FLOWSGS: IMPROVING FLOW PRIORS FOR INVERSE IMAGING WITH STOCHASTIC INTERPOLANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching has emerged as the state-of-the-art generative model and has been used for plug-and-play (PnP) priors to solve inverse problems in computational imaging. However, existing flow-based inverse solvers make simplifying approximations in posterior sampling and/or assume linear forward models. In contrast, principled diffusion-based methods exist but are computationally expensive. To circumvent these problems, we introduce **FlowSGS**, a flow-based posterior sampling method using Split Gibbs Sampling (SGS) with annealed coupling, and make flow-specific interventions to the prior sampling and annealing steps, theoretically grounded in the Stochastic Interpolants (SI) framework. Using a novel timestep correction and accelerated warm-up with one-step flow models, our method is faster and more accurate than the state of the art across a range of imaging problems. We demonstrate the first use of flow-based samplers for a nonlinear inverse problem (Fourier phase retrieval) and apply one-step flow models to principled Bayesian inverse imaging.

1 INTRODUCTION

Computational imaging systems aim to reconstruct an image $\mathbf{x} \in \mathbb{R}^n$ encoded in incomplete and noisy measurements $\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathbf{n} \in \mathbb{R}^m$, where $\mathcal{A}(\cdot)$ is the forward model and $\mathbf{n}$ represents noise. These inverse problems are often ill-posed due to information loss, leading to uncertainty in the reconstructed image $\hat{\mathbf{x}}$. To fully explore the solution space, it is often desired to sample from the Bayesian posterior

$$\hat{\mathbf{x}} \sim p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}), \tag{1}$$

where $p(\mathbf{y}|\mathbf{x})$ is the data likelihood and $p(\mathbf{x})$ is the prior. This is particularly important for scientific and medical imaging, where high-stakes decisions are being made in downstream applications.

Deep generative models have significantly improved posterior sampling methods by implicitly capturing complex priors. Specifically, score-based diffusion models (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021) are capable of learning complex data distributions of images and videos, and have been widely used to learn expressive prior distributions $p(\mathbf{x})$. Most diffusion-based posterior samplers condition on the measurements $\mathbf{y}$, either by taking gradient steps towards higher data likelihood (e.g., DPS (Chung et al., 2023); Jalal et al. (2021); Kawar et al. (2022); Graikos et al. (2022); Mardani et al. (2024)), or by projecting onto the measurement space (e.g., Song et al., 2022; Chung & Ye, 2022; Chung et al., 2022). PnP-DM (Wu et al., 2024) and several others (Zhu et al., 2023; Xu & Chi, 2024; Coeurdoux et al., 2024; Chu et al., 2025) leverage variable splitting to decompose the sampling into a likelihood step and a prior step, where a pretrained diffusion model is incorporated as a PnP prior. InverseBench (Zheng et al., 2025) provides a comprehensive benchmark of diffusion samplers on a variety of real-world inverse problems, where PnP-DM (Wu et al., 2024) and DAPS (Zhang et al., 2025) stand out as the state-of-the-art diffusion samplers, although they are computationally expensive due to solving the reverse-time SDE repeatedly. Zhao et al. (2024); Garber & Tirer (2025); Gulle et al. (2026) leverage consistency models to increase efficiency.

Flow matching is an emerging generative technique that improves upon diffusion models by constructing straighter probability paths, which provide more efficient sampling, easier training, and higher quality data generation (Lipman et al., 2023; Liu et al., 2023). However, existing flow-based inverse solvers rely on mathematical approximations (Pokle et al., 2024; Ben-Hamu et al., 2024; Zhang et al., 2024; Yan et al., 2025; Martin et al., 2025; Patel et al., 2025; Kim et al., 2025; Askari et al., 2025) or assume a linear forward model (Pokle et al., 2024; Zhang et al., 2024; Yan et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

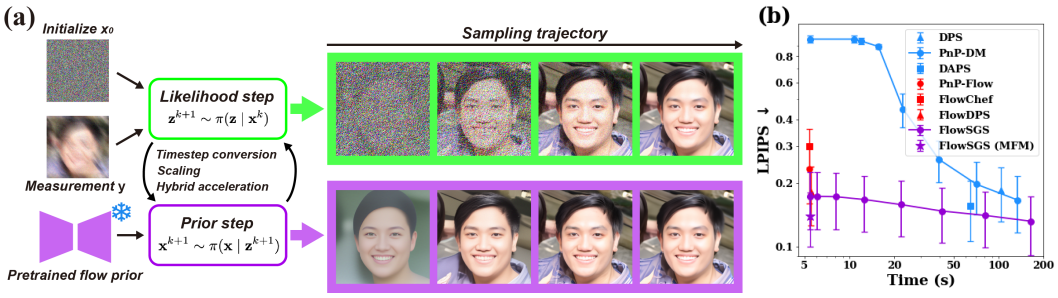

Figure 1: **A high-level overview of FlowSGS, a principled posterior sampler using flow matching as a plug-and-play prior.** (a) Our method iterates between a likelihood step and a prior step, where the former uses proximal sampling or Langevin dynamics as appropriate to the problem, and the latter involves solving a reverse-time SDE using a pretrained flow prior. (b) Our method shows high efficiency with flow priors, shown here averaged across 20 samples of motion deblurring.

2025; Martin et al., 2025; Askari et al., 2025). Specifically, PnP-Flow (Martin et al., 2025) iterates between a data-fidelity gradient step, an interpolation step, and a denoising step that uses a flow network as a time-dependent denoiser. FlowChef (Patel et al., 2025) uses a gradient-free vector field to steer the ODE towards high measurement likelihood. FlowDPS (Kim et al., 2025) derives the flow version of Tweedie's formula, and estimates both the clean image and noise component to simultaneously maintain data-consistency and generative quality. Each of these approaches relies on approximations that shift their posterior samples away from the Bayesian posterior.

We present **FlowSGS**, a method that avoids the approximations used by current flow-based samplers through Split Gibbs Sampling (SGS) of the likelihood and prior steps, enabling us to target the actual Bayesian posterior. We adapt SGS to flow matching using the Stochastic Interpolants (SI) framework, which has revealed the deep connection between diffusion models' score functions and flow matching's velocity fields (Albergo et al., 2025). A high-level overview of FlowSGS and its SOTA performance and efficiency is shown in Figure 1. Our experiments in Section 4 show superior image reconstruction compared to both diffusion- and flow-based methods. We demonstrate accurate posterior sampling on a toy problem with a known posterior, and show high image quality on linear (deblurring, super-resolution, compressed-sensing MRI) and nonlinear (Fourier phase retrieval) tasks. Notably, FlowSGS is the first flow matching posterior sampler demonstrated on a nonlinear inverse problem (which is more sensitive to approximations in likelihood incorporation). We also leverage one-step flow models (Potaptchik et al., 2026) (see also Geng et al. (2025; 2026); Lu et al. (2026)) to accelerate our solution without losing accuracy.

2 Foundational Related Work

2.1 Stochastic Interpolants

Stochastic interpolants (SI, Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025) provide a unified framework for diffusion models and flow matching, which both consider the process of interpolating data distribution $p(\mathbf{x}_0)$ and noise (usually a standard normal distribution $\mathcal{N}(0, \mathbf{I})$) for generative modeling. Such interpolation processes have the form

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{2}$$

where (α_t, σ_t) are a pair of smooth functions defined in $t \in [0, 1]$, satisfying: (i) $\alpha_t^2 + \sigma_t^2 > 0$ for all $t \in [0, 1]$; (ii) α_t and σ_t are differentiable on $[0, 1]$; and (iii) $\alpha_0 = \sigma_1 = 1, \alpha_1 = \sigma_0 = 0$. Under these conditions, the stochastic process in Equation (2) interpolates without bias between clean data $p(\mathbf{x_0})$ at $t = 0$ and isotropic Gaussian noise $\mathcal{N}(0, \mathbf{I})$ at $t = 1$. To draw samples from the data distribution using the interpolant process, one can solve the reverse-time SDE (Anderson, 1982; Albergo et al., 2025) given by

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{v}(\mathbf{x}_t, t) - \frac{1}{2} w_t \mathbf{s}(\mathbf{x}_t, t)\right] \mathrm{d}t + \sqrt{w_t}\, \mathrm{d}\bar{\mathbf{w}}_t \tag{3}$$

where $\mathbf{v}(\mathbf{x}, t) = \mathbb{E}\left[\dot{\mathbf{x}}_t | \mathbf{x}_t = \mathbf{x}\right]$ is the velocity field, $\mathbf{s}(\mathbf{x}, t) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x})$ is the score function, and $w_t \geq 0$ is a diffusion coefficient picked at sampling time, and $\bar{\mathbf{w}}_t$ is the reverse-time Wiener

process. Different choices of the interpolant schedule (α_t, σ_t) have been used in diffusion models and flow matching (e.g., the VP-SDE used in score-based diffusion models (Ho et al., 2020; Song et al., 2021), the linear interpolant used in Rectified flow (Liu et al., 2023)). We summarize common choices of the interpolant schedules (α_t, σ_t) and diffusion coefficients w_t in Appendix A.1. While sampling with Equation (3) requires training both a flow model for the velocity field $\mathbf{v}(\mathbf{x}, t)$ and a diffusion model for the score function $\mathbf{s}(\mathbf{x}, t)$, prior work (Albergo et al., 2025; Ma et al., 2024) has shown that the velocity and the score convert from one to the other:

$$\mathbf{s}(\mathbf{x}_t, t) = \frac{\alpha_t \mathbf{v}(\mathbf{x}_t, t) - \dot{\alpha}_t \mathbf{x}_t}{\sigma_t \gamma_t}, \tag{4}$$

where $\gamma_t \triangleq \dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t < 0$. Equation (4) allows sampling with the reverse-time SDE in Equation (3) using a single pretrained score or velocity model. For instance, replacing $\mathbf{s}(\cdot, \cdot)$ with $\mathbf{v}(\cdot, \cdot)$ in Equation (3) yields

$$\mathrm{d}\mathbf{x}_t = \left[\left(1 - \frac{w_t \alpha_t}{2\sigma_t \gamma_t} \right) \mathbf{v}(\mathbf{x}_t, t) + \frac{w_t \dot{\alpha}_t}{2\sigma_t \gamma_t} \mathbf{x}_t \right] \mathrm{d}t + \sqrt{w_t}\, \mathrm{d}\bar{\mathbf{w}}_t. \tag{5}$$

When the diffusion coefficient $w_t = 0$, Equation (5) reduces to the probability-flow ODE which has been widely used in the sampling process of flow matching.

To enable one-step sampling, mean flows (Geng et al., 2025; 2026; Lu et al., 2026) learn the average velocity $\mathbf{u}(\mathbf{x}, t, s)$ between any two timestep $0 \le t \le s \le 1$ in probability-flow ODE. Meta Flow Maps (MFM, Potaptchik et al., 2026) further learn a stochastic map $X_{s,u}(\boldsymbol{\epsilon}; \mathbf{x}_t, t), \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$ to directly sample from the conditional posterior of the interpolant $p(\mathbf{x}_0 | \mathbf{x}_t, t)$.

2.2 Split Gibbs sampling

Variable splitting has been widely applied to inverse problem solving, even before the era of deep generative priors. Algorithms such as Half-Quadratic Splitting (HQS, Charbonnier et al., 1994; Geman & Yang, 1995), and the Alternating Direction Method of Multipliers (ADMM, Boyd & Vandenberghe, 2004; Neal et al., 2011) split the regularized MAP optimization into a likelihood step and a denoising step, where a pretrained denoiser can be plugged in Venkatakrishnan et al. (2013); Chan et al. (2016); Zhang et al. (2021); Kamilov et al. (2023). Unrolled methods further turn these plug-and-play solvers into a neural network, and allow task-specific training of the plug-in denoisers (Monga et al., 2021). However, these methods only produce a point estimate, rather than drawing random samples from the Bayesian posterior.

