Consistency Posterior Sampling for Diverse Image Synthesis

Vishal Purohit[1,*], Matthew Repasky[2,*], Jianfeng Lu[3,4], Qiang Qiu[1],
Yao Xie[2], and Xiuyuan Cheng[3,‡]

[1]Elmore Family School of Electrical and Computer Engineering, Purdue University, USA
[2]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, USA
[3]Department of Mathematics, Duke University, USA
[4]Department of Physics and Department of Chemistry, Duke University, USA

Abstract

Posterior sampling in high-dimensional spaces using generative models holds significant promise for various applications, including but not limited to inverse problems and guided generation tasks. Generating diverse posterior samples remains expensive, as existing methods require restarting the entire generative process for each new sample. In this work, we propose a posterior sampling approach that simulates Langevin dynamics in the noise space of a pre-trained generative model. By exploiting the mapping between the noise and data spaces which can be provided by distilled flows or consistency models, our method enables seamless exploration of the posterior without the need to re-run the full sampling chain, drastically reducing computational overhead. Theoretically, we prove a guarantee for the proposed noise-space Langevin dynamics to approximate the posterior, assuming that the generative model sufficiently approximates the prior distribution. Our framework is experimentally validated on image restoration tasks involving noisy linear and nonlinear forward operators applied to LSUN-Bedroom (256 $\times$ 256) and ImageNet (64 $\times$ 64) datasets. The results demonstrate that our approach generates high-fidelity samples with enhanced semantic diversity even under a limited number of function evaluations, offering superior efficiency and performance compared to existing diffusion-based posterior sampling techniques.

1. Introduction

Generative models that approximate complex data priors have been widely used for guided generation [12, 14]. While early approaches relied on GANs [4, 18, 27, 28, 39, 43], diffusion models have since outperformed them, becoming the state-of-the-art for conditional generation [9, 14, 22, 23, 47]. Posterior sampling methods have gained traction for inverse problems, where the goal is to sample from $p(x \mid y) \propto p(y \mid x)p(x)$[12, 30, 31]. Although these posteriors are often intractable, generative models enable efficient approximations. Earlier diffusion-based solutions required task-specific training[34, 38, 44–46], while recent works use pre-trained diffusion priors in a training-free manner [10, 11, 30, 31, 53], with extensions to nonlinear tasks [12, 21, 48, 49].

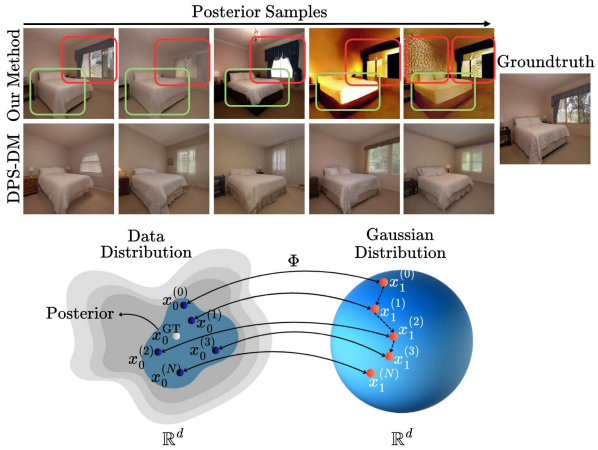

Figure 1. **(Top)**: Posterior samples generated by our method and DPS-DM [12]. Our approach exhibits higher perceptual diversity, capturing variations in high-level features such as lighting, window style, and wall patterns. Red boxes highlight uncertain semantic features, while green boxes show persistent properties. **(Bottom) :** A schematic representation of posterior sampling via Langevin dynamics in our proposed framework. The sampling process begins with an initial sample $x_1^{(0)}$ from the noise space and maps to data space as $x_0^{(0)}$ using a deterministic mapper Φ and progressively updates the noise space input to obtain diverse posterior samples.

Inverse problem solvers are typically grouped into *point estimate* or *multiple estimate* approaches. Most recent methods focus on the former [12, 21, 48, 49] and face chal-

*The first two authors contributed equally and are listed alphabetically.
†Email: xiuyuan.cheng@duke.edu
‡Code: https://github.com/Vishal-S-P/CPS_Diverse

lenges in generating diverse samples efficiently. For example, DPS [12] requires full denoising for each sample, increasing computational cost. While EBMs offer progressive sampling via MCMC [56, 57] and recent works explore HMC in generative noise spaces [25, 40, 55], such techniques are largely unexplored for inverse problems. We bridge this gap by introducing a measurement-guided posterior sampler in noise space.

