveness of the Langevin-Proximal Update

While we demonstrated the quantitative trade-off of Langevin steps in Section 5.3.1 of the main paper, here we provide additional visual evidence to further clarify the impact of N_L. Fig. 7 compares the visual quality across different N_L settings: $N_L = 0$ (pure optimization) leads to high FID/LPIPS due to blurriness. Moderate stochasticity ($N_L = 5 - 7$) achieves an optimal balance, producing

Table 8: **Pixel-space results on CelebA-HQ 1k** with $\sigma_n = 0.03$. **Bold**: best. Underline: second best.

Task	Method	CelebA-HQ 1k PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)
Inpainting (Box)	DAPS	24.14	0.775	41.43	0.128
	RLSD	22.61	0.557	53.15	0.205
	FlowChef	18.16	0.633	35.58	0.296
	FlowDPS	23.08	0.814	48.81	0.171
	Flower	**25.66**	**0.835**	35.51	0.107
	Ours	24.46	0.821	**26.91**	**0.106**
Super-Resolution (x8)	DAPS	21.57	0.362	92.61	0.394
	RLSD	24.05	0.430	75.57	0.336
	FlowChef	20.89	0.563	37.13	0.326
	FlowDPS	26.87	0.766	49.19	**0.203**
	Flower	**27.35**	**0.770**	45.57	0.232
	Ours	26.05	0.725	**28.61**	0.214
Phase Retrieval	DAPS	21.82	0.611	117.2	0.352
	RLSD	15.35	0.380	154.5	0.503
	FlowChef	16.61	0.397	224.8	0.535
	FlowDPS	21.49	0.634	106.7	0.359
	Flower	**31.08**	**0.784**	69.37	0.220
	Ours	30.75	0.775	**68.10**	**0.218**

detailed restorations while maintaining measurement consistency. Conversely, excessive Langevin steps ($N_L > 10$) degrade perceptual quality, introducing high-frequency artifacts and grainy noise patterns. Fig. 7 visualizes these extremes, contrasting the over-smoothed textures of pure optimization ($N_L = 0$) with the noisy artifacts of excessive Langevin dynamics ($N_L = 15$).

C.2 Dynamic Update-Budget Allocation

In this section, we analyze whether the update budget of FlowLPS can be reduced without substantially degrading reconstruction quality.

Decaying Langevin Steps (N_L). We hypothesize that the benefit of Langevin updates decreases over reverse time. In early sampling stages (high t), the clean estimate is less reliable, so posterior-oriented Langevin exploration is more useful. In later stages (low t), the flow prediction becomes more stable, making intensive Langevin updates less necessary.

To test this, we implement a linear decay schedule for N_L, reducing it toward zero at the end of sampling. Our goal is to reduce the update budget to $N_T = N_L + N_P = 10$ in the final block. For deblurring, we reduce N_L from $6 \to 1$. For super-resolution and inpainting ($4 \to 0$), we additionally remove one proximal step to match the $N_T = 10$ target. As shown in Table 9, this dynamic allocation maintains performance close to the fixed schedule, suggesting that Langevin updates are most useful in early stages and can be reduced later to improve efficiency.

Table 9: **Performance comparison: Fixed vs. Decaying N_L.** "Decay" indicates a linear reduction of Langevin steps over iterations, adjusted to meet a reduced total budget ($N_T = 10$) in the final sampling block. The results show that tapering off exploration in later stages maintains performance.

Task	N_L	PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)
Inpainting (Box)	Fixed (4)	21.89	**0.854**	**37.40**	**0.101**
	Decay ($4 \to 0$)	21.68	0.842	**37.00**	0.108
Gaussian Deblurring	Fixed (6)	27.83	0.744	**53.83**	**0.126**
	Decay ($6 \to 1$)	**27.86**	**0.750**	54.13	0.129
Motion Deblurring	Fixed (6)	29.86	0.780	**35.97**	**0.065**
	Decay ($6 \to 1$)	**30.06**	**0.790**	36.94	0.067
Super-Resolution ($\times 12$)	Fixed (4)	**27.12**	0.720	62.44	**0.154**
	Decay ($4 \to 0$)	27.09	**0.724**	**62.38**	0.156

Impact of Total Budget (N_T). Separately, we investigated the trade-off between performance and the total number of Langevin-Proximal steps $N_T = N_L + N_P$. Fig. 8 indicates that performance