Split Gibbs Sampling (SGS, Vono et al., 2019) provides a solution for this and has already seen applications in posterior sampling with classic priors Pereyra et al. (2023), pretrained denoisers (Bouman & Buzzard, 2023; Faye et al., 2024), or diffusion priors (Xu & Chi, 2024; Wu et al., 2024; Chu et al., 2025). SGS decomposes Bayesian posterior sampling in Equation (1) into a likelihood step and a prior step, and alternates between sampling from these two steps until convergence. Specifically, in the k-th iteration, SGS has updates

$$\mathbf{z}^{k+1} \sim \pi(\mathbf{z} | \mathbf{x}^k) \propto \exp\left[-f(\mathbf{z}; \mathbf{y}) - \frac{\|\mathbf{x}^k - \mathbf{z}\|^2}{2\rho_k^2} \right], \qquad \triangleright \textbf{ Likelihood step} \tag{6}$$

$$\mathbf{x}^{k+1} \sim \pi(\mathbf{x} | \mathbf{z}^{k+1}) \propto \exp\left[-g(\mathbf{x}) - \frac{\|\mathbf{x} - \mathbf{z}^{k+1}\|^2}{2\rho_k^2} \right], \qquad \triangleright \textbf{ Prior step} \tag{7}$$

where $f(\mathbf{z}; \mathbf{y}) = -\log p(\mathbf{y} | \mathbf{z})$ is the negative log data likelihood, $g(\mathbf{x}) = -\log p(\mathbf{x})$ is the negative log prior, and $\{\rho_k\}_{k=0}^{K-1}$ are augmented Lagrangian variables, optionally with an annealing schedule (Wu et al., 2024). By introducing a modular structure in the sampling process, SGS solves the likelihood step and prior step independently, and simplifies the complexity of each subproblem.

3 Method

FlowSGS uses Split Gibbs Sampling (SGS) to alternate between a likelihood step and a prior step coupled by augmented Lagrangian parameters $\{\rho_k\}$. We sample from the likelihood step using proximal sampling for its speed on linear $\mathcal{A}(\cdot)$, or Langevin dynamics for nonlinear $\mathcal{A}(\cdot)$ (see details in Appendix C), and use a user-defined exponential decay annealing strategy for the coupling parameter $\rho_k = \max(\rho_0 \alpha^k, \rho_{\min})$ (see Appendix C.4). When $\rho_k > \rho_{\min}$, the sampler goes into a

warm-up stage, which acts as an initialization heuristic that we find accelerates convergence. When $\rho_{\min}$ is reached, the sampler goes into a *stationary* stage. Among existing methods, this is most similar to PnP-DM (Wu et al., 2024), which leverages the same SGS-based problem decomposition but uses a diffusion model in its prior step. We also include a flow-specific annealing acceleration.

Prior step We present several significant interventions in the prior step, expressed as Lines 4-11 in Algorithm 1, which accommodate the integration of flow matching. Line 4 computes the correct starting timestep t_k for the prior step in the k-th iteration, adapted to the current noise level ρ_k and the interpolant schedule of any pretrained flow model; see Theorem 1. Line 5 scales the output of the likelihood step $\mathbf{z}^k$ such that its noise level matches the noise σ_{t_k} in the interpolant at t_k. Line 10 samples from the prior step by solving the reverse-time SDE with the Euler-Maruyama method (Kloeden & Pearson, 1977). Line 11 shows the SI's reverse-time SDE using only the velocity field in the prior step, to our knowledge, used here for the first time in posterior sampling for inverse problems. Note that the SDE discretization error is more important in flow-based methods than in diffusion models, because the relative sampling step count is greatly reduced. We propose a novel timestep-correction technique to alleviate such error for fewer SDE steps. Specifically, instead of naively starting at the closest discrete timestep to t_k, we use an adaptive stepsize in the first Euler step and start from the correct time t_k; see Figure 5.

Lines 6-8 introduce a hybrid acceleration to the prior step. We draw intuition from the observation that the prior steps during the warm-up stage cost many more network evaluations than the stationary stage (see Appendix A.1), and can be accelerated with one-step flow models. However, such models are difficult to train (Potaptchik et al., 2026), and using them in the stationary stage leads to decreased image quality (see Section 4.4). We further observe that the prior step outputs $\mathbf{x}^k$ during warm-up (see Figure 6d) have low image quality compared to the stationary stage. Therefore, it is feasible to use one-step samplers during warm-up for acceleration, but keep flow matching in the stationary stage for final image quality. Such a *hybrid* acceleration approach results in 1-NFE prior steps during all stages without the loss of accuracy (Section 4). Importantly, we maintain convergence with the stationary stage unmodified (see Section 3.3).

Corollary 1 specifies how our general framework can be reduced to PnP-DM with a specific choice of diffusion coefficient. See more implementation details of our method in Appendix C and analysis of convergence in Corollary 2. Corollary 3 establishes the validity of annealing acceleration through one-step sampling or other methods. Our theory-driven improvements are shown to be empirically effective in our benchmark experiments in Section 4.2. Our ablation studies in Section 4.4 confirm that each of our interventions on the SI-based prior step and hybrid one-step sampling is necessary.

Algorithm 1 Posterior sampling with FlowSGS.

Require: measurements $\mathbf{y}$, forward model $\mathcal{A}(\cdot)$, velocity network $\mathbf{v}_\theta(\cdot,\cdot)$ with interpolant schedule (α_t, σ_t), diffusion coefficient w_t, iteration count K, coupling parameters $\{\rho_k\}_{k=0}^{K-1}$, flag `accel`, and Meta Flow Map $\mathbf{X}_{s,u}(\cdot\,; t, \mathbf{x}_t)$ sharing the schedule (α_t, σ_t) (if `accel`)

1: $\mathbf{x}^0 \sim \mathcal{N}(0, \mathbf{I})$ ▷ Initialization.
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: $\quad \mathbf{z}^{k+1} \sim \pi(\mathbf{z}|\mathbf{x}^k) \propto \exp\left[-f(\mathbf{z};\mathbf{y}) - \frac{\|\mathbf{x}^k - \mathbf{z}\|^2}{2\rho_k^2}\right]$ ▷ **Proximal or Langevin dynamics.**
4: $\quad t_k \leftarrow \left(\frac{\sigma_t}{\alpha_t}\right)^{-1}(\rho_k)$ ▷ **Compute time t_k corresponding to noise level ρ_k.**
5: $\quad \mathbf{x}_{t_k} \leftarrow \alpha_{t_k}\, \mathbf{z}^{k+1}$ ▷ **Scale $\mathbf{z}^{k+1}$ to match noise level at t_k.**
6: $\quad$ **if** `accel` **and** $\rho_k > \rho_{\min}$ **then**
7: $\quad\quad \boldsymbol{\epsilon}^k \sim \mathcal{N}(0, \mathbf{I})$
8: $\quad\quad \mathbf{x}^{k+1} \leftarrow \mathbf{X}_{1-\varepsilon,\,\varepsilon}\left(\boldsymbol{\epsilon}^k\,;\, t_k,\, \mathbf{x}_{t_k}\right)$ ▷ **One-step prior step sampling with MFM.**
9: $\quad$ **else**
10: $\quad\quad \mathbf{x}^{k+1} \leftarrow$ Euler–Maruyama(SDE, $\mathbf{x}_{t_k}, t_k, \epsilon$) ▷ **Solve SDE with timestep correction.**
11: $\quad\quad\quad$ where the SDE is $\mathrm{d}\mathbf{x}_t = \left[\left(1 - \frac{w_t \alpha_t}{2\sigma_t \gamma_t}\right)\mathbf{v}(\mathbf{x}_t, t) + \frac{w_t \dot{\alpha}_t}{2\sigma_t \gamma_t}\,\mathbf{x}_t\right]\mathrm{d}t + \sqrt{w_t}\,\mathrm{d}\bar{\mathbf{w}}_t$.
12: $\quad$ **end if**
13: **end for**
14: **return** $\mathbf{x}^K$

Likelihood step Our approach to the likelihood step is also significant compared to existing flow-based methods. We are the first flow-based method to incorporate measurement likelihood via SGS.

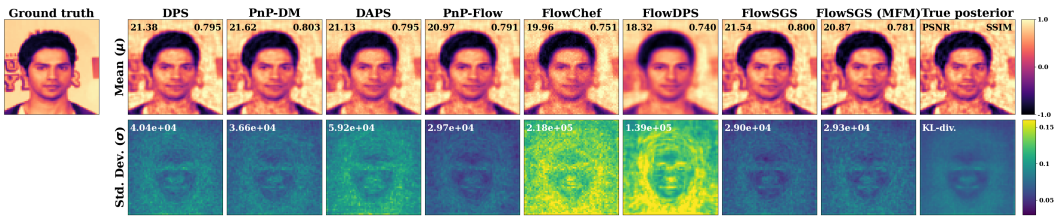

Figure 2: **Results on a toy compressed-sensing problem with an analytically available Gaussian posterior.** We show the mean image (row 1) and pixel-wise standard deviations (row 2) for the Gaussian posteriors. Our method estimates an accurate posterior with the lowest KL-divergence compared to the true posterior. See Appendix B for problem setup and Section 4 for method settings.

This change enables key applications in inverse imaging: more accurate posterior estimation (Section 4.1) and nonlinear inverse problems (Section 4.3) with bimodal posteriors. Many existing methods rely on simplifying approximations: some use linear projections between measurement and recovery space (Pokle et al., 2024; Zhang et al., 2024; Yan et al., 2025; Martin et al., 2025; Askari et al., 2025), while others incorporate the likelihood gradient as guidance (Kim et al., 2025) or use gradient-free vector-field steering for measurement consistency Patel et al. (2025). In contrast, our method is a *sampling* method that directly targets the actual Bayesian posterior. See more discussions on *guidance* and *sampling* methods in Wu et al. (2024). In incorporating the general likelihood step, we present the first SGS-based *sampling* method using flow priors.

3.1 Prior step with flow matching

The key observation that drives the design of FlowSGS's prior step is the fact that sampling from the SGS prior step Equation (7) is equivalent to running SI's reverse-time SDE Equation (3) at the specific starting point related to the coupling parameter ρ_k, which we state below in Theorem 1.

Theorem 1 (Prior step equivalence to SI reverse-time SDE)**.** *Suppose (α_t, σ_t) defines an interpolant process with Gaussian probability path in Equation* (2)*, and $\alpha_{t^*} > 0$ and t^* satisfies $\sigma_{t^*}/\alpha_{t^*} = \rho_k$, running the reverse-time SDE in Equation* (3) *from $\mathbf{x}_{t^*} = \alpha_{t^*}\mathbf{z}^{k+1}$ to $t = 0$ produces samples from the SGS prior step $\pi(\mathbf{x}|\mathbf{z}^{k+1})$ in Equation* (7)*, with the KL-optimal diffusion coefficient w_t^{KL}.*

See proof of Theorem 1 in Appendix A.2. While PnP-DM (Wu et al., 2024) shows a similar equivalence between its prior step and the diffusion's reverse-time SDE, our Theorem 1 generalizes to any Gaussian probability path defined by Equation (2) under the SI framework. In fact, we later show in Section 3.2 that PnP-DM's prior step is a special case of our prior step. The insights drawn from Theorem 1 directly lead to the timestep conversion, scaling, and reverse-SDE solving in Lines 4-11 of Algorithm 1, which we will show to be effective in our ablation studies.

3.2 Relation to Existing PnP Methods

While in this paper we assume a flow prior, another way of applying the score–velocity conversion in Equation (4) to the reverse-time SDE in Equation (3) is to replace the velocity field $\mathbf{v}(\cdot;\cdot)$ with the score function $\mathbf{s}(\cdot;\cdot)$, leading to a FlowSGS prior step using a diffusion model. We show in the following corollary that FlowSGS's diffusion-based prior step with a specific choice of the diffusion coefficient w_t results in the prior step of PnP-DM Wu et al. (2024). In other words, FlowSGS provides a general framework that can incorporate either a diffusion or flow prior, with arbitrary choices of using a score or velocity model, the interpolant process, and diffusion coefficients, and PnP-DM is a special case of FlowSGS under specific design choices.