We propose a posterior sampling method that performs exploration directly in the noise space of a pre-trained generative model. By leveraging measurements from the inverse problem to initialize the noise space, our approach enables targeted and efficient exploration. We employ Langevin dynamics in noise space, taking advantage of the deterministic, one-to-one mapping between noise and data provided by models like consistency models [51]. This eliminates the need to approximate the measurement likelihood and allows us to derive a theoretical bound on the approximation error of our posterior samples. Sampling in noise space enables progressive accumulation of diverse reconstructions without repeated full denoising runs. As demonstrated in Figure 1, our method yields high-quality, diverse solutions. Furthermore, as shown in Figure A.1, unlike DPS, whose runtime scales poorly with the number of posterior samples, our method incurs only a negligible increase in reconstruction time, highlighting its computational efficiency. The key contributions of this work are summarized as follows:

- We present a posterior sampling method defined by Langevin dynamics in the noise space of a pre-trained generative model, enabling the accumulation of posterior samples.
- We provide a theoretical guarantee on the posterior sampling approximation error, which is bounded by the approximation error of the prior by the pre-trained generative model.

Notation. We use $\propto$ to stand for the expression of a probability density up to a normalizing constant to enforce integral one, e.g. $p(x) \propto F(x)$ means that $p(x) = F(x)/Z$ where $Z = \int F(x)dx$. For a mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ and a distribution P, $T_{\#}P$ stands for the push-forwarded distribution, that is $T_{\#}P(A) = P(T^{-1}A)$ for any measurable set A. When both P and $T_{\#}P$ has density, $dP = pdx$, we also use $T_{\#}p$ to denote the density of $T_{\#}P$.

2. Background

Diffusion models. Sampling from diffusion models (DMs) is performed by simulating the reverse process of a forward-time noising stochastic differential equation (SDE) $dx_t = \mu(x_t, t)dt + \beta(t)dW_t$ [50], where W_t denotes Brownian motion, and $t \in [0, 1]$. This forward SDE transforms data from p_{data} into a Gaussian distribution γ. The marginal densities p_t are shared with the *probability flow ODE* (PF-ODE):

$$dx_t = \left[\mu(x_t, t) - \frac{1}{2}\beta(t)^2 \nabla \log p_t(x_t)\right] dt. \tag{1}$$

Score-based generative models use neural networks to approximate $\nabla \log p_t(x_t)$, enabling reverse-time integration of (1) using numerical techniques [29, 47].

Deterministic diffusion solvers. Unlike stochastic samplers [23, 50], deterministic solvers simulate the PF-ODE (1). DDIM [47] introduces an implicit, deterministic mapping from noise to data, while higher-order solvers [29] further reduce function evaluations needed for quality samples.

Flow models. Continuous normalizing flows (CNFs) use neural networks to define continuous ODE dynamics mapping noise to data [6]. Recent advancements have improved trajectory efficiency [37] and training methods [35]. Similar to PF-ODE-based diffusion solvers, these methods require ODE simulation.

Consistency models. To improve DM sampling efficiency, score model distillation techniques, like Consistency Models (CMs), enable few-step sampling [51]. CMs learn a mapping f_θ from a PF-ODE trajectory point x_t back to the initial state:

$$x_0 = f_\theta(x_t, t), \quad t \in [0, 1], \tag{2}$$

where x_0 is drawn from p_{data}. This allows for single-step sampling by drawing $x_1 \sim \gamma$ and applying f_θ, or multi-step sampling with a balance between efficiency and fidelity.

3. Methodology

Assume that a pre-trained generative model is given, which provides a one-to-one mapping Φ from the noise space to the data space. The data x_0 and noise x_1 both belong to $\mathbb{R}^d$, and $x_0 = \Phi(x_1)$. The observation is y, and the goal is to sample the data x_0 from the posterior distribution $p(x_0|y)$. We derive the posterior sampling of the data vector x_0 via that of the noise vector x_1, making use of the mapping Φ.