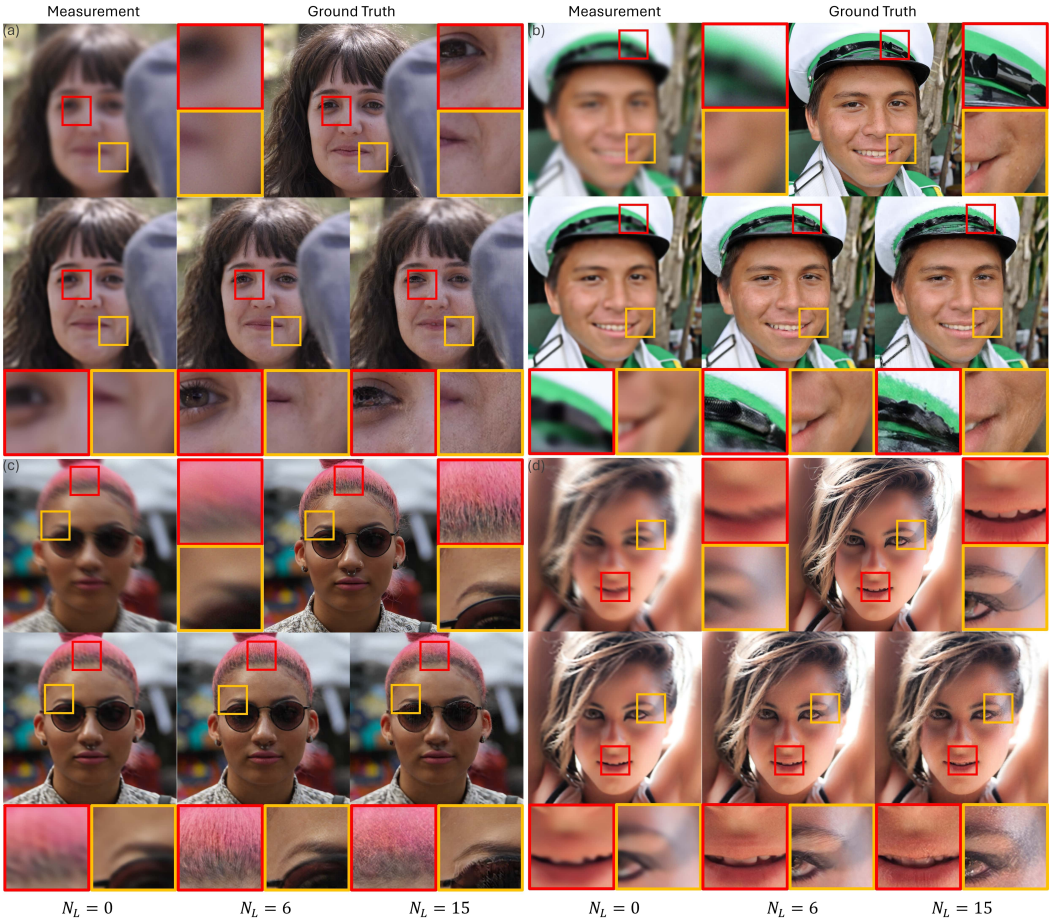

Figure 7: **Visual artifacts from excessive Langevin steps.** (a), (b) Gaussian deblurring, (c), (d) Motion deblurring. $N_L = 0$ yields over-smoothed results. $N_L = 15$ introduces speckled, high-frequency noise. $N_L = 6$ produces clean, high-fidelity images.

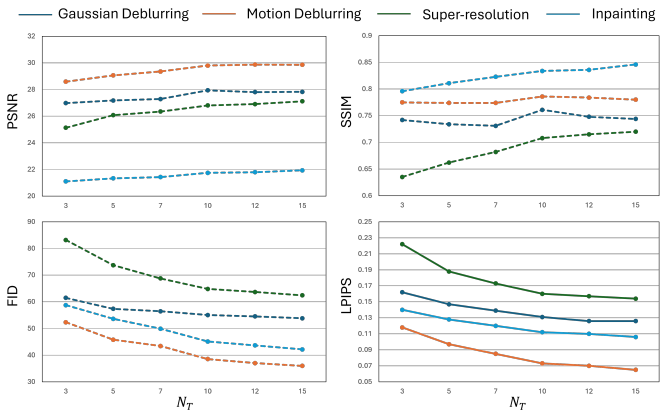

Figure 8: **Performance vs. Total Steps ($N_T = N_P + N_L$).** Performance gains show diminishing returns beyond $N_T \approx 10$, suggesting that $N_T = 15$ serves as a robust operating point for maximum quality, while lower N_T remains viable for efficiency.

gains show diminishing returns beyond $N_T \approx 10$. While we utilized a fixed $N_T = 15$ in our main experiments to prioritize maximum quality, these results suggest that a lower budget (e.g., $N_T = 10$) remains a viable choice for time-constrained applications.

Runtime Analysis. To quantify practical efficiency, we measured inference time on the inpainting task using an NVIDIA RTX 3090 GPU. We evaluated two VAE backbones: the standard "Large VAE" (SD3) and the distilled "Tiny VAE" [19]. Table 10 summarizes the results. The column "$15 \to 10$" represents the N_L decay strategy. As shown, the decay strategy reduces inference time compared to the fixed $N_T = 15$ setting (e.g., 262.2s $\to$ 232.7s with Large VAE) while preserving quality. Furthermore, utilizing the Tiny VAE significantly accelerates the process (down to $\approx$ 30s), demonstrating that FlowLPS can be substantially accelerated with an efficient decoder.