Corollary 1 (PnP-DM as a special case of FlowSGS)**.** *Let $\lambda_t \triangleq -\gamma_t/\alpha_t = (\alpha_t\dot{\sigma}_t - \dot{\alpha}_t\sigma_t)/\alpha_t$. At the sampling stage, FlowSGS's timestep conversion and reverse-time SDE (3) reduce to PnP-DM (see (7) in Wu et al. (2024)) under specific conditions: the EDM diffusion coefficients $\alpha_t = s_{\mathrm{E}}(t)$, $\sigma_t = s_{\mathrm{E}}(t)\sigma_{\mathrm{E}}(t)$, the score–velocity conversion, and the KL-optimal diffusion coefficient $w_t^{\mathrm{KL}} = 2\lambda_t\sigma_t$.*

See proof of Corollary 1 in Appendix A.3. In fact, such a connection is no surprise, since the stochastic interpolant framework also includes diffusion models in its mathematical formulations. Our contribution lies in specifying the exact form of this relationship. FlowSGS's structural similarity to PnP-DM allows us to extend its convergence guarantee to FlowSGS under the SI parameterization, which we summarize in the following corollary.

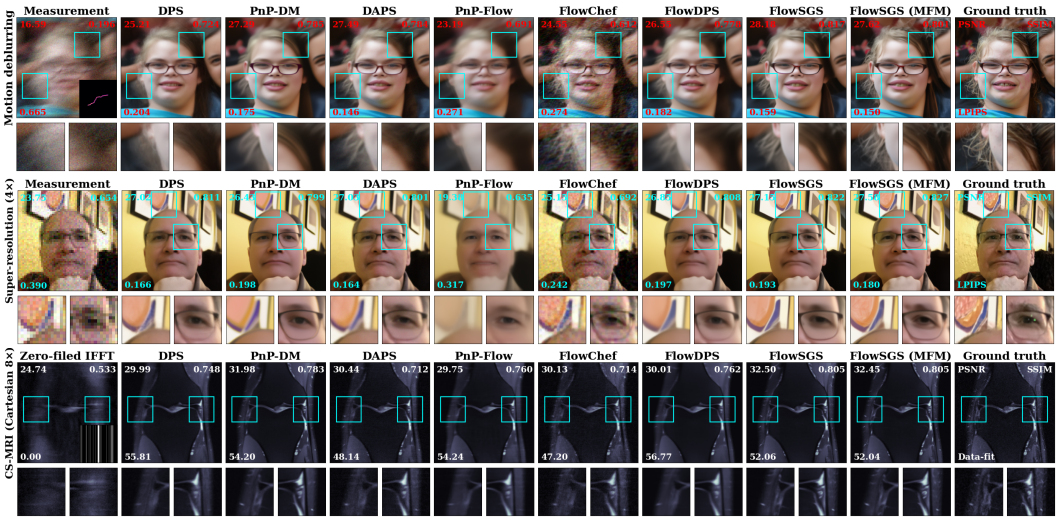

Figure 3: **Visual results on linear inverse problems.** We draw 8 posterior samples for each method and report the mean image. More results are shown in Appendix D.

Corollary 2 (Convergence analysis)**.** *Suppose the interpolant schedule (α_t, σ_t) satisfies that σ_t/α_t is strictly increasing on $(0, 1]$, and the diffusion coefficient w_t is bounded away from zero on the interval used by the prior step. Then, under the L^2-accurate condition on the pretrained flow network, Algorithm 1 with constant coupling $\rho_k \equiv \rho$ enjoys the same non-asymptotic convergence guarantee as PnP-DM (Wu et al., 2024, Theorem 3.1): the average Fisher information between the non-stationary process driven by the velocity network and the stationary process driven by the true velocity field decays at a rate of $O(1/K)$ in the number of SGS iterations K, up to a floor set by the velocity network approximation error. Writing μ_k for the law of the k-th iterate, π_ρ for the stationary distribution of the exact SGS chain at coupling ρ, $\delta = \inf \sqrt{w_t}$ over the prior-step interval, and ε_v for the L^2 velocity-approximation error, the bound reads*

$$\frac{1}{K}\sum_{k=1}^{K} \mathrm{FI}(\mu_k \,\|\, \pi_\rho) \;\le\; \frac{C\,\mathrm{KL}(\mu_0 \,\|\, \pi_\rho)}{K \min(\rho, \delta)^2} \;+\; E(\delta, \varepsilon_v), \tag{8}$$

where C is an absolute constant and $E(\delta, \varepsilon_v)$ is the velocity-approximation floor.

See detailed proof of Corollary 2 in Appendix A.4. Compared to PnP-DM, the assumptions become more flexible: PnP-DM inherits its diffusion-coefficient floor from the EDM forward process (which can vanish at the endpoints for iDDPM/EDM, requiring the workaround in (Wu et al., 2024, Appendix A.3)), whereas in FlowSGS, the diffusion coefficient w_t is a free sampling-time choice, so the floor condition is a constraint on the user's choice rather than on the model. The monotonicity condition on σ_t/α_t is the SI analog of PnP-DM's Assumption A.1 and is satisfied by all standard schedules we consider. Note that Corollary 2 describes the stationary stage when $\rho_k = \rho_{\min}$. We show in Corollary 3 that the warm-up stage does not affect the final convergence of the stationary chain, so that the varying ρ in warm-up does not break convergence. Note that the KL-optimal coefficient approaches 0 as $t \to 0$ for the linear interpolant ($w_t^{\mathrm{KL}} = 2t/(1-t)$) and GVP ($w_t^{\mathrm{KL}} = \pi \tan(\pi t/2)$), which violates the assumptions in Corollary 2. Following standard practice for diffusion-based samplers Zheng et al. (2025), we avoid this irregularity by truncating the interpolant at $t \in [\epsilon, 1-\epsilon]$ with $\epsilon = 1 \times 10^{-5}$ and terminating the reverse SDE at $t = \epsilon$ rather than $t = 0$, returning $\mathbf{x}^{k+1} = \mathbf{x}_\epsilon$. Additionally, for both schedules is not regular as $t \to 1$. To avoid this, we follow Ma et al. (2024) and use regularization $w_t^{\mathrm{KL},\eta} = w_t^{\mathrm{KL}} \sqrt{\frac{\mathcal{L}_t}{\mathcal{L}_t + 2\eta (w_t^{\mathrm{KL}})^2}}$, where $\mathcal{L}_t$ is the velocity matching loss at time t and η is fine-tuned for each task.

3.3 Prior step with hybrid one-step flow models

Our observation that the majority of the computation in FlowSGS is dominated by the warm-up annealing phase (see Appendix A.1) led to our hybrid prior-step acceleration by integrating one-step

flow models in the warm-up prior steps. We show that this does not affect convergence: any non-pathlogical prior step sampler may be used during warm-up, including one-step flow-map models, and the convergence rate and approximation floor of the subsequent stationary stage are unaffected.

Corollary 3 (Warm-up invariance)**.** *Let μ_w denote the law of the prior-step output at the end of the warm-up stage. If $\mathrm{KL}(\mu_w \,\|\, \pi_{\rho_{\min}}) < \infty$, then the convergence properties of Corollary 2 hold for the stationary stage, except that the bound in Equation* (8) *takes the new form*

$$\frac{1}{K_s}\sum_{k=1}^{K_s} \mathrm{FI}(\mu_k \,\|\, \pi_{\rho_{\min}}) \;\le\; \frac{C\,\mathrm{KL}(\mu_w \,\|\, \pi_{\rho_{\min}})}{K_s \min(\rho_{\min}, \delta)^2} \;+\; E(\delta, \varepsilon_v), \tag{9}$$

where K_s is the number of steps in the stationary stage.

See proof in Section A.5. Note that the warm-up affects only the bound, leaving the convergence rate and error floor unaffected, and that the assumption of non-infinite KL divergence is extremely loose. Consequently, we see in practice that the multi-step prior step SDE can be replaced by a cheaper one-step model during annealing without weakening performance.

4 Experiments

We compare our method with three diffusion-based samplers: DPS (Chung et al., 2023), PnP-DM (Wu et al., 2024), DAPS (Zhang et al., 2025), as well as three flow-based samplers: PnP-Flow (Martin et al., 2025), FlowChef (Patel et al., 2025), and FlowDPS (Kim et al., 2025). We pretrain diffusion/flow models using the same NCSN++ (Song et al., 2021) architecture for fair comparisons (see details in Appendix C.1). All flow-based methods use the linear interpolant schedule (Liu et al., 2023) with 32 discretization steps by default. In our hybrid prior-step acceleration, we use Meta Flow Maps (MFM, Potaptchik et al., 2026) in the prior steps (referred to as "FlowSGS (MFM)") during warm-up. We tune the hyperparameters of all methods for each task.

4.1 Toy problem

To validate that FlowSGS can accurately sample from the Bayesian posterior, we create a toy compressed sensing problem with a Gaussian likelihood and a Gaussian prior, such that the posterior is an analytically available Gaussian distribution. See problem setups in Appendix B.1. We draw 128 posterior samples and compute the mean and pixel-wise standard deviation of the Gaussian posterior, and compute the PSNR and SSIM of the mean image with respect to the ground truth image, the KL-divergence of the Gaussian posterior with respect to the true posterior, shown in Figure 2. We show that FlowSGS and PnP-DM (both being *sampling* methods) produce the most accurate posterior in terms of both mean image quality and KL-divergence. Note that PnP-Flow, although not claimed as a posterior sampler Martin et al. (2025), also produces an accurate posterior. In contrast, FlowChef, and FlowDPS significantly overestimate the uncertainty.

4.2 Linear inverse problems

We benchmark the baselines and FlowSGS (with different interpolant schedules) over a variety of linear inverse problems (motion/Gaussian deblurring, 4× super-resolution, and compressed sensing MRI) using the FFHQ dataset Karras et al. (2019) and the fastMRI knee dataset Zbontar et al. (2018) resized to 256×256. See problem setups in Appendix B.2. We pretrain diffusion/flow models on these datasets (checkpoints available). In Tables 1 and 5, we report metrics averaged over 100 test samples not used during training. We draw 8 samples for each method and report the mean image. Visual samples are shown in Figure 3. We show that FlowSGS stands out across a variety of inverse problems and is robust across different interpolant schedules. See more results in Appendix D.1.

4.3 Nonlinear Fourier phase retrieval

Fourier phase retrieval (FPR, Fienup & Wackerman, 1986; Fienup, 2012; Shechtman et al., 2015) is an ill-posed inverse problem that widely exists in real-world imaging systems, where a bimodal posterior could exist due to the loss of Fourier phase information. For the first time among flow-based samplers, we provide a benchmark on nonlinear FPR (see Table 1 and Figure 4). Since FPR is very ill-posed and relies on good initialization, we characterize image reconstruction quality by reporting the single best sample among 8 posterior samples we draw, similar to Chung et al. (2023); Wu et al. (2024) (see visual examples in Figure 10). For uncertainty quantification, we follow Li et al. (2026) and draw 512 samples from each method and classify the samples into modes using the PSNR with respect to each mode (samples with PSNR$<$15 dB for both modes are considered

Table 1: **Quantitative results averaged over 100 test samples.**

Method	Motion deblurring			Gaussian deblurring			Super-resolution		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DPS	26.76	0.777	0.184	28.54	0.810	0.151	28.63	0.819	0.153
PnP-DM	28.52	0.822	0.174	29.24	0.832	0.160	28.07	0.811	0.188
DAPS	28.85	0.813	0.155	29.52	0.822	0.146	28.96	0.815	0.154
PnP-Flow	25.42	0.761	0.233	28.34	0.814	0.181	21.92	0.684	0.292
FlowChef	25.18	0.595	0.299	28.31	0.796	0.166	26.91	0.709	0.231
FlowDPS	28.07	0.818	0.182	29.49	0.838	0.158	28.36	0.816	0.186
FlowSGS (VP)	28.93	0.840	0.147	29.58	0.844	0.139	28.50	0.824	0.157
FlowSGS (GVP)	29.01	0.841	0.149	29.41	0.843	0.144	28.59	0.829	0.161
FlowSGS (Linear)	**29.23**	**0.843**	0.148	**29.61**	**0.845**	0.144	**29.09**	0.840	0.159
FlowSGS (MFM)	28.85	0.832	**0.139**	29.45	0.839	**0.131**	29.06	**0.841**	**0.142**