Likelihood and posterior. We consider a general observation model where the conditional law $p(y|x_0)$ is known and differentiable. Define the negative log conditional likelihood as $L_y(x_0) := -\log p(y|x_0)$, which is differentiable with respect to x_0 for fixed y. A typical case is the inverse problem setting: the *forward* model is

$$y = \mathcal{A}(x_0) + n, \tag{3}$$

where $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^d$ is the (possibly nonlinear) measurement operator, and n is the additive noise. For fixed y, we aim to sample x_0 from $p(x_0|y) = p(y|x_0)p(x_0)/p(y) \propto p(y|x_0)p(x_0)$, where $p(x_0)$ is the true prior distribution of all data x_0, which we now denote as p_{data}. We also call $p(x_0|y)$ the *true* posterior of x_0, donated as

$$p_{0,y}(x_0) := p(x_0|y) \propto p(y|x_0)p_{\text{data}}(x_0). \tag{4}$$

Posterior approximated via generative model. The true data prior p_{data} is nonlinear and complicated. Let p_{model} denote the prior distribution approximated by a pre-trained generative model $x_0 = \Phi(x_1)$, where $x_1 \sim \gamma$. A distribution from which samples are easily generated, such as the standard multi-variate Gaussian, is typically chosen for γ; we choose $\gamma = \mathcal{N}(0, I)$. In other words,

$$p_{\text{data}} \approx p_{\text{model}} = \Phi_{\#}\gamma. \tag{5}$$

Replacing p_{data} with p_{model} in (4) gives the *model* posterior of x_0, denoted $\tilde{p}_{0,y}$, which approximates the true posterior:

$$p_{0,y}(x_0) \approx \tilde{p}_{0,y}(x_0) \propto p(y|x_0)\Phi_{\#}\gamma(x_0). \tag{6}$$

Because $x_0 = \Phi(x_1)$, we have that $\tilde{p}_{0,y} = \Phi_{\#}\tilde{p}_{1,y}$, where, by a change of variable from (6),

$$\tilde{p}_{1,y}(x_1) \propto p(y|\Phi(x_1))\gamma(x_1). \tag{7}$$

The distribution $\tilde{p}_{1,y}(x_1)$ approximates the posterior distribution $p(x_1|y)$ in the noise space. When $p_{\text{data}} = \Phi_{\#}\gamma$, we have $p_{0,y} = \tilde{p}_{0,y}$ and $p(\cdot|y) = \tilde{p}_{1,y}$. When the generative model prior is inexact, the error in approximating the posterior can be bounded by that in approximating the data prior; see more in Section 4.

Posterior sampling by Langevin dynamics. It is direct to sample the approximated posterior (7) in the noise space using Langevin dynamics. Specifically, since we have $\gamma(x_1) \propto \exp(-\|x_1\|^2/2)$ and $\log p(y|\Phi(x_1)) = -L_y(\Phi(x_1))$, the following SDE of x_1 will have $\tilde{p}_{1,y}$ as its equilibrium distribution (proved in Lemma A.1):

$$dx_1 = -(x_1 + \nabla_{x_1} L_y(\Phi(x_1)))dt + \sqrt{2}dW_t. \tag{8}$$

The sampling in the noise space gives the sampling in the data space by the one-to-one mapping of the generative model, namely $x_0 = \Phi(x_1)$.

Example 3.1 (Inverse problem with Gaussian noise). For (3) with white noise, i.e., $n \sim \mathcal{N}(0, \sigma^2 I)$, we have that, with a constant c depending on (σ, d),

$$L_y(x_0) = -\log p(y|x_0) = \frac{1}{2\sigma^2}\|y - \mathcal{A}(x_0)\|_2^2 + c.$$

The noise-space SDE (8) can be written as

$$dx_1 = -\left(x_1 + \nabla_{x_1}\frac{\|\mathbf{y} - \mathcal{A}(x_0)\|_2^2}{2\sigma^2}\right)dt + \sqrt{2}dW_t.$$

Given $L_y(x_0)$, standard techniques can be used to sample (overdamped) Langevin dynamics (8). Evaluation of the gradient $\nabla_{x_1} L_y(x_0)$ is the major computational cost, requiring differentiation through the model Φ. One technique to improve sampling efficiency is to employ a warm-start of the SDE integration by letting the minimization-only dynamics (using $\nabla_{x_1} L_y(x_0)$) to converge to a minimum first, especially when the posterior concentrates around a particular point. We postpone the algorithmic details to Section 5.

4. Theory

In this section, we derive the theoretical guarantee of the model posterior $\tilde{p}_{0,y}$ in (6) to the true posterior $p_{0,y}$ in (4), and also extend to the computed posterior $\tilde{p}^S_{0,y}$ by discrete-time SDE integration. The analysis reveals a conditional number which indicates the intrinsic difficulty of the posterior sampling problem. All proofs are in Appendix A.