Table 10: **Runtime analysis on inpainting task.** We compare the inference time (seconds per image) across different total budgets (N_T). "$15 \to 10$" denotes the dynamic decay strategy where N_L is linearly reduced. "Large VAE" refers to the standard Stable Diffusion VAE, while "Tiny VAE" refers to the distilled efficient decoder used in FLAIR.

N_T	3	5	7	10	12	$15 \to 10$	15
large VAE	66.13	101.1	161.6	183.8	217.8	232.7	262.2
tiny VAE	15.63	18.48	21.79	24.74	27.70	29.34	31.65

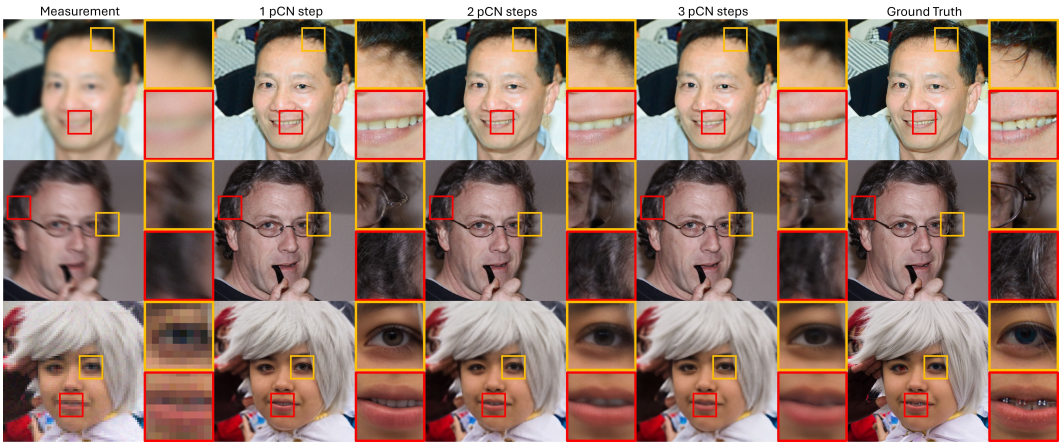

Figure 9: **Effect of multiple pCN steps.** Increasing pCN steps leads to progressive over-smoothing.

C.3 Detailed Analysis of pCN Steps

C.3.1 Impact of Number of pCN Re-noising Steps.

For completeness, we recall the Gaussian-preserving property underlying the pCN-style refresh. If the target distribution is the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, the corresponding pCN proposal has acceptance probability:

$$\begin{aligned} &\min\left\{1, \frac{p_{\text{tar}}(\hat{\mathbf{z}}_{\text{new}}) q_{\text{prop}}(\hat{\mathbf{z}}_{\text{old}}|\hat{\mathbf{z}}_{\text{new}})}{p_{\text{tar}}(\hat{\mathbf{z}}_{\text{old}}) q_{\text{prop}}(\hat{\mathbf{z}}_{\text{new}}|\hat{\mathbf{z}}_{\text{old}})}\right\} \\ =&\min\left\{1, \frac{\mathcal{N}(\hat{\mathbf{z}}_{\text{new}}; 0, I)\mathcal{N}(\hat{\mathbf{z}}_{\text{old}}; \rho_t \hat{\mathbf{z}}_{\text{new}}, (1-\rho_t^2)I)}{\mathcal{N}(\hat{\mathbf{z}}_{\text{old}}; 0, I)\mathcal{N}(\hat{\mathbf{z}}_{\text{new}}; \rho_t \hat{\mathbf{z}}_{\text{old}}, (1-\rho_t^2)I)}\right\} \\ =&\min\{1, 1\} = 1. \end{aligned} \tag{36}$$

Thus, in this idealized Gaussian case, every pCN proposal is accepted. This idealized property motivates the variance-preserving form of the refresh, although FlowLPS initializes it from the model-predicted noise in practice.

Table 11 compares performance across varying numbers of pCN steps. We observe that increasing the number of pCN steps slightly improves PSNR but degrades perceptual metrics. As visualized in Fig. 9, multiple pCN steps dilute the structural information embedded in the predicted noise $\hat{\mathbf{z}}_{1|t}^{(0)}$, leading to progressive over-smoothing.

Hypothesizing that stronger stochastic refresh is beneficial primarily in early stages, we further experiment with a linear decay schedule for pCN steps ($4 \rightarrow 1$). As shown in Table 11, this decay strategy improves FID in generation-heavy tasks such as box inpainting and super-resolution. However, this perceptual improvement comes at the cost of pixel-level fidelity, and in deblurring tasks the decay strategy offers only marginal benefits over the single-step baseline. Therefore, considering computational efficiency, consistent performance across tasks, and implementation simplicity, we use a single pCN step as the default choice.

Table 11: **Ablation on the number of pCN-style re-noising steps (FFHQ 0.2k).** "Decay ($4 \rightarrow 1$)" indicates a linear reduction of pCN steps from 4 to 1 over the sampling process. While the decay strategy offers the best FID for inpainting and super-resolution, the single-step approach provides the best overall balance.