Method	CS-MRI (Cartesian, 8×)			CS-MRI (radial, 8×)			Fourier phase retrieval		
	PSNR ↑	SSIM ↑	Data-fit ↓	PSNR ↑	SSIM ↑	Data-fit ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DPS	30.79	0.802	55.503	34.09	0.850	52.301	15.40	0.416	0.484
PnP-DM	32.43	0.835	53.693	34.77	0.865	50.808	32.13	0.901	0.104
DAPS	31.92	0.819	49.968	33.79	0.846	52.202	33.74	0.899	0.070
PnP-Flow	29.76	0.804	53.980	33.91	0.856	52.477	24.63	0.720	0.221
FlowChef	29.83	0.752	**47.151**	33.30	0.817	**48.896**	18.24	0.414	0.433
FlowDPS	29.09	0.783	56.554	34.02	0.854	54.141	27.55	0.808	0.183
FlowSGS (VP)	31.77	0.819	51.732	34.71	0.864	49.758	34.48	0.925	0.082
FlowSGS (GVP)	31.99	0.826	53.425	**34.90**	**0.869**	51.157	31.59	0.899	0.128
FlowSGS (Linear)	32.37	0.842	51.930	34.82	**0.869**	50.938	**36.35**	**0.939**	**0.059**
FlowSGS (MFM)	**32.57**	**0.844**	51.924	34.81	**0.869**	50.916	14.88	0.474	0.584

discarded). Using the best 25% of the samples, we show the mean image of each mode in Figure 4a, and the histogram of a pixel in the red channel in Figure 4b. While most methods reconstruct a bimodal posterior, we point out that our method reconstructs the best mean image for both modes (see metrics). FlowSGS, PnP-DM, and DAPS are sampling methods that produce clear bimodal distributions centered at the true pixel value (vertical dashed lines in the histogram), while DPS, FlowChef, and FlowDPS, which rely on guidance or trajectory-steering rather than direct posterior sampling, recover corrupted rotated modes. This demonstrates a strong advantage of principled posterior sampling over approximate methods in capturing multi-modal posteriors.

To handle this challenging ill-posed problem, we use 100 SDE steps in the prior step and Langevin dynamics instead of proximal sampling for the likelihood step, resulting in a significant slowdown

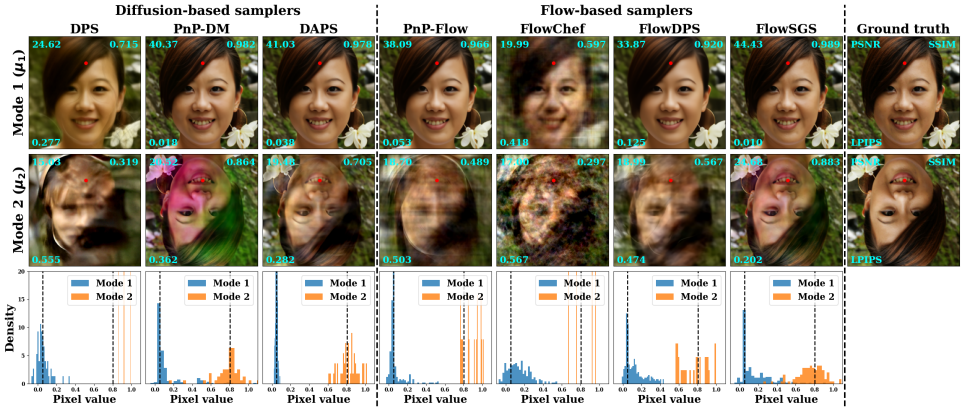

Figure 4: **Bimodal posterior for nonlinear Fourier phase retrieval.** With each method, we draw 512 samples and classify them into two modes. We compute the mean image using the top 25% images for each mode, and show the histograms for a pixel in the red channel (marked by red dots).

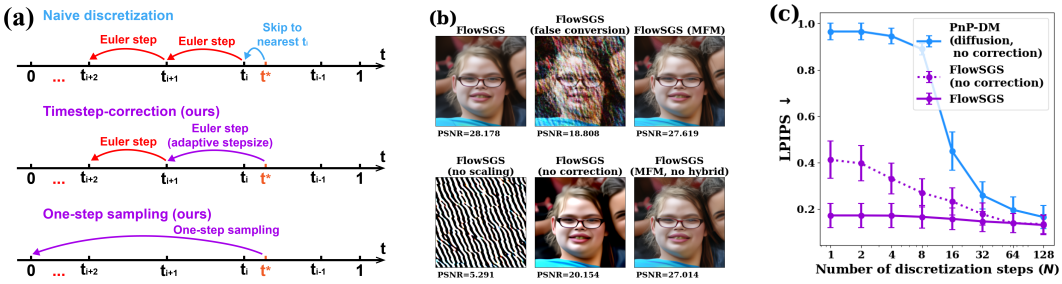

Figure 5: **Timestep-correction and one-step flow models enable efficient sampling.** (a) Illustration of FlowSGS's robust timestep correction to reduce discretization error. (b) Ablating our interventions decreases image quality on motion deblurring. See ground truth in Figure 3. (c) Removing the timestep correction (dashed purple) limits reconstruction performance for lower discretization step counts N. For small N, the timestep correction improvement (dashed purple to purple) is on the same order as the diffusion-to-flow improvement (blue to dashed purple). All experiments here are done with motion deblurring, averaged over 100 test samples (error bars show standard deviations).

(see Table 6). We also disable the hybrid prior-step acceleration, since the warm-up iterates of the non-convex FPR likelihood lie too far off the data manifold for a one-step flow map to recover, leading to corrupted image samples (see Table 1 and Figure 10).

4.4 Ablation Studies

Timestep conversion, scaling, and timestep correction In Figure 5b and Table 7, we show FlowSGS results on motion deblurring with missing components in Algorithm 1: false conversion (using the VP schedule for a linear flow model in Line 4), no scaling (removing Line 5), no correction (using naive discretization in the SDE solver). In Figure 5c (corresponding visual examples in Figure 11), we observe that the timestep correction is particularly important at low discretization step counts. This sensitivity comes from the low NFE count of flow-based methods, which stands in contrast to the large NFE count required by diffusion-based methods. As a result, PnP-DM (blue) provides a useful comparison: its naive discretization step and reliance on diffusion reveal the relative importance of switching to flow matching and correcting the time step.

Hybrid prior step acceleration We show in Figure 5 and Table 7 that using MFM for the hybrid prior step acceleration (Figure 5a) maintains image quality. Using MFM in both stages (no hybrid) slightly decreases performance, since MFM is much more challenging to train and leads to decreased image quality if used in the stationary phase; see Figure 5b for visual examples.

Interpolant schedules and diffusion coefficients In Tables 1 and 8, we compare FlowSGS results generated using three schedules and 5 diffusion coefficients (see Appendix A.1) on motion deblurring. We find that our method maintains performance under different settings.

5 Conclusion

We present modifications to SGS that accommodate SI-based generative models, which could even include yet-to-be-developed models. In doing so, we outperform existing flow-based methods by targeting the exact Bayesian posterior. We also compare to principled diffusion-based methods, demonstrating improvements beyond direct replacement of diffusion with flow priors. We provide theoretical grounding and SOTA empirical results. Current major limitations include a reduction in speed for nonlinear inverse problems, opening a significant direction for future work. Additionally, we rely on pixel space models which scale poorly with image size, and require tuning of a small set of hyperparameters ($\rho_0, \alpha, \rho_{\min}, \sigma$ for linear problems, with two additional Langevin parameters for nonlinear problems). Promising future directions include further exploration of annealing acceleration approaches and extending FlowSGS into semi-blind or blind samplers by incorporating our flow-based prior step into EM frameworks.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, provide feedback on research methodology and experiments, implement methods, and interpret results. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, propose or refine hypotheses, clean or reformat datasets, or support qualitative and thematic data analysis; assistance with translation is not applicable to this work. Additionally, we used generative AI tools to create and edit software code, edit a research paper to improve readability, draft parts of a research paper, and to identify relevant literature. We have reviewed all AI-assisted work. We reviewed all code and analysis generated by AI before running experiments and writing the paper. We have read papers suggested by AI before including them as citations. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Proofs of all theoretical results and the assumptions they require are given in Appendix A. Experiment setups are specified in Appendix B; network architectures, training settings, the proximal and Langevin likelihood samplers, and the annealing hyperparameters for every task are given in Appendix C. Source code and model checkpoints will be shared if accepted.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

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

Hossein Askari, Yadan Luo, Hongfu Sun, and Fred Roosta. Latent refinement via flow matching for training-free linear inverse problem solving. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=CF6qEI6rH0.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. In *International Conference on Machine Learning*, pp. 3462–3483. PMLR, 2024.

Charles A Bouman and Gregery T Buzzard. Generative plug and play: Posterior sampling for inverse problems. In *2023 59th annual Allerton conference on communication, control, and computing (Allerton)*, pp. 1–7. IEEE, 2023.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Stanley H Chan, Xiran Wang, and Omar A Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1): 84–98, 2016.

Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st international conference on image processing*, volume 2, pp. 168–172. IEEE, 1994.

Wenda Chu, Zihui Wu, Yifan Chen, Yang Song, and Yisong Yue. Split gibbs discrete diffusion posterior sampling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=U7ab14f4YQ.

Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated mri. *Medical image analysis*, 80:102479, 2022.

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

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

Florentin Coeurdoux, Nicolas Dobigeon, and Pierre Chainais. Plug-and-play split gibbs sampler: embedding deep generative priors in bayesian inference. *IEEE Transactions on Image Processing*, 33:3496–3507, 2024.

Elhadji C Faye, Mame Diarra Fall, and Nicolas Dobigeon. Regularization by denoising: Bayesian model and langevin-within-split gibbs sampling. *IEEE Transactions on Image Processing*, 34: 221–234, 2024.

James Fienup. Understanding the twin-image problem in phase retrieval. *Journal of the Optical Society of America A*, 2012.

James R Fienup and CC Wackerman. Phase-retrieval stagnation problems and solutions. *Journal of the Optical Society of America A*, 3(11):1897–1907, 1986.

Tomer Garber and Tom Tirer. Zero-shot image restoration using few-step guidance of consistency models (and beyond). In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2398–2407. IEEE, 2025.

Donald Geman and Chengda Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE transactions on Image Processing*, 4(7):932–946, 1995.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=uWj4s7rMnR.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30467–30476, 2026.

Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35:14715–14728, 2022.

Merve Gulle, Junno Yun, Yasar Utku Alçalar, and Mehmet Akçakaya. Pnp-cm: Consistency models as plug-and-play priors for inverse problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38464–38474, 2026.

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

Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. Robust compressed sensing mri with deep generative priors. *Advances in Neural Information Processing Systems*, 34:14938–14954, 2021.

Ulugbek S Kamilov, Charles A Bouman, Gregery T Buzzard, and Brendt Wohlberg. Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 40(1):85–97, 2023.

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*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022.

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*, pp. 12328–12337, 2025.

Peter E Kloeden and RA Pearson. The numerical solution of stochastic differential equations. *The ANZIAM Journal*, 20(1):8–12, 1977.

Tianao Li, Tjitske Starkenburg, Yu Sun, and Emma Alexander. Shuffleflow: Scalable posterior inference for bayesian inverse imaging. In *IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2026.

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://openreview.net/forum?id=PqvMRDCJT9t.

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://openreview.net/forum?id=XVjTT1nw5z.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=WUK8JIeetF.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1YO4EE3SPB.

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://openreview.net/forum?id=5AtHrq3B5R.

Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

Parikh Neal, Chu Eric, Peleato Borja, and Eckstein Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Maitreya Patel, Song Wen, Dimitris N Metaxas, and Yezhou Yang. Flowchef: Steering of rectified flow models for controlled generations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15308–15318, 2025.

Marcelo Pereyra, Luis A Vargas-Mieles, and Konstantinos C Zygalakis. The split gibbs sampler revisited: improvements to its algorithmic structure and augmented target distribution. *SIAM Journal on Imaging Sciences*, 16(4):2040–2071, 2023.

Ashwini Pokle, Matthew J. Muckley, Ricky T. Q. Chen, and Brian Karrer. Training-free linear image inverses via flows. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=PLIt3a4yTm.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael Samuel Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=K5gV8Yptne.