4.1. Total Variation (TV) guarantee and condition number

Consider the approximation (5), that is, the pre-trained model generates a data prior distribution $\Phi_{\#}\gamma$ that approximates the true data prior p_{data}. We quantify the approximation in TV distance, namely

$$\text{TV}(p_{\text{data}}, \Phi_{\#}\gamma) \le \varepsilon. \tag{9}$$

Generation guarantee in terms of TV bound has been derived in several flow-based generative model works, such as [7, 26, 33] on the PF-ODE of a trained score-based diffusion model [50], and [8] on the JKO-type flow model [58]. The following theorem proved in Appendix A shows that the TV distance between the model and true posteriors can be bounded proportional to that between the priors.

Theorem 4.1 (TV guarantee)**.** *Assuming* (9)*, then* $\text{TV}(p_{0,y}, \tilde{p}_{0,y}) \le 2\kappa_y\varepsilon$*, where*

$$\kappa_y := \frac{\sup_{x_0} p(y|x_0)}{\int p(y|x)p_{\text{data}}(x)dx}. \tag{10}$$

Remark 4.1 (κ_y as a condition number). The constant factor κ_y is determined by the true data prior p_{data} and the conditional likelihood $p(y|x_0)$ of the observation, and is independent of the flow model and the posterior sampling method. Thus κ_y quantifies an intrinsic "difficulty" of the posterior sampling, which can be viewed as a condition number of the problem.

Example 4.1 (Well-conditioned problem). Suppose $p(y|x_0) \le c_1$ for any x_0, and on a domain Ω_y of the data space,

$$P_{\text{data}}(\Omega_y) \ge \alpha > 0, \quad \text{and} \quad p(y|x_0) \ge c_0 > 0,\ \forall x_0 \in \Omega_y,$$

then we have $\int p(y|x)p_{\text{data}}(x)dx \ge \int_{\Omega_y} p(y|x)p_{\text{data}}(x)dx \ge \alpha c_0$, and then

$$\kappa_y \le \frac{1}{\alpha}\frac{c_1}{c_0}.$$

This shows that if the observation y can be induced from some cohort of x_0 and this cohort is well-sampled by the data prior p_{data} (the concentration of p_{data} on this cohort is

lower bounded by α), plus that the most likely x_0 is not too peaked compared to the likelihood of any other x_0 within this cohort (the ratio is upper bounded by c_1/c_0), then the posterior sampling is well-conditioned.

Example 4.2 (Ill-conditioned problem). Suppose $p(y|x_0)$ is peaked at one data value x_0' and almost zero at other places, and this x_0' lies on the tail of the data prior density p_{data}. This means that the integral $\int p(y|x_0)p_{\text{data}}(x_0)dx_0$ has all the contribution on a nearby neighborhood of x_0' on which p_{data} is small, resulting in a small value on the denominator of (10). Meanwhile, the value of $p(y|x_0')$ is large. In this case, κ_y will take a large value, indicating an intrinsic difficulty of the problem. Intuitively, the desired data value x_0' for this observation y is barely represented within the (unconditional) data distribution p_{data}, while the generative model can only learn from p_{data}. Since the pre-trained unconditional generative model does not have enough knowledge of such x_0', it is hard for the conditional generative model (based on the unconditional model) to find such a data value.

4.2. TV guarantee of the sampled posterior

Theorem 4.1 captures the approximation error of $\tilde{p}_{0,y}$ to the true posterior, where $\tilde{p}_{0,y}$ is the distribution of data x_0 when the noise x_1 in noise space achieves the equilibrium $\tilde{p}_{1,y}$ of the SDE (8). In practice, we use a numerical solver to sample the SDE in discrete time. The convergence of discrete-time SDE samplers to its equilibrium distribution has been established under various settings in the literature. Here, we assume that the discrete-time algorithm to sample the Langevin dynamics of x_1 outputs $x_1 \sim \tilde{p}^S_{1,y}$, which may differ from but is close to the equilibrium $\tilde{p}_{1,y}$. Specifically, suppose $\text{TV}(\tilde{p}_{1,y}, \tilde{p}^S_{1,y})$ is bounded by some ε_S.

Lemma 4.2 (Sampling error)**.** *If* $\text{TV}(\tilde{p}_{1,y}, \tilde{p}^S_{1,y}) \le \varepsilon_S$*, then* $\text{TV}(\tilde{p}_{0,y}, \tilde{p}^S_{0,y}) \le \varepsilon_S$.