Task	# pCN	PSNR (↑)	SSIM (↑)	FID (↓)	LPIPS (↓)
Inpainting (Box)	1	21.94	**0.846**	42.15	**0.106**
	2	21.67	0.842	42.55	0.110
	3	**21.98**	**0.846**	43.87	0.108
	Decay ($4 \rightarrow 1$)	21.46	0.843	**40.34**	0.110
Gaussian Deblurring	1	27.83	0.744	**53.83**	**0.126**
	2	27.97	0.750	54.62	0.129
	3	**28.01**	**0.752**	56.36	0.129
	Decay ($4 \rightarrow 1$)	27.75	0.745	54.32	0.128
Motion Deblurring	1	29.86	0.780	**35.97**	**0.065**
	2	30.10	0.789	37.09	0.067
	3	**30.13**	**0.790**	37.60	0.067
	Decay ($4 \rightarrow 1$)	30.02	0.786	36.90	0.067
Super-Resolution ($\times 12$)	1	27.12	0.720	62.44	**0.154**
	2	**27.14**	**0.725**	65.04	0.158
	3	**27.14**	**0.725**	65.46	0.161
	Decay ($4 \rightarrow 1$)	26.92	0.715	**62.03**	0.156

C.3.2 Qualitative Analysis of ρ_t Schedules

Complementing the quantitative results in Section 5.3.2, we provide visual comparisons of different ρ_t schedules in Fig. 10. As discussed, the adaptive schedule $\rho_t = \sqrt{1-t}$ reconstructs continuous semantic details, such as hair strands across the mask, whereas other schedules often fail to maintain such structural continuity or produce artifacts. This visual evidence reinforces the advantage of the adaptive mixing schedule shown in Table 2.

C.4 Component-wise Effect of Langevin Updates and pCN-style Re-Noising

We performed a controlled 2×2 ablation over Langevin updates and pCN-style re-noising, while keeping the proximal stage fixed on the inpainting task. Table 12 shows that both components help and that their combination gives the best perceptual quality. This suggests that the gain does not come from a single stochastic component: Langevin updates provide posterior-oriented initialization, while pCN-style re-noising supplies complementary stochastic refreshment.

Table 12: **Component-wise ablation of Langevin updates and pCN-style re-noising on box inpainting.** Disabling Langevin sets $N_L = 0$. Disabling controlled pCN-style re-noising uses independent Gaussian re-noising ($\rho_t = 0$). Both components improve FID, and their combination yields the strongest perceptual quality under the fixed update budget.

pCN re-noising	Langevin	PSNR ↑	FID ↓
✗	✗	22.62	45.37
✓	✗	21.88	28.81
✗	✓	**22.74**	30.95
✓	✓	21.68	**21.90**

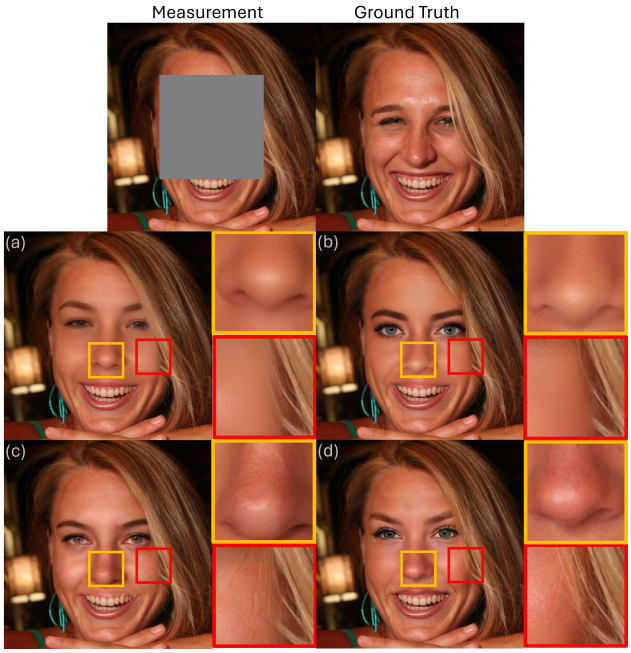

Figure 10: **Qualitative comparison of ρ_t.** (a) $\rho_t = 0$, (b) $\rho_t = 0.5$, (c) $\rho_t = 1 - t$, (d) $\rho_t = \sqrt{1-t}$. Our adaptive schedule (d) best preserves high-frequency details and semantic continuity (e.g., hair strands) compared to other schedules.

D Additional Qualitative Results

D.1 Visual Analysis of Generative Bias in FLAIR

As discussed in the main paper (Sec. 5.2), FLAIR can produce sharp but less measurement-faithful reconstructions. Fig. 11 presents a detailed visual comparison. While FLAIR generates sharp images, it can alter semantic attributes such as facial expressions or specific features (e.g., eye shape, wrinkles), suggesting a stronger reliance on the generative prior than on the measurement. In contrast, FlowLPS better balances prior plausibility with measurement-consistent semantic structure.