Yoav Shechtman, Yonina C Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev. Phase retrieval with application to optical imaging: a contemporary overview. *IEEE signal processing magazine*, 32(3):87–109, 2015.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vaRCHVj0uGI.

George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. *Advances in neural information processing systems*, 32, 2019.

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

Maxime Vono, Nicolas Dobigeon, and Pierre Chainais. Split-and-augmented gibbs sampler—application to large-scale inference problems. *IEEE Transactions on Signal Processing*, 67(6):1648–1661, 2019.

Zihui Wu, Yu Sun, Yifan Chen, Bingliang Zhang, Yisong Yue, and Katherine L Bouman. Principled probabilistic imaging using diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 37:118389–118427, 2024.

Xingyu Xu and Yuejie Chi. Provably robust score-based diffusion posterior sampling for plug-and-play image reconstruction. *Advances in Neural Information Processing Systems*, 37:36148–36184, 2024.

Yici Yan, Yichi Zhang, Xiangming Meng, and Zhizhen Zhao. FIG: Flow with interpolant guidance for linear inverse problems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fs2Z2z3GRx.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08839*, 2018.

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

Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2021.

Yasi Zhang, Peiyu Yu, Yaxuan Zhu, Yingshan Chang, Feng Gao, Ying N Wu, and Oscar Leong. Flow priors for linear inverse problems via iterative corrupted trajectory matching. *Advances in Neural Information Processing Systems*, 37:57389–57417, 2024.

Jiankun Zhao, Bowen Song, and Liyue Shen. Cosign: Few-step guidance of consistency model to solve general inverse problems. In *European Conference on Computer Vision*, pp. 108–126. Springer, 2024.

Hongkai Zheng, Wenda Chu, Bingliang Zhang, Zihui Wu, Austin Wang, Berthy Feng, Caifeng Zou, Yu Sun, Nikola Kovachki, Zachary Ross, et al. Inversebench: Benchmarking plug-and-play diffusion priors for inverse problems in physical sciences. In *International Conference on Learning Representations*, volume 2025, pp. 90912–90940, 2025.

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*, pp. 1219–1229, 2023.

A THEORY

A.1 INTERPOLANT SCHEDULES AND DIFFUSION COEFFICIENTS

In Table 2, we summarize three common interpolant schedules we experimented with FlowSGS, as well as their score–velocity conversion Equation (4) functions and time inversion functions. In Table 3, we summarize common choices of diffusion coefficients w_t. Note that w_t^{KL} is not regular at $t = 1$ for Linear and GVP interpolants because $\alpha_t \to 0$; see Ma et al. (2024) for details. We show the timestep conversion for each interpolant in Figure 6a.

Table 2: **Summary of interpolant schedules, score–velocity conversion (4), and time inversion.**

Property	**VP-SDE**	**Linear**	**GVP**
Signal scale	$\alpha_t = e^{-\frac{1}{2}\int_0^t \beta_s \mathrm{d}s}$	$\alpha_t = 1 - t$	$\alpha_t = \cos\left(\frac{\pi}{2}t\right)$
Noise scale	$\sigma_t = \sqrt{1-\alpha_t^2}$	$\sigma_t = t$	$\sigma_t = \sin\left(\frac{\pi}{2}t\right)$
Score-velocity conversion	$\mathbf{s} = -\frac{2}{\beta_t}\mathbf{v}_t - \mathbf{x}_t$	$\mathbf{s} = -\frac{(1-t)\mathbf{v}_t + \mathbf{x}_t}{t}$	$\mathbf{s} = -\frac{2}{\pi}\cot(\frac{\pi t}{2})\mathbf{v}_t - \mathbf{x}_t$
Time inversion	$t = B^{-1}\left(\log(1+\rho^2)\right)^*$	$t = \frac{\rho}{1+\rho}$	$t = \frac{2}{\pi}\arctan(\rho)$
Representative models	DDPM (Ho et al., 2020), Score-SDE (Song et al., 2021)	Flow Matching (Lipman et al., 2023), Rectified Flow (Liu et al., 2023)	SiT (Ma et al., 2024)

*For VP-SDE, $\beta_t > 0$ is the variance-preserving noise-rate schedule and $B(t) = \int_0^t \beta_s\, ds$ is its cumulative noise level. Therefore $B(t) = \log(1+\rho^2)$.

Table 3: **Diffusion coefficients w_t for the reverse-time SDE**, all chosen *post-training* without affecting $\mathbf{v}_\theta$. "Regular" means finite and non-negative at the endpoint.

	ODE	**Sigma**	**Sine**	**KL-optimal**	**Regularized KL-optimal**
Expression	0	σ_t	$\sin^2(\pi t)$	$2\left(\dot\sigma_t\sigma_t - \frac{\dot\alpha_t\sigma_t^2}{\alpha_t}\right)$	$w_t^{\mathrm{KL},\eta} = w_t^{\mathrm{KL}}\sqrt{\frac{\mathcal{L}_t}{\mathcal{L}_t + 2\eta(w_t^{\mathrm{KL}})^2}}$
Regular at $t=0$	✓	✓	✓	✓	✓
Regular at $t=1$	✓	✓	✓	×	✓

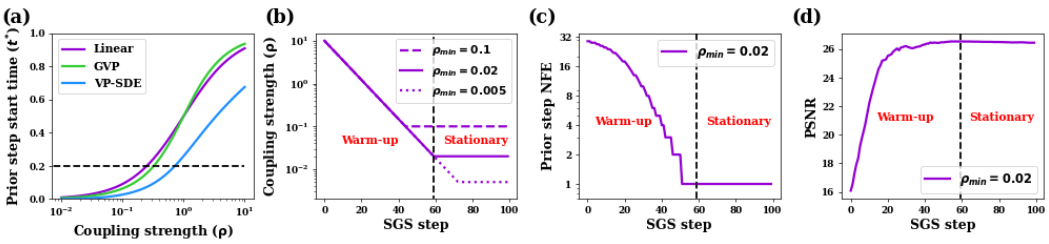

Figure 6: **Annealing of the coupling parameter ρ.** (a) Timestep conversion $t^*(\rho)$ for different interpolant schedules (see Table 2). (b) Annealing schedule for different $\rho_{\min}$. ρ decreases during the warm-up phase and is fixed at $\rho_{\min}$ at the stationary phase. (c) Prior step NFE as a function of SGS step k. Most of the NFE is dominated by the warm-up stage. (d) PSNR of prior step output $\mathbf{x}^k$ as a function of SGS step k on a motion deblurring test sample. The warm-up stage does not require high image quality and is therefore more robust to a one-step flow sampler with slightly degraded image quality. We label the warm-up and stationary stage for $\rho_{\min} = 0.02$ with dashed vertical lines in b,c,d.

A.2 Proof of Theorem 1

Proof of Theorem 1. The SGS prior step targets

$$\pi(\mathbf{x}_0|\mathbf{z}) \propto \exp[-g(\mathbf{x}_0)] \exp\left(-\frac{\|\mathbf{x}_0 - \mathbf{z}\|^2}{2\rho_k^2}\right), \tag{10}$$

which has the form of a Bayesian denoising posterior with prior $p(\mathbf{x}_0) \propto \exp[-g(\mathbf{x}_0)]$ and Gaussian likelihood $\pi(\mathbf{z}|\mathbf{x}_0) = \mathcal{N}(\mathbf{z}; \mathbf{x}_0, \rho_k^2\mathbf{I})$. This implies that $\mathbf{z}|\mathbf{x}_0 \sim \mathcal{N}(\mathbf{z}; \mathbf{x}_0, \rho_k^2\mathbf{I})$.

Suppose t^* satisfies $\sigma_{t^*}/\alpha_{t^*} = \rho_k$ and $\alpha_{t^*} > 0$. Solving reverse-time SDE in Equation (3) from t^* to 0 follows the conditional probability

$$\begin{aligned}
\pi(\mathbf{x}_0|\mathbf{x}_{t^*}) &\propto \pi(\mathbf{x}_{t^*}|\mathbf{x}_0)p(\mathbf{x}_0) \\
&= \mathcal{N}(\mathbf{x}_{t^*}; \alpha_{t^*}\mathbf{x}_0, \sigma_{t^*}^2\mathbf{I})p(\mathbf{x}_0) \\
&\propto \exp\left(-\frac{\|\mathbf{x}_{t^*} - \alpha_{t^*}\mathbf{x}_0\|^2}{2\sigma_{t^*}^2}\right) \exp[-g(\mathbf{x}_0)] \\
&= \exp\left(-\frac{\|\mathbf{x}_{t^*}/\alpha_{t^*} - \mathbf{x}_0\|^2}{2(\sigma_{t^*}/\alpha_{t^*})^2}\right) \exp[-g(\mathbf{x}_0)] \\
&= \exp\left(-\frac{\|\mathbf{x}_{t^*}/\alpha_{t^*} - \mathbf{x}_0\|^2}{2\rho_k^2}\right) \exp[-g(\mathbf{x}_0)],
\end{aligned} \tag{11}$$

where the first line uses Bayes' rule, and the third line uses the definition of the Gaussian probability path in Equation (2). Equation (11) indicates that the conditional probability $\pi(\mathbf{x}_0|\mathbf{x}_{t^*})$ is equal to the prior step $\pi(\mathbf{x}_0|\mathbf{z})$ when $\mathbf{x}_{t^*} = \alpha_{t^*}\mathbf{z}$. Therefore, running the SI's reverse-time SDE with initialization at t^* and $\mathbf{x}_{t^*} = \alpha_{t^*}\mathbf{z}$, the distribution we get at $\mathbf{x}_0$ is exact the distribution of the SGS prior step $\pi(\mathbf{x}_0|\mathbf{z})$. This provides the foundation of our prior step in Algorithm 1, where the time conversion (Line 4) follows

$$t^*(\rho) = \left(\frac{\sigma_{t^*}}{\alpha_{t^*}}\right)^{-1}(\rho), \tag{12}$$

the scaling (Line 5) of the output of the likelihood step $\mathbf{z}$ follows

$$\mathbf{x}_{t^*} = \alpha_{t^*}\mathbf{z}. \tag{13}$$

These two steps are essential preparation steps in service for the prior step for exact noise level matching. Note that with common choices of interpolants, $\left(\frac{\sigma_{t^*}}{\alpha_{t^*}}\right)$ is monotonic such that there always exists a unique solution $t^* \in [0, 1]$ for any coupling parameter $\rho > 0$.

□

A.3 Proof of Corollary 1

PnP-DM Wu et al. (2024) uses EDM diffusion Karras et al. (2022) to formulate the diffusion process:

$$\mathbf{x}_t = s_{\mathrm{E}}(t)\big(\mathbf{x}_0 + \sigma_{\mathrm{E}}(t)\boldsymbol{\epsilon}\big). \tag{14}$$

While under the stochastic-interpolant parameterization, we have

$$\mathbf{x}_t = \alpha_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}. \tag{15}$$

These two parameterizations have the relation

$$s_{\mathrm{E}}(t) = \alpha_t, \qquad \sigma_{\mathrm{E}}(t) = \frac{\sigma_t}{\alpha_t}. \tag{16}$$

Define

$$\gamma_t = \dot{\alpha}_t\sigma_t - \alpha_t\dot{\sigma}_t, \qquad \lambda_t = -\frac{\gamma_t}{\alpha_t} = \frac{\alpha_t\dot{\sigma}_t - \dot{\alpha}_t\sigma_t}{\alpha_t}. \tag{17}$$

Lemma 2. *Under the parameter correspondence in Equation* (16)*, we have*

$$\dot{\sigma}_{\mathrm{E}}(t) = \frac{\lambda_t}{\alpha_t}. \tag{18}$$

Proof. Since $\sigma_{\mathrm{E}}(t) = \sigma_t/\alpha_t$,

$$\dot{\sigma}_{\mathrm{E}}(t) = \frac{\dot{\sigma}_t\alpha_t - \sigma_t\dot{\alpha}_t}{\alpha_t^2} = -\frac{\dot{\alpha}_t\sigma_t - \alpha_t\dot{\sigma}_t}{\alpha_t^2} = -\frac{\gamma_t}{\alpha_t^2} = \frac{\lambda_t}{\alpha_t}. \tag{19}$$

□

Now, we can prove that the PnP-DM reverse-time SDE under the EDM parameterization Wu et al. (2024); Karras et al. (2022)

$$\mathrm{d}\mathbf{x}_t = \left[\frac{\dot{s_{\mathrm{E}}}(t)}{s_{\mathrm{E}}(t)}\mathbf{x}_t - 2s_{\mathrm{E}}(t)^2\dot{\sigma}_{\mathrm{E}}(t)\sigma_{\mathrm{E}}(t)\nabla_{\mathbf{x}_t}\log p\left(\frac{\mathbf{x}_t}{s_{\mathrm{E}}(t)};\sigma_{\mathrm{E}}(t)\right)\right]\mathrm{d}t + s_{\mathrm{E}}(t)\sqrt{2\dot{\sigma}_{\mathrm{E}}(t)\sigma_{\mathrm{E}}(t)}\,\mathrm{d}\mathbf{w}_t, \tag{20}$$

is a special case of the FlowSGS reverse-time SDE.