The lemma is by Data Processing Inequality, and together with Theorem 4.1 it directly leads to the following corollary on the TV guarantee of the sampled posterior.

Corollary 4.3 (TV of sampled posterior)**.** *Assuming* (9) *and* $\text{TV}(\tilde{p}_{1,y}, \tilde{p}^S_{1,y}) \le \varepsilon_S$*, then*

$$\text{TV}(p_{0,y}, \tilde{p}^S_{0,y}) \le 2\kappa_y \varepsilon + \varepsilon_S.$$

5. Algorithm

Numerical integration of the Langevin dynamics. To numerically integrate the noise-space SDE (8), one can use standard SDE solvers. We adopt the Euler-Maruyama (EM) scheme. Let $\tau > 0$ be the time step, and denote the discrete sequence of x_1 as z^i, $i = 0, 1, \cdots$. The EM scheme gives, with $\xi^i \sim \mathcal{N}(0, I)$ and $g^i := \nabla_{x_1} L_y(x_0)|_{x_1 = z^i}$,

$$z^{i+1} = (1 - \tau)z^i - \tau g^i + \sqrt{2\tau}\xi^i. \tag{11}$$

See Algorithm 1 for an outline of our approach using EM. However, any general numerical scheme for solving SDEs can be applied; see Table A.4 in Appendix C for a comparison between our method using EM discretization and exponential integrator (EI) [24]. An initial value of z^0 in the noise space is required. We adopt a warm-start procedure to initialize sampling; additional details are provided below.

Algorithm 1 Posterior Sampling in Noise Space

Require: Forward model $\mathcal{A}$, measurement y, loss function L_y, pre-trained noise-to-data map Φ, number of steps N, step size τ, and initial x_1^0
for $i = 0, \ldots, N$ **do**
  $x_0^i \leftarrow \Phi(x_1^i)$
  $g^i \leftarrow \nabla_{x_1^i} L_y(x_0^i)$
  $\xi^i \sim \mathcal{N}(0, I)$
  $x_1^{i+1} \leftarrow x_1^i - \tau(x_1^i + g^i) + \sqrt{2\tau}\xi^i$
end for
return $x_0^1, x_0^2, \ldots, x_0^N$

Computation of $\nabla x_1 L_y(x_0)$. The computation of the loss gradient depends on the type of generative model representing the mapping Φ. For instance, if Φ is computed by solving an ODE driven by a normalizing flow, then its gradient can be computed using the adjoint sensitivity method [6]. If Φ is a DM or CM sampler, one can backpropagate through the nested function calls to the generative model. Since we use one- or few-step CM sampling to represent Φ in the experiments, we take the latter approach to compute $\nabla x_1 L_y(x_0)$.

Choice of initial value and warm-start. A natural initialization for the noise variable z^0 is a random sample $z^0 \sim \gamma$, which aligns with the data prior but may lie far from the posterior. To address this, we warm-start the sampler by optimizing $L_y(x_0)$ w.r.t. x_1 using standard optimizers (e.g., Adam). We use K Adam steps and set z^0 to the resulting output before starting EM sampling. Further details are provided in Appendix B.1.

Computational requirements. The main computational burden arises from computing the loss gradient $\nabla_{x_1} L_y(x_0)$, which requires differentiating through the mapping Φ. This burden can be reduced by selecting a Φ with a small number of function evaluations (NFEs). Additional overhead comes from the burn-in or warm start needed to initialize EM simulation with z^0. Consequently, the total NFEs for simulating N steps of EM to generate N samples is $\eta \cdot (K + N)$, where η represents the NFEs required to evaluate Φ. However, this cost diminishes over time, as EM simulation progressively reduces the NFEs per sample, asymptotically approaching η. We use CM sampling to represent Φ, achievable with $\eta = 1$ or 2. Although multi-step ($\eta > 1$) CM sampling is typically

stochastic [51], we fix the noise to create a deterministic mapping. Further details are in Appendix B.1.

Role of EM step size τ. The step size of EM, τ, controls the time scales over which the Langevin dynamics are simulated with respect to the number of EM steps. Larger τ results in more rapid exploration of the posterior, potentially leading to more diverse samples over shorter timescales. However, τ must also be kept small enough to ensure the stability of EM sampling. Thus, this hyper-parameter provides control over sample diversity. Choosing large τ while maintaining stability can yield diverse samples, potentially revealing uncertain semantic features within the posterior.