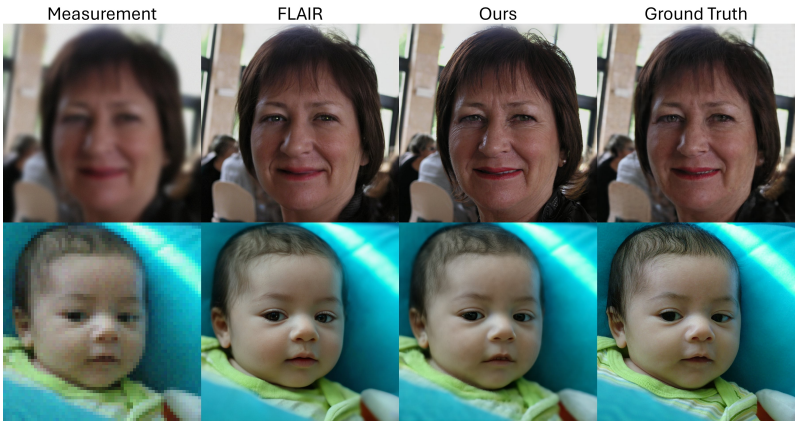

Figure 11: **Visual evidence of generative bias in FLAIR.** Top: Gaussian Deblurring, Bottom: Super-Resolution. While FLAIR generates sharp images, it ignores the measurement's underlying structure in favor of its generative prior. Note how FLAIR alters facial features (e.g., eye shape), prioritizing the generation of "plausible" faces over the recovery of the true identity. In contrast, FlowLPS better balances prior plausibility with measurement-consistent semantic structure.

D.2 Sample Diversity

While FlowLPS includes a proximal refinement step, it remains different from deterministic refinement-dominant solvers. Its stochasticity comes from two sources: Langevin updates during posterior-oriented initialization and pCN-style re-noising along the reverse trajectory. These components allow different random seeds to produce distinct posterior-oriented initializations, after which the proximal step refines each trajectory toward a nearby measurement-consistent reconstruction. Thus, repeated runs can yield diverse valid solutions rather than collapsing to a single deterministic output.

To quantitatively evaluate diversity, we follow the RLSD protocol. For each measurement, we generate 32 samples, retain only samples whose measurement residual is below a fixed validity threshold, and compute the average pairwise cosine similarity of DINO features among the valid samples; lower similarity indicates higher diversity. We also report the valid-sample ratio, since diversity among measurement-inconsistent samples can be misleading.

Table 13: **Diversity analysis on the inpainting task.** We report pairwise DINO cosine similarity among valid samples and the valid-sample ratio. Lower cosine similarity indicates higher diversity.

Methods	DAPS (SD3)	FlowDPS	FlowChef	Flower	FlowLPS (Ours)
Pairwise DINO cosine ↓	0.9681	0.9215	0.8341	0.9823	0.9714
Valid-sample ratio ↑	1.0	0.47	0.12	1.0	1.0

We omit RLSD because it rarely produced valid samples under the same threshold, making its diversity score unreliable. FlowDPS and FlowChef achieve lower pairwise similarity, but their valid-sample ratios are substantially lower, indicating that much of their apparent diversity comes from measurement-inconsistent reconstructions. By contrast, FlowLPS maintains a 100% valid-sample ratio while retaining competitive diversity. Among methods with a 100% valid-sample ratio, FlowLPS is much closer to DAPS than to the refinement-dominant baseline Flower, suggesting that proximal refinement does not reduce FlowLPS to a deterministic refinement-only solver.

Figure 12 demonstrates this behavior on box inpainting. Given the same masked input, FlowLPS generates multiple valid reconstructions with meaningful variations in high-frequency details while remaining consistent with the unmasked regions. This suggests that FlowLPS preserves stochastic solution diversity while maintaining measurement consistency.

D.3 Full Reconstruction Results

Additional qualitative comparisons on FFHQ and DIV2K across all tasks are provided in the supplementary material. These results complement the representative visual comparisons in the main paper and further illustrate that FlowLPS preserves sharper textures and natural local details while maintaining measurement consistency.