Proof of Corollary 1. First, by replacing the EDM diffusion coefficients using Equation (16), we have

$$\frac{\dot{s_{\mathrm{E}}}(t)}{s_{\mathrm{E}}(t)} = \frac{\dot{\alpha}_t}{\alpha_t}, \tag{21}$$

$$2s_{\mathrm{E}}(t)^2\dot{\sigma}_{\mathrm{E}}(t)\sigma_{\mathrm{E}}(t) = 2\alpha_t^2\left(\frac{\lambda_t}{\alpha_t}\right)\left(\frac{\sigma_t}{\alpha_t}\right) = 2\lambda_t\sigma_t, \tag{22}$$

$$s_{\mathrm{E}}(t)^2 2\dot{\sigma}_{\mathrm{E}}(t)\sigma_{\mathrm{E}}(t) = \alpha_t^2 2\left(\frac{\lambda_t}{\alpha_t}\right)\left(\frac{\sigma_t}{\alpha_t}\right) = 2\lambda_t\sigma_t. \tag{23}$$

This shows that the PnP-DM reverse-time SDE has drift coefficient $\dot{\alpha}_t/\alpha_t$, score coefficient $2\lambda_t\sigma_t$, and diffusion coefficient squared $2\lambda_t\sigma_t$.

The velocity-score relation can be written as

$$\mathbf{v}(\mathbf{x}_t, t) = \frac{\dot{\alpha}_t}{\alpha_t}\mathbf{x}_t - \lambda_t\sigma_t\mathbf{s}(\mathbf{x}_t, t). \tag{24}$$

Substituting this into the SI reverse-time SDE in Equation (3) gives

$$\begin{aligned}\mathrm{d}\mathbf{x}_t &= \left[\frac{\dot{\alpha}_t}{\alpha_t}\mathbf{x}_t - \lambda_t\sigma_t\mathbf{s}(\mathbf{x}_t, t) - \frac{w_t}{2}\mathbf{s}(\mathbf{x}_t, t)\right]\mathrm{d}t + \sqrt{w_t}\,\mathrm{d}\bar{\mathbf{w}}_t \\ &= \left[\frac{\dot{\alpha}_t}{\alpha_t}\mathbf{x}_t - \left(\lambda_t\sigma_t + \frac{w_t}{2}\right)\mathbf{s}(\mathbf{x}_t, t)\right]\mathrm{d}t + \sqrt{w_t}\,\mathrm{d}\bar{\mathbf{w}}_t.\end{aligned} \tag{25}$$

Choosing $w_t = 2\lambda_t\sigma_t$ yields

$$\mathrm{d}\mathbf{x}_t = \left[\frac{\dot{\alpha}_t}{\alpha_t}\mathbf{x}_t - 2\lambda_t\sigma_t\mathbf{s}(\mathbf{x}_t, t)\right]\mathrm{d}t + \sqrt{2\lambda_t\sigma_t}\,\mathrm{d}\bar{\mathbf{w}}_t. \tag{26}$$

This matches the PnP-DM's prior step reverse-time SDE. In FlowSGS, the initialization at t^* follows $\mathbf{x}_{t^*} = \alpha_{t^*}\mathbf{z}^{k+1}$, which directly translates to PnP-DM's initialization since $\alpha_{t^*} = s_{\mathrm{E}}(t^*)$. In summary, PnP-DM is recovered as the special case of FlowSGS with equivalent reverse-time SDE and initializations, when $w_t = 2\lambda_t\sigma_t$.

For VP-SDE, $\lambda_t\sigma_t = \beta_t/2$, so this diffusion coefficient aligns with and recovers the reverse process in VP-SDE.

$$w_t^{\mathrm{KL}} = 2\lambda_t\sigma_t = \beta_t. \tag{27}$$

□

Remark 1. *EDM fixes the reverse-SDE diffusion coefficient to the value implied by the forward diffusion process $w_t = s_{\mathrm{E}}(t)\sqrt{2\dot{\sigma}_{\mathrm{E}}(t)\sigma_{\mathrm{E}}(t)}$. In contrast, the SI framework allows any $w_t \geq 0$ at sampling time: $w_t = 0$ gives the probability-flow ODE, $w_t = 2\lambda_t\sigma_t$ recovers PnP-DM, and other choices can be tuned without retraining.*

A.4 Proof of Corollary 2

Rather than reproduce the full Fokker–Planck calculation, we show that FlowSGS satisfies all the structural conditions used in PnP-DM's Wu et al. (2024) Theorem 3.1, so that the same proof applies after a parameter substitution. We highlight three aspects and then describe the substitution.

Identical SGS skeleton. The likelihood step of FlowSGS (Algorithm 1, Line 3) is identical to PnP-DM's: both target the same conditional distribution $\pi(z|x) \propto \exp[-f(z;y) - \|x-z\|^2/2\rho^2]$ via proximal samping or Langevin dynamics. PnP-DM's Brownian-bridge interpretation ((Wu et al., 2024, Proposition A.2)) and the resulting KL/Fisher contraction (their Lemma 2 of Vempala & Wibisono (2019)) therefore carry over to FlowSGS.

Prior step has the canonical drift–diffusion form. PnP-DM's key technical lemma ((Wu et al., 2024, Lemma A.4)) is a generic Fokker–Planck argument that bounds the time-derivative of the KL divergence between any pair of diffusion processes that share the same diffusion coefficient $c(t)$ and differ only in their drifts. It makes no assumption about *how* the drift is constructed.

The FlowSGS prior-step SDE Equation (5) is exactly such a process: its drift is an affine combination of the velocity $\mathbf{v}(\mathbf{x}, t)$ and the state $\mathbf{x}$, and its diffusion coefficient is $\sqrt{w_t}$. The non-stationary process (driven by the pretrained $\mathbf{v}_\theta$) and the stationary process (driven by the true $\mathbf{v}$) share the same $c(t) = \sqrt{w_t}$, which is the only structural requirement of the lemma. Hence PnP-DM's lemma applies to FlowSGS's prior step verbatim, with the squared drift gap proportional to $\|\mathbf{v}_\theta - \mathbf{v}\|_2^2$ instead of $\|\mathbf{s}_\theta - \mathbf{s}\|_2^2$.

SI-equivalent assumptions. PnP-DM's Theorem 3.1 relies on two assumptions:

- *(Assumption A.1)* $\sigma_E(t)$ is strictly increasing, so there is a unique $t^\star$ with $\sigma_E(t^\star) = \rho$. Under the SI parameter correspondence $\sigma_E(t) = \sigma_t/\alpha_t$ (Corollary 1), this becomes the requirement that the noise-to-signal ratio σ_t/α_t is strictly increasing – precisely the well-posedness condition of our Theorem 1, and satisfied by all schedules in Table 2.
- *(Diffusion-coefficient floor)* The infimum $\delta := \inf_{[0,t^\star]} s_E(t)\sqrt{2\dot\sigma_E\sigma_E}$ of PnP-DM's reverse-SDE diffusion coefficient is strictly positive. Under the same correspondence this infimum becomes $\inf_{[0,t^\star]} \sqrt{w_t}$, where w_t is FlowSGS's sampling-time diffusion coefficient. This is where the assumptions become *looser*: PnP-DM's δ is fixed by the EDM forward process and can vanish at the endpoints (e.g. for VP/iDDPM), forcing the workaround in (Wu et al., 2024, Appendix A.3); FlowSGS lets the user pick w_t freely at sampling time, so the floor is a constraint on the user's choice rather than on the model.

The score-accuracy assumption of PnP-DM translates to an L^2-accuracy condition on the velocity field under π_τ, which is equivalent to the score condition via the velocity–score relation (Equation (4) in the main paper) up to bounded scalar factors.

Reduction. Combining the three remarks, every step of the proof of Theorem 3.1 in (Wu et al., 2024, Appendix A.2) goes through: the likelihood-step contraction is unchanged; the prior-step contraction follows from (Wu et al., 2024, Lemma A.4) applied to the canonical-form SDE described in Remark 2, with PnP-DM's $v(t)$ replaced by $\sqrt{w_t}$ and the score-error term replaced by the corresponding velocity-error term; the telescoping over K outer iterations is identical. The resulting non-asymptotic bound has the same shape as PnP-DM's: an $O(1/K)$ contraction from initialization governed by $\min(\rho, \delta)^2$, plus a velocity-approximation floor governed by δ^2. Collecting the constants from that argument gives the bound stated in Equation (8). □

A.5 Proof of Corollary 3

Write $\pi \equiv \pi_{\rho_{\min}}$ for the SGS target at coupling $\rho_{\min}$, and let P denote the Markov kernel of one iteration of Algorithm 1 in the stationary phase ($\rho_k \equiv \rho_{\min}$) with the reverse-time SDE prior step (Lines 9–11). The bound in Corollary 2 is obtained, through the reduction in Appendix A.4 to (Wu et al., 2024, Theorem 3.1), by telescoping a per-iteration KL descent inequality: for any law ν with $\mathrm{KL}(\nu \,\|\, \pi) < \infty$, one iteration of P decreases the KL divergence to π by a multiple of the Fisher information, up to an additive velocity-approximation error. The contraction constant is set

by $\min(\rho_{\min}, \delta)^2$ and the error term by the L^2-accuracy of $\mathbf{v}_\theta$ and the floor δ; both are properties of P alone, and the inequality holds for any input law with finite KL divergence to π, regardless of how that law was produced.

The warm-up stage is a Markov kernel Q (a composition of K_w iterations at couplings $\rho_k > \rho_{\min}$ with any prior-step sampler), so that $\mu_w = \mu_0 Q$ and the stationary-stage iterates have laws $\mu_w P^k$, $k = 1, \ldots, K_s$. Since $\mathrm{KL}(\mu_w \,\|\, \pi) < \infty$ by assumption, the descent inequality applies to each of the K_s stationary iterations starting from μ_w. Summing the KL terms telescope and using $\mathrm{KL} \geq 0$, one obtains the bound of Corollary 2 with K replaced by K_s and the initial KL divergence replaced by $\mathrm{KL}(\mu_w \,\|\, \pi)$; see Equation (9). The constants, the floor, and the approximation error are those of P and are therefore identical to Corollary 2. The warm-up kernel Q enters nowhere else. □

B Inverse problem setups

B.1 Toy problem

In our toy problem, we used images from the CelebA dataset Liu et al. (2015)[1]. All images are converted to grayscale, resized to 64×64, and normalized to [-1, 1] before any calculations. We generate a Random Gaussian matrix $\mathbf{A} \in \mathbb{R}^{1024 \times 4096}$ as the compressed sensing matrix. The forward model is $\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ and $\sigma = 0.01$. For the prior, we sample 50,000 male face images from the CelebA dataset and fit a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ to it. Along with the Gaussian likelihood, this gives us a Gaussian posterior

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \boldsymbol{\mu} = \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + \frac{1}{\sigma^2}\mathbf{A}^T\mathbf{y}\right), \quad \boldsymbol{\Sigma} = \left(\boldsymbol{\Sigma}_0^{-1} + \frac{1}{\sigma^2}\mathbf{A}^T\mathbf{A}\right)^{-1} \tag{28}$$

In Figure 7, we illustrate the measurement, prior, and analytical posterior used in the toy problem.