6. Experiments

Baselines. We group baselines into two categories. **(1) DM-based methods:** DPS [12], LGD [49], and MPGD [21] use stronger priors than our method, making them strong but backbone-incompatible baselines. To ensure fairer comparison, we introduce **(2) CM-based variants:** where each DM method is adapted to a consistency model (CM) backbone. We also include CMEdit, a CM-based sampler from [51], for linear tasks. All DM baselines use the EDM model from [51], and CM baselines use the corresponding LPIPS-distilled CM. Full details and hyperparameters are in Appendix B.2.

Datasets. We conduct experiments on LSUN-Bedroom (256×256)[60] and ImageNet (64×64)[13], using 100 validation images each. All experiments use pre-trained Consistency Models (CMs) from [51], distilled with the LPIPS objective from EDM models [29]. Further method and hyperparameter details are in Appendix B.1. For linear inverse problems, we consider: (i) random mask inpainting; (ii) super-resolution via adaptive average pooling; and (iii) Gaussian deblurring with a 61×61 kernel standard deviation of 3.0. Nonlinear tasks include: (i) neural network-based deblurring [52]; (ii) phase retrieval via Fourier magnitude; and (iii) HDR reconstruction via clipping scaled intensities. Gaussian noise with $\sigma = 0.1$ is added to all tasks except phase retrieval where $\sigma = 0.05$. See Appendix B.3 for operator details, and Appendices C, D for more results.

Metrics. To evaluate reconstruction fidelity, we report PSNR, SSIM, LPIPS, and Fréchet Inception Distance (FID). For diversity, we use: (i) Diversity Score (DS), computed as the ratio of inter- to intra-cluster distances across six-nearest-neighbor clusters of ResNet-50 features; and (ii) Average CLIP Cosine Similarity (CS), measuring the mean cosine similarity between CLIP embeddings of all sample pairs for a given image.

6.1. Image Restoration Results

Linear inverse problems. We compare our method against baselines for point estimation under linear forward models, using 10 samples per method across 100 validation images. Results for LSUN-Bedroom (256×256) and ImageNet (64×64) are shown in Table 1 (top and bottom, respectively), with visual comparisons in Figures 2 and 3. Our approach outperforms CM baselines with higher fidelity and fewer artifacts and remains competitive with DM baselines in both visual quality and quantitative metrics.

Nonlinear inverse problems. Quantitative results for nonlinear tasks on 100 LSUN-Bedroom images are reported in Table 2, using 10 samples per image. Our method performs competitively with CM-based baselines and matches the quality of DM-based methods. Visual results are shown in the bottom three rows of Figure 2. While CM variants and MPGD-DM struggle with artifact removal and noise, our approach produces clean, detailed reconstructions comparable to DM outputs. Notably, in challenging settings like phase retrieval, our method achieves PSNR and SSIM on par with DM baselines, reflecting strong alignment with the ground truth.

6.2. Diversity of posterior samples

To assess sample diversity, we compare our method with strong baselines based on DM, DPS, and LGD. We generate 25 samples per image on 100 LSUN-Bedroom (256×256) for all six linear and nonlinear tasks. As shown in Table 3, our method matches or surpasses DM baselines in diversity metrics. Figure 4 illustrates the visual diversity, especially in inpainting (top three rows) and nonlinear deblurring (bottom three rows). Our approach captures varying high-level features like lighting and shading and reveals semantic variability, e.g., windows and lamps differ significantly across samples.

7. Ablation Study

Number of warm-start iterations (K). We investigate how the number of warm-start optimization steps K affects reconstruction quality and diversity. As shown in the top row of Figure 5, increasing K leads to consistent improvements across fidelity metrics, including PSNR and SSIM (left), and perceptual metrics such as LPIPS and FID (right). Notably, FID drops significantly from 95 to below 82.5 as K increases from 200 to 1200. Simultaneously, diversity improves, with Diversity Score (DS) increasing and CLIP Cosine Similarity (CS) decreasing (middle), indicating that warm-starting helps explore the posterior more effectively.

Number of EM iterations (N). We analyze the impact of EM sampling iterations N for both 8× super-resolution and nonlinear deblurring. As seen in the bottom row of Figure 5, performance is relatively stable over 50 EM steps, with minimal variation across PSNR, LPIPS, and FID. This suggests that a small number of EM steps (e.g., $N \leq 10$) suffices for accurate posterior sampling, enabling efficient generation without sacrificing quality.