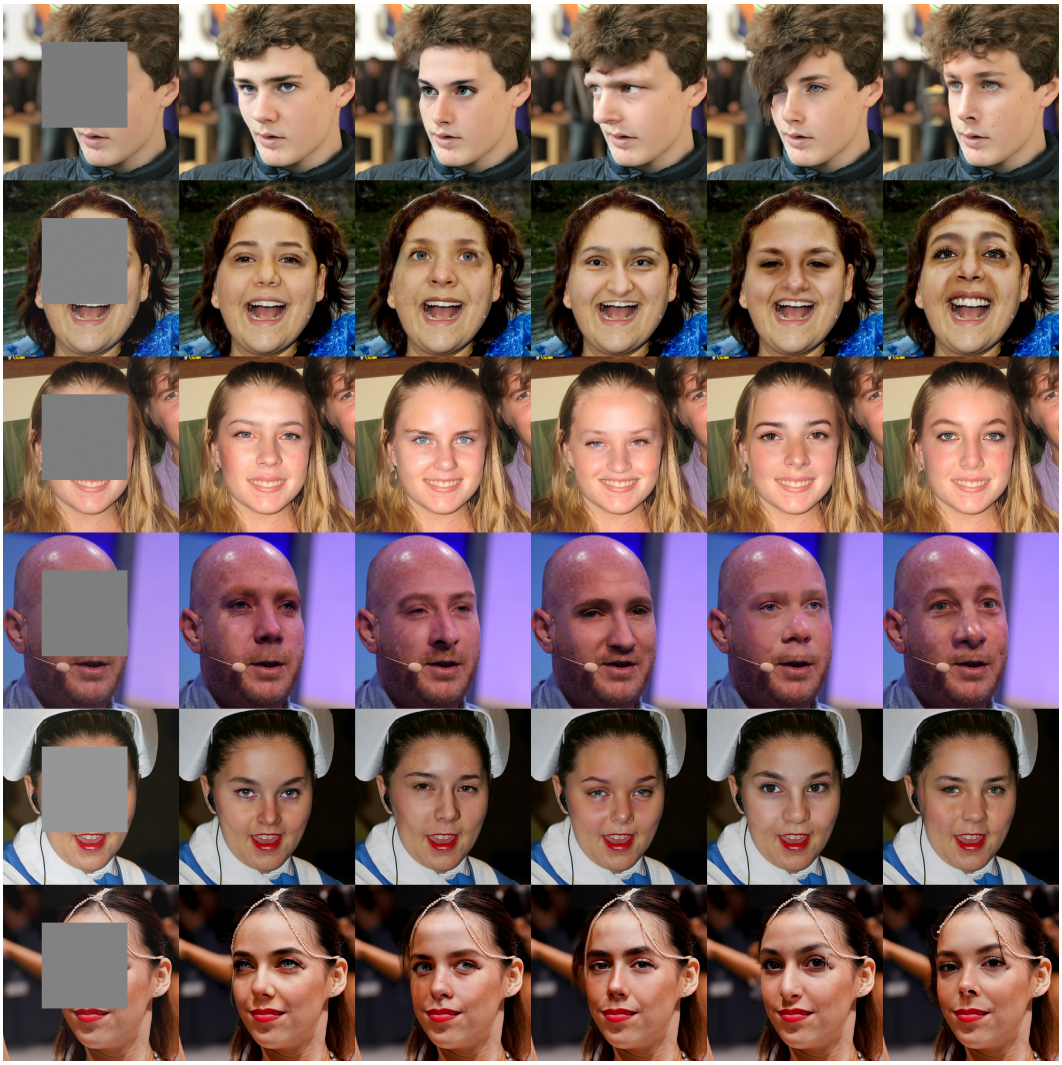

Figure 12: **Diversity of solutions in box inpainting.** Given the same masked input (leftmost column), FlowLPS generates distinct yet plausible measurement-consistent reconstructions. The results suggest that the Langevin-Proximal design can produce multiple valid completions rather than collapsing to a single deterministic reconstruction.