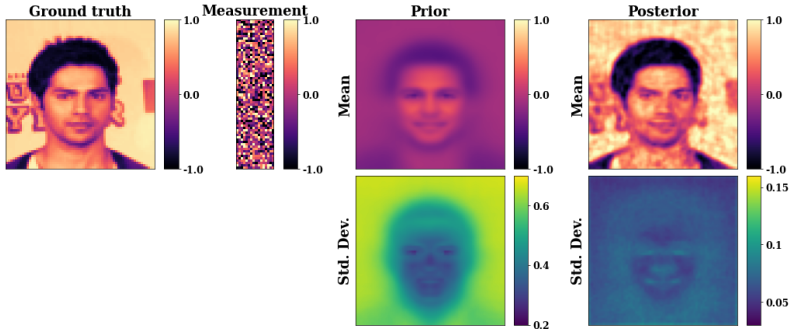

Figure 7: **Setup of the toy compressed sensing problem.**

B.2 Pixel-space model experiments

In our experiments with pixel-space models (Section 4.2), we use the FFHQ dataset Karras et al. (2019)[2] and fastMRI single-coil knee dataset Zbontar et al. (2018). All images are resized to 256×256 and normalized to [0, 1] as ground truths, which are then used to generate degraded measurements for different tasks.

Motion deblurring and Gaussian deblurring The forward model $\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon}$, where $\mathbf{H}$ is a convolution operator with a blur kernel and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ with $\sigma = 0.05$. For motion deblurring, we generate 64×64 motion blur kernels using this code[3] with trajectory intensity 0.5 and then pad them to 256×256. For Gaussian deblurring, we directly generate 256×256 blur kernels with standard deviation $\sigma_k = 3.0$ pixels. All kernels are normalized to have a sum of 1.0.

[1] License unknown.

[2] Creative Commons BY-NC-SA 4.0 license.

[3] `https://github.com/LeviBorodenko/motionblur` (license unknown).

Super-resolution The degradation process is modeled as $\mathbf{y} = \mathbf{S}\mathbf{x} + \boldsymbol{\epsilon}$, where $\mathbf{S}$ is a $4\times$ average pooling downsampling operator, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ with $\sigma = 0.05$.

Compressed sensing MRI The forward model is $\mathbf{y} = \mathbf{M}\mathcal{F}(\mathbf{x}) + \boldsymbol{\epsilon}$, where $\mathbf{M}$ is the masking operator, $\mathcal{F}(\cdot)$ is the 2D Fourier transform, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ with $\sigma = 0.02$. We generate both Cartesian and radial sampling masks with an acceleration rate of 8.0. The center 8% of the components are fully sampled with the Cartesian mask, while the other lines are randomly sampled. Lines in the radial mask are uniformly sampled.

Fourier phase retrieval The forward model is $\mathbf{y} = |\mathcal{F}(\mathbf{P}\mathbf{x})| + \boldsymbol{\epsilon}$, where $\mathbf{P}$ is the padding operator with an oversampling rate of 8.0 (padding the 256×256 images to 512×512), $\mathcal{F}(\cdot)$ is the 2D Fourier transform, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ with $\sigma = 0.01$. The oversampling operation is used to increase the number of measurements to reduce the ill-posedness of the problem Wu et al. (2024).

C Implementation details

C.1 Network architectures

In our toy problem and benchmarking experiments with pixel-space models (Sections 4.1 and 4.2), we pretrain our own diffusion and flow models using the NCSN++ architecture (Song et al., 2021) implemented by InverseBench (Zheng et al., 2025). For Meta Flow Maps, we add additional zero-initialized conditional embedding layers to NCSN++ and initialize the weights from the velocity checkpoint before training. All models are trained for 500 epochs with the Adam optimizer on each dataset and use an exponential moving average (EMA) for better performance. For the diffusion models, we use the VP-SDE as the diffusion schedule, while for the flow models, we train with the VP-SDE, linear, and GVP schedules (see summary in Table 2) and compare them in our ablation studies (see Table 1). Within every task, all diffusion/flow models are trained on the same training set and share the same architecture and optimization parameters for fair comparisons. Our model checkpoints trained on CelebA, FFHQ, and fastMRI knee are available on our project page.

C.2 Likelihood step proximal sampling

We assume a Gaussian noise model $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, and the likelihood step in Equation (6) follows the form:

$$\mathbf{z} \sim \pi(\mathbf{z}|\mathbf{x}^k, \mathbf{y}) \propto \exp\left[-\frac{\|\mathcal{A}(\mathbf{z}) - \mathbf{y}\|^2}{2\sigma^2} - \frac{\|\mathbf{z} - \mathbf{x}^k\|^2}{2\rho_k^2}\right], \tag{29}$$

where σ represents the standard deviation for the noise model. When the forward model is linear ($\mathcal{A}(\mathbf{x}) = \mathbf{A}\mathbf{x}$), there exists an analytical proximal sampling solution to the likelihood step (Wu et al., 2024):

$$\mathbf{z} \sim \mathcal{N}(\mathbf{m}(\mathbf{x}^k), \Lambda^{-1}) \tag{30}$$

where $\Lambda := \mathbf{A}^\top\Sigma^{-1}\mathbf{A} + \frac{1}{\rho_k^2}\mathbf{I}$ and $\mathbf{m}(\mathbf{x}) := \Lambda^{-1}\left(A^\top\Sigma^{-1}\mathbf{y} + \frac{1}{\rho_k^2}\mathbf{x}\right)$. We use such proximal sampling for all our linear inverse problems. For each task, we tune σ for the best reconstruction quality.

C.3 Likelihood step with Langevin dynamics

When the forward model $\mathcal{A}(\cdot)$ is nonlinear (e.g., Fourier phase retrieval), the proximal generator does not exist. In this case, we sample from Equation (29) using Langevin dynamics (Uhlenbeck & Ornstein, 1930), which leads to an increase in sampling time. We use the implementation of Langevin dynamics from InverseBench (Zheng et al., 2025). For each Langevin step, we compute the gradient:

$$\nabla E(\mathbf{z}) = \frac{1}{2\sigma^2}\nabla_{\mathbf{z}}\|\mathcal{A}(\mathbf{z}) - \mathbf{y}\|^2 + \frac{\mathbf{z} - \mathbf{x}^k}{\rho_k^2}, \tag{31}$$

and update $\mathbf{z}$ via:

$$\mathbf{z}^{(\ell+1)} = \mathbf{z}^{(\ell)} - \eta\,\nabla E\left(\mathbf{z}^{(\ell)}\right) + \sqrt{2\eta}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{32}$$

where η denotes the Langevin step size. We use 100 steps in the Langevin sampling in our Fourier phase retrieval experiments.

C.4 ANNEALING SCHEDULE

We use 100 iterations in SGS and summarize the annealing hyperparameters used in each task for FlowSGS in Table 4. We discovered that the sampling quality is most sensitive to $\rho_{\min}$, which determines the stationary coupling of the likelihood and prior step, as well as the starting point of the ending prior step that refines the final outputs. A reasonably large starting level ρ_0 and an annealing rate $\alpha > 0.9$ lead to similar performance.

Table 4: **Hyperparameters of the annealing schedule for different tasks.**

Inverse problem	Starting level (ρ_0)	Minimum level ($\rho_{\min}$)	Decay rate (α)
Toy problem	10	0.05	0.9
Motion deblurring	10	0.05	0.9
Gaussian deblurring	10	0.05	0.9
Super-resolution ($4\times$)	10	0.05	0.9
CS-MRI (Cartesian, $8\times$)	10	0.05	0.9
CS-MRI (radial, $8\times$)	10	0.02	0.9
Fourier phase retrieval	10	0.01	0.9

D ADDITIONAL RESULTS

D.1 MAIN EXPERIMENTS

In this section, we provide additional results in our main experiments using pixel-space models in Sections 4.2 and 4.3. In Table 5, we show the standard deviation of the mean for each metric provided in Table 1. We also show the runtime for each method when drawing 8 posterior samples in a batch in Table 6. All timing is averaged over 100 runs on a single NVIDIA H100 GPU. Note that FlowSGS's runtime differs across different interpolant schedules, because different schedules convert to different starting time points for the prior step SDE, leading to different NFEs (881 for linear, 828 for GVP, and 481 for VP-SDE). With all considered interpolant schedules, FlowSGS is faster than existing diffusion-based samplers with proximal likelihood step. In Figures 8 to 10, we show additional visual samples for the 6 inverse problems in our main experiments in Sections 4.2 and 4.3.

D.2 ABLATION STUDY

Key components in FlowSGS In Table 7, we test the effectiveness of the key components in FlowSGS by removing them individually. We show that timestep conversion, scaling, and timestep correction all lead to improved accuracy. The hybrid prior step acceleration leads to slightly better image quality compared to using MFM in both stages. Results are averaged over 100 test samples on the motion deblurring task, with errors (standard deviation of the mean) provided.

Interpolant schedules and diffusion coefficients In Tables 1 and 8, we provide a quantitative comparison of different interpolant coefficients and diffusion coefficients on motion deblurring. Results are averaged over all 100 test images, with errors (standard deviation of the mean) provided. We observe that our method is robust across different interpolant schedules and diffusion coefficients.

Number of SDE discretization steps In Figure 11, we show visual samples from the ablation study on the SDE discretization steps in Figure 5c for motion deblurring. We see that FlowSGS with timestep correction is more robust under a small discretization step setting, while PnP-DM drastically fails due to high discretization error from the diffusion model.



Figure 8: **Additional visual samples on motion deblurring, Gaussian deblurring, and 4× super-resolution.** We draw 8 posterior samples for each method and show the mean image.

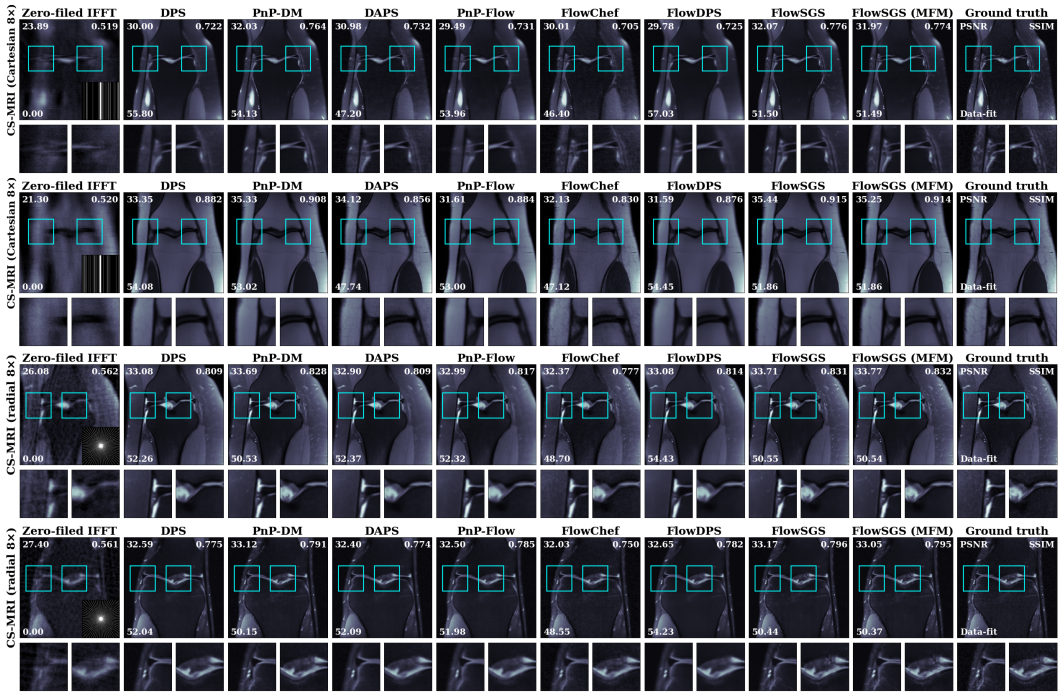

Figure 9: **Additional visual samples on compressed sensing MRI with Cartesian masks and radial masks.** We draw 8 posterior samples for each method and show the mean image.