Table 1. Quantitative comparison of linear restoration tasks on LSUN-Bedroom (256 x 256) (top) and ImageNet (64 x 64) (bottom).

Method	8x Super-resolution				Gaussian Deblur				10% Inpainting			
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS↓	FID ↓
DPS-DM	20.4*	0.538*	0.470*	67.7*	22.1	0.589	0.407	65.3	22.4	0.634	0.417	67.7
MPGD-DM	19.2	0.338	0.689	288	23.6*	0.579	0.438	85.0	15.4	0.176	0.667	221
LGD-DM	20.1	0.529	0.483	69.3	22.2	0.590*	0.371*	60.1*	24.7*	0.742*	0.289*	47.3*
DPS-CM	10.7	0.077	0.758	307	11.2	0.092	0.735	279	19.9	0.454	0.517	128
LGD-CM	10.5	0.072	0.764	316	11.1	0.092	0.737	283	19.9	0.475	0.514	134
CMEdit	N/A				N/A				18.0	0.523	0.548	167
Ours(1-step)	20.4	**0.535**	**0.418**	**71.1**	**22.4**	**0.598**	**0.368**	70.6	**23.8**	**0.682**	**0.358**	**72.9**
Ours(2-step)	**20.5**	0.534	0.433	72.2	21.3	0.554	0.421	**69.2**	22.2	0.611	0.419	75.6

Method	4x Super-resolution				Gaussian Deblur				20% Inpainting			
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS↓	FID ↓
DPS-DM	21.0*	0.531	0.310*	110*	19.2	0.429	0.348*	117*	22.3*	0.664*	0.220*	89.2*
LGD-DM	21.0*	0.536*	0.311	114	19.6*	0.432*	0.352	117*	22.1	0.652	0.228	96.2
DPS-CM	12.8	0.168	0.602	267	9.89	0.093	0.650	334	18.9	0.470	0.371	167
LGD-CM	12.8	0.164	0.607	269	10.1	0.097	0.668	363	18.7	0.451	0.380	173
Ours(1-step)	16.9	**0.418**	**0.388**	**129**	**18.2**	**0.413**	**0.381**	**134**	**20.3**	**0.600**	**0.304**	**124**
Ours(2-step)	**18.1**	0.412	0.410	151	17.2	0.347	0.435	150	18.6	0.458	0.439	161

Bold denotes the best CM method, underline denotes the second best CM method, and * denotes the best DM method.