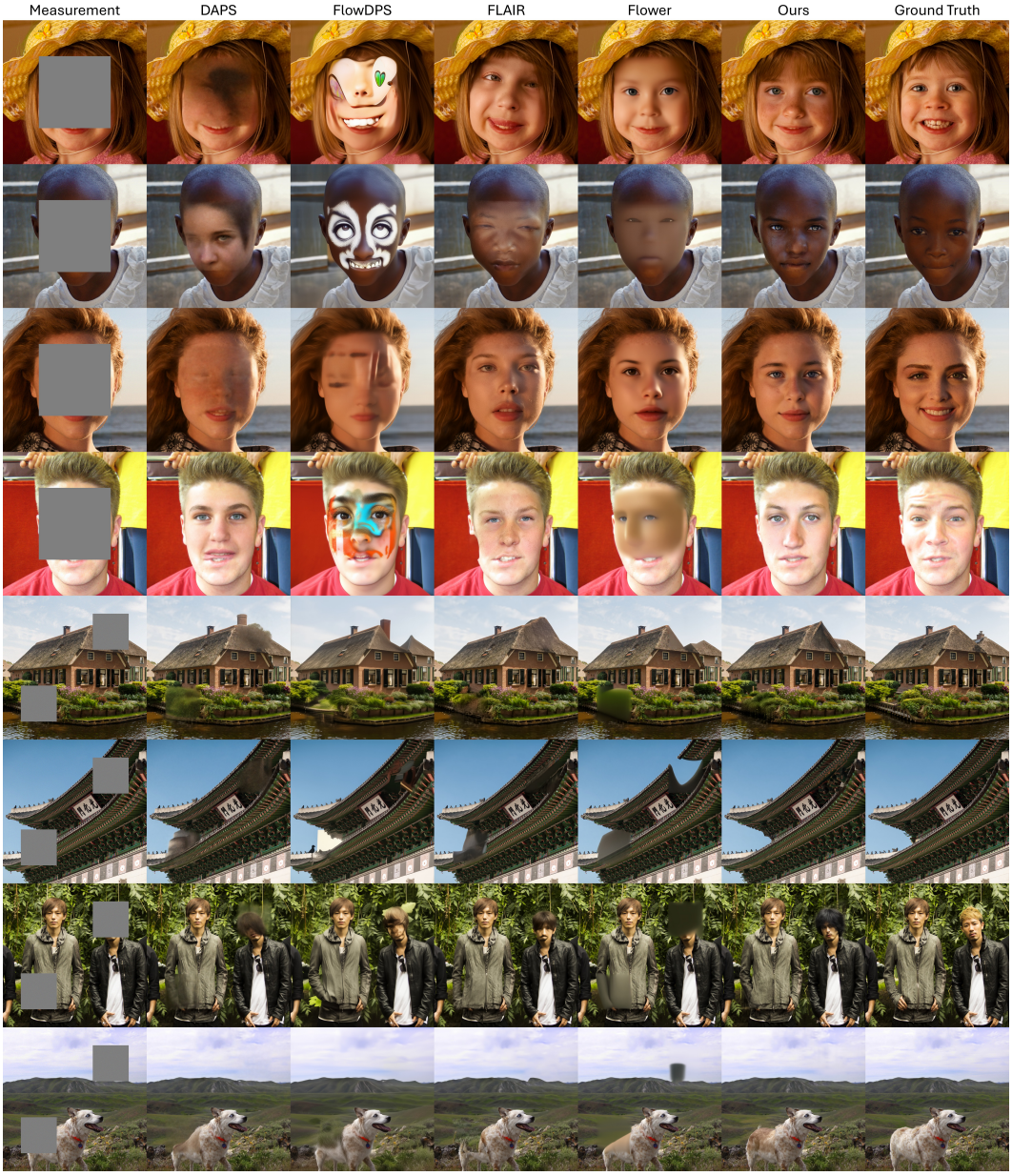

Figure 13: Box Inpainting results on FFHQ and DIV2K.