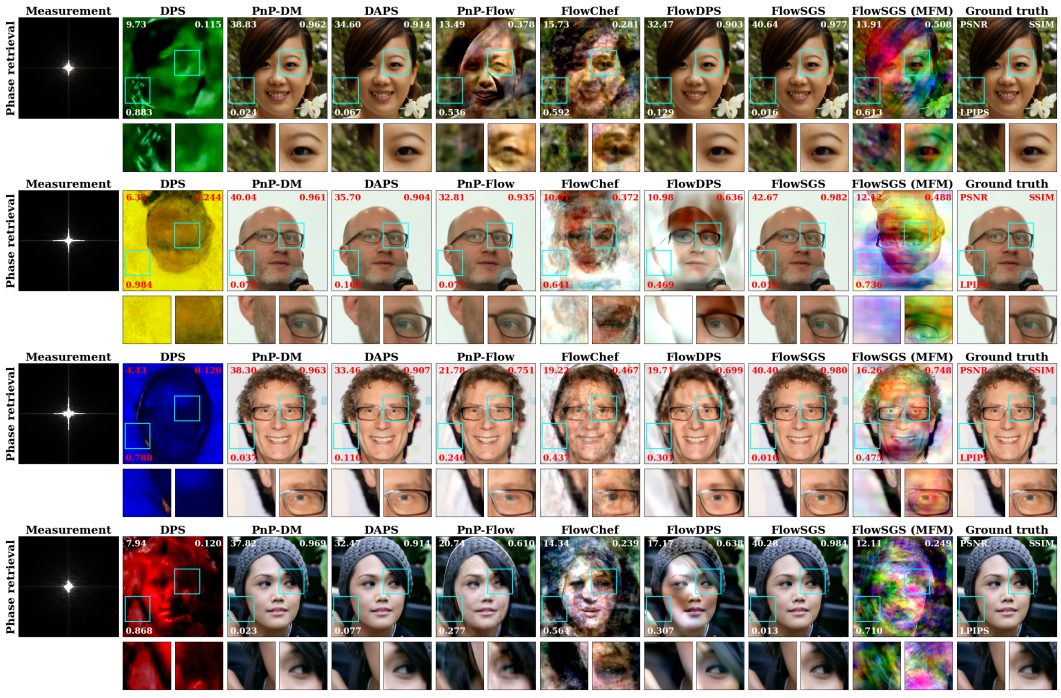

Figure 10: **Additional visual samples on Fourier phase retrieval.** We draw 8 posterior samples from each method and report the best one. Note that FlowSGS (MFM) results in degraded samples, as discussed in Section 4.3.

Table 5: **Quantitative results averaged over 100 test images with errors.** On each of 100 test images, we take the mean across 8 samples for each method and score the mean reconstruction. We report the mean ± standard deviation of the mean image score for several metrics. For Fourier phase retrieval, performance is computed on the best of each batch of 8 samples rather than their mean.

Method	Motion deblurring			Gaussian deblurring		
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
DPS	26.76 ± 0.24	0.777 ± 0.007	0.184 ± 0.005	28.54 ± 0.22	0.810 ± 0.006	0.151 ± 0.004
PnP-DM	28.52 ± 0.21	0.822 ± 0.005	0.174 ± 0.005	29.24 ± 0.21	0.832 ± 0.005	0.160 ± 0.004
DAPS	28.85 ± 0.21	0.813 ± 0.005	0.155 ± 0.005	29.52 ± 0.21	0.822 ± 0.004	0.146 ± 0.004
PnP-Flow	25.42 ± 0.30	0.761 ± 0.008	0.233 ± 0.007	28.34 ± 0.23	0.814 ± 0.006	0.181 ± 0.005
FlowChef	25.18 ± 0.14	0.595 ± 0.004	0.299 ± 0.006	28.31 ± 0.21	0.796 ± 0.005	0.166 ± 0.004
FlowDPS	28.07 ± 0.25	0.818 ± 0.006	0.182 ± 0.006	29.49 ± 0.23	0.838 ± 0.005	0.158 ± 0.004
FlowSGS (VP)	28.93 ± 0.22	0.840 ± 0.005	0.147 ± 0.004	29.58 ± 0.22	0.844 ± 0.005	0.139 ± 0.004
FlowSGS (GVP)	29.01 ± 0.21	0.841 ± 0.005	0.149 ± 0.004	29.41 ± 0.22	0.843 ± 0.005	0.144 ± 0.004
FlowSGS (Linear)	**29.23 ± 0.20**	**0.843 ± 0.005**	0.148 ± 0.004	**29.61 ± 0.22**	**0.845 ± 0.005**	0.144 ± 0.004
FlowSGS (MFM)	28.85 ± 0.21	0.832 ± 0.005	**0.139 ± 0.004**	29.45 ± 0.22	0.839 ± 0.005	**0.131 ± 0.004**

Method	CS-MRI (Cartesian, 8×)			CS-MRI (radial, 8×)		
	PSNR (↑)	SSIM (↑)	Data-fit (↓)	PSNR (↑)	SSIM (↑)	Data-fit (↓)
DPS	30.79 ± 0.18	0.802 ± 0.007	55.503 ± 0.127	34.09 ± 0.20	0.850 ± 0.006	52.301 ± 0.028
PnP-DM	32.43 ± 0.24	0.835 ± 0.007	53.693 ± 0.065	34.77 ± 0.22	0.865 ± 0.007	50.808 ± 0.032
DAPS	31.92 ± 0.19	0.819 ± 0.006	49.968 ± 0.030	33.79 ± 0.20	0.846 ± 0.007	52.202 ± 0.068
PnP-Flow	29.76 ± 0.23	0.804 ± 0.007	53.980 ± 0.063	33.91 ± 0.21	0.856 ± 0.006	52.477 ± 0.036
FlowChef	29.83 ± 0.23	0.752 ± 0.006	**47.151 ± 0.027**	33.30 ± 0.19	0.817 ± 0.006	**48.896 ± 0.022**
FlowDPS	29.09 ± 0.33	0.783 ± 0.008	56.554 ± 0.162	34.02 ± 0.20	0.854 ± 0.007	54.141 ± 0.079
FlowSGS (VP)	31.77 ± 0.26	0.819 ± 0.006	51.732 ± 0.026	34.71 ± 0.21	0.864 ± 0.006	49.758 ± 0.026
FlowSGS (GVP)	31.99 ± 0.26	0.826 ± 0.007	53.425 ± 0.039	**34.90 ± 0.22**	**0.869 ± 0.006**	51.157 ± 0.033
FlowSGS (Linear)	32.37 ± 0.25	0.842 ± 0.007	51.930 ± 0.031	34.82 ± 0.22	**0.869 ± 0.006**	50.938 ± 0.034
FlowSGS (MFM)	**32.57 ± 0.24**	**0.844 ± 0.007**	51.924 ± 0.030	34.81 ± 0.22	**0.869 ± 0.006**	50.916 ± 0.034

Method	Super-resolution (4×)			Fourier phase retrieval		
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
DPS	28.63 ± 0.20	0.819 ± 0.005	0.153 ± 0.004	15.40 ± 0.56	0.416 ± 0.018	0.484 ± 0.018
PnP-DM	28.07 ± 0.18	0.811 ± 0.005	0.188 ± 0.005	32.13 ± 0.99	0.901 ± 0.012	0.104 ± 0.013
DAPS	28.96 ± 0.18	0.815 ± 0.004	0.154 ± 0.004	33.74 ± 0.24	0.899 ± 0.005	0.070 ± 0.003
PnP-Flow	21.92 ± 0.16	0.684 ± 0.008	0.292 ± 0.006	24.63 ± 0.61	0.720 ± 0.018	0.221 ± 0.014
FlowChef	26.91 ± 0.12	0.709 ± 0.003	0.231 ± 0.005	18.24 ± 0.35	0.414 ± 0.014	0.433 ± 0.010
FlowDPS	28.36 ± 0.19	0.816 ± 0.005	0.186 ± 0.005	27.55 ± 0.63	0.808 ± 0.015	0.183 ± 0.011
FlowSGS (VP)	28.50 ± 0.16	0.824 ± 0.005	0.157 ± 0.004	34.48 ± 0.90	0.925 ± 0.008	0.082 ± 0.012
FlowSGS (GVP)	28.59 ± 0.16	0.829 ± 0.005	0.161 ± 0.004	31.59 ± 1.11	0.899 ± 0.014	0.128 ± 0.016
FlowSGS (Linear)	**29.09 ± 0.19**	0.840 ± 0.005	0.159 ± 0.004	**36.35 ± 0.90**	**0.939 ± 0.010**	**0.059 ± 0.011**
FlowSGS (MFM)	29.06 ± 0.19	**0.841 ± 0.004**	**0.142 ± 0.004**	14.88 ± 0.26	0.474 ± 0.017	0.584 ± 0.011

Table 6: **Average sampling runtimes when drawing 8 posterior samples in parallel.** All runtimes are measured on a single NVIDIA H100 GPU. We report the mean ± standard deviation computed from 100 runs.

Method	Runtime (s)
DPS	103.3 ± 0.0
PnP-DM (w/ proximal sampling)	106.3 ± 0.0
PnP-DM (w/ Langevin dynamics)	123.1 ± 0.0
DAPS	64.2 ± 0.0
PnP-Flow	5.4 ± 0.0
FlowChef	5.4 ± 0.0 ± 0.0
FlowDPS	5.5± 0.0
FlowSGS (Linear, w/ proximal sampling)	52.9 ± 0.0
FlowSGS (Linear, w/ Langevin dynamics)	144.2 ± 0.0
FlowSGS (GVP, w/ proximal sampling)	41.5 ± 0.0
FlowSGS (VP, w/ proximal sampling)	25.9 ± 0.0
FlowSGS (MFM, w/ proximal sampling)	5.5 ± 0.0

Table 7: **Ablation study on key components in the prior step of FlowSGS for motion deblurring**, averaged over 100 test samples in our test set. We draw 8 posterior samples for each method and report the metric on the mean image.

Method	Motion deblurring		
	PSNR ↑	SSIM ↑	LPIPS ↓
FlowSGS (false conversion)	23.69 ± 0.28	0.491 ± 0.014	0.311 ± 0.022
FlowSGS (no scaling)	7.04 ± 0.18	0.129 ± 0.017	0.694 ± 0.018
FlowSGS (no timestep correction)	19.40 ± 0.14	0.608 ± 0.006	0.233 ± 0.006
FlowSGS	**29.23 ± 0.20**	**0.843 ± 0.005**	0.148 ± 0.004
FlowSGS (MFM, no hybrid acceleration)	28.03 ± 0.19	0.813 ± 0.005	0.143 ± 0.004
FlowSGS (MFM)	28.85 ± 0.21	0.832 ± 0.005	**0.139 ± 0.004**



Figure 11: **Visual samples of the ablation study on the SDE discretization steps for motion deblurring.** We draw 8 posterior samples for each method and show the mean image.

Table 8: **Ablation results on diffusion coefficients** w_t, averaged over 100 test images on motion deblurring. We report the mean $\pm$ standard deviation of the mean image score for several metrics. We take the mean across 8 samples for each method and score the mean reconstruction.

Diffusion Coefficient	**Motion deblurring**		
	PSNR ↑	SSIM ↑	LPIPS ↓
$w_t = 0$ (ODE)	29.22 ± 0.20	0.844 ± 0.005	$\mathbf{0.148 \pm 0.004}$
$w_t = \sigma_t$	$\underline{29.32 \pm 0.20}$	$\mathbf{0.845 \pm 0.005}$	0.150 ± 0.004
$w_t = \sin^2(\pi t)$	$\mathbf{29.36 \pm 0.20}$	$\underline{0.844 \pm 0.005}$	0.153 ± 0.005
$w_t = w_t^{\mathrm{KL}}$	29.21 ± 0.21	0.839 ± 0.005	0.161 ± 0.005
$w_t = w_t^{\mathrm{KL},\eta}$	29.23 ± 0.20	0.843 ± 0.005	$\underline{0.148 \pm 0.004}$