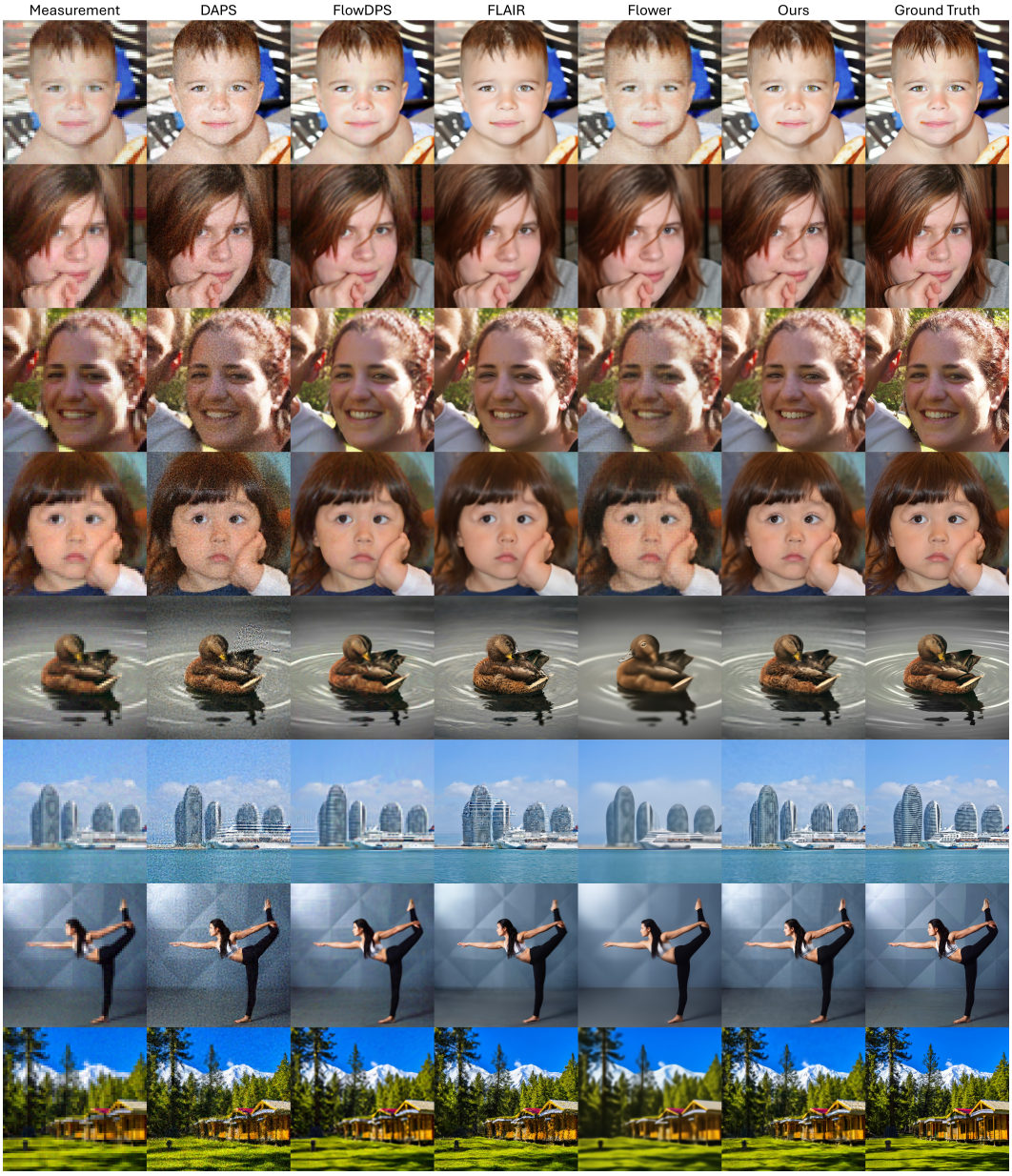

Figure 14: Super-Resolution($\times 12$) results on FFHQ and DIV2K.

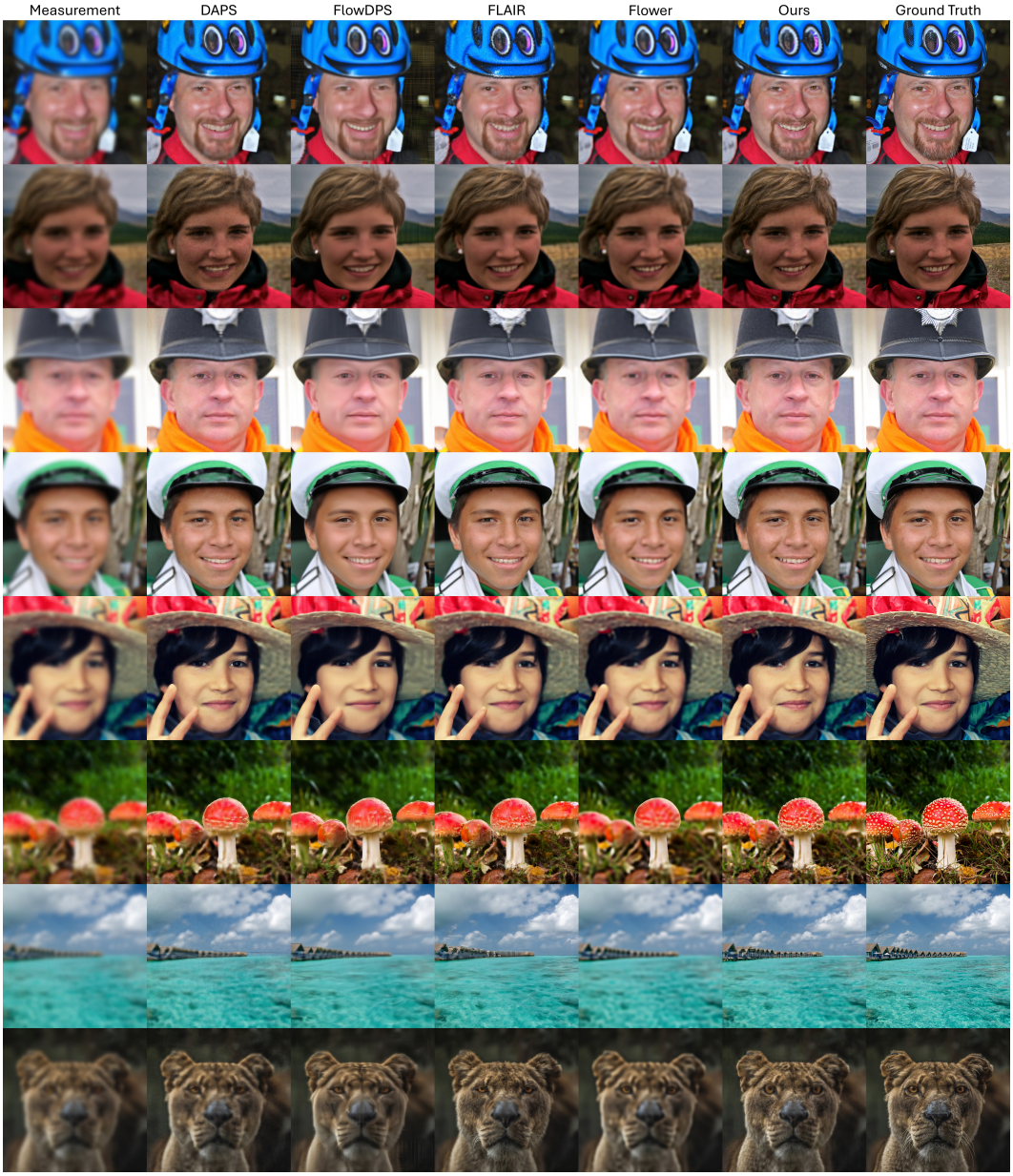

Figure 15: Gaussian deblurring results on FFHQ and DIV2K.