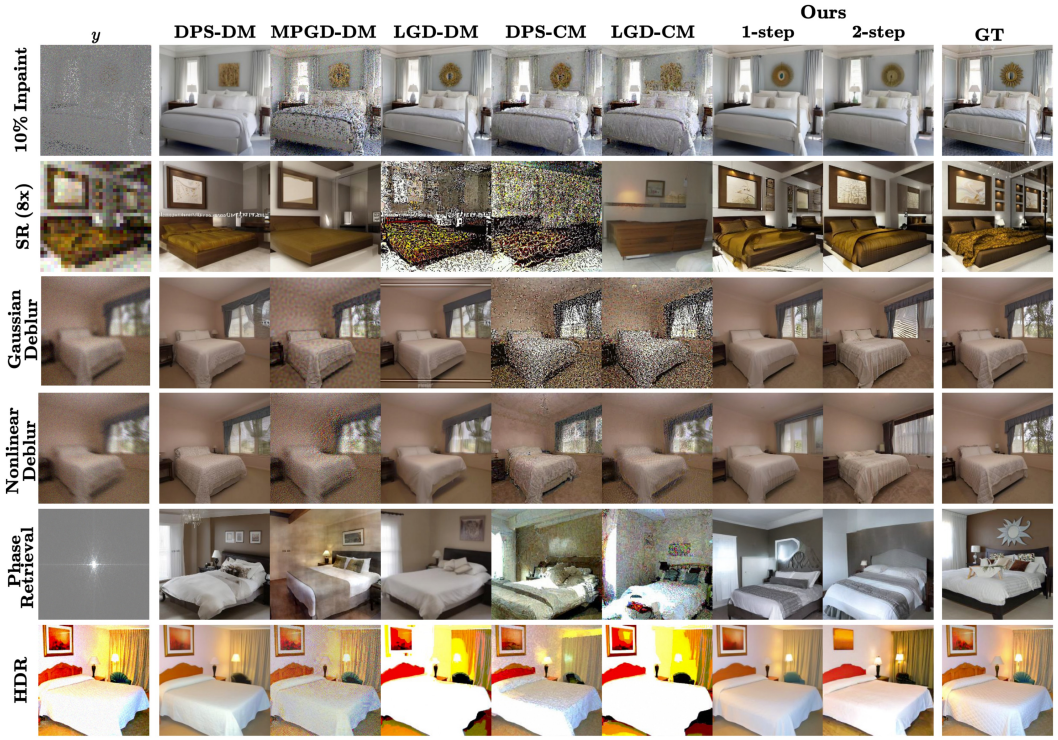

Figure 2. Image reconstructions for the linear and nonlinear tasks on LSUN-Bedroom (256 x 256).

8. Related works

Posterior sampling with generative models. Diffusion-based inverse problem solvers include task-specific methods [34, 38, 45], optimized approaches [36, 44, 46], and training-free techniques leveraging pre-trained diffusion priors [10–12, 15, 21, 30, 31, 48, 49, 53]. Early training-free methods used measurement-space projections [9, 47] or spectral consistency [30, 31, 53], while others enforced manifold constraints [11, 21]. Recent works approximate the measurement likelihood to address noisy and nonlinear problems [12, 48, 49]. Diffusion posterior sampling with provable guarantees is emerging [5, 59]: [59] introduce alter-

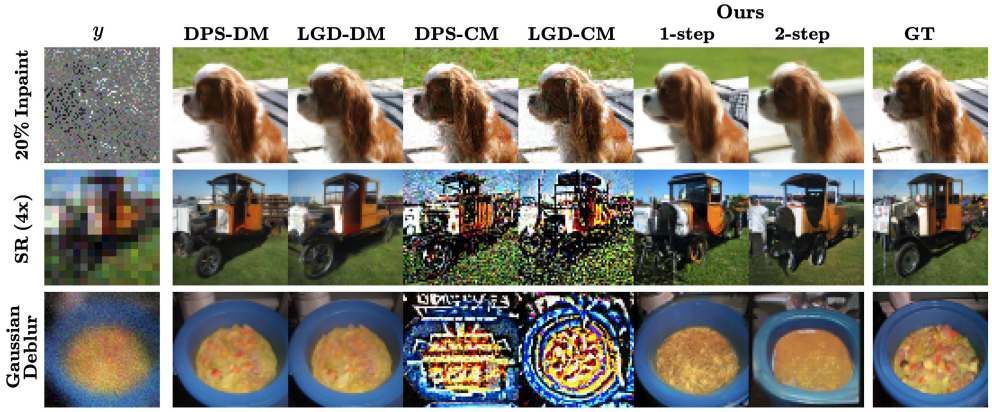

Figure 3. Image reconstructions for the linear tasks on ImageNet (64 x 64).

Table 2. Quantitative comparison of nonlinear image restoration tasks on LSUN-Bedroom (256 x 256).

Method	Nonlinear Deblur				Phase Retrieval				HDR Reconstruction			
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS↓	FID ↓	PSNR ↑	SSIM ↑	LPIPS↓	FID ↓
DPS-DM	21.6	0.586	0.413	75.7*	10.7	0.302	0.697*	90.1	21.7*	0.659*	0.396*	69.6*
MPGD-DM	17.0	0.194	0.683	259	9.96	0.271	0.728	118	20.5	0.586	0.408	73.2
LGD-DM	22.3*	0.632*	0.408*	106	10.8*	0.351*	0.709	82.0*	12.4	0.459	0.560	172
DPS-CM	17.7	0.303	0.574	137	10.1	0.197	0.726	195	13.5	0.405	0.597	173
MPGD-CM	13.1	0.100	0.762	306	9.39	0.111	0.786	312	11.7	0.296	0.638	223
LGD-CM	**21.3**	0.519	0.482	163	9.36	0.113	0.767	186	11.2	0.397	0.621	245
Ours(1-step)	20.3	**0.566**	**0.440**	76.7	**10.3**	**0.315**	0.709	82.9	**19.6**	**0.599**	**0.436**	**88.0**
Ours(2-step)	18.7	0.501	0.492	**73.3**	10.2	0.309	**0.708**	**81.4**	16.6	0.481	0.532	101

Bold denotes the best CM method, underline denotes the second best CM method, and * denotes the best DM method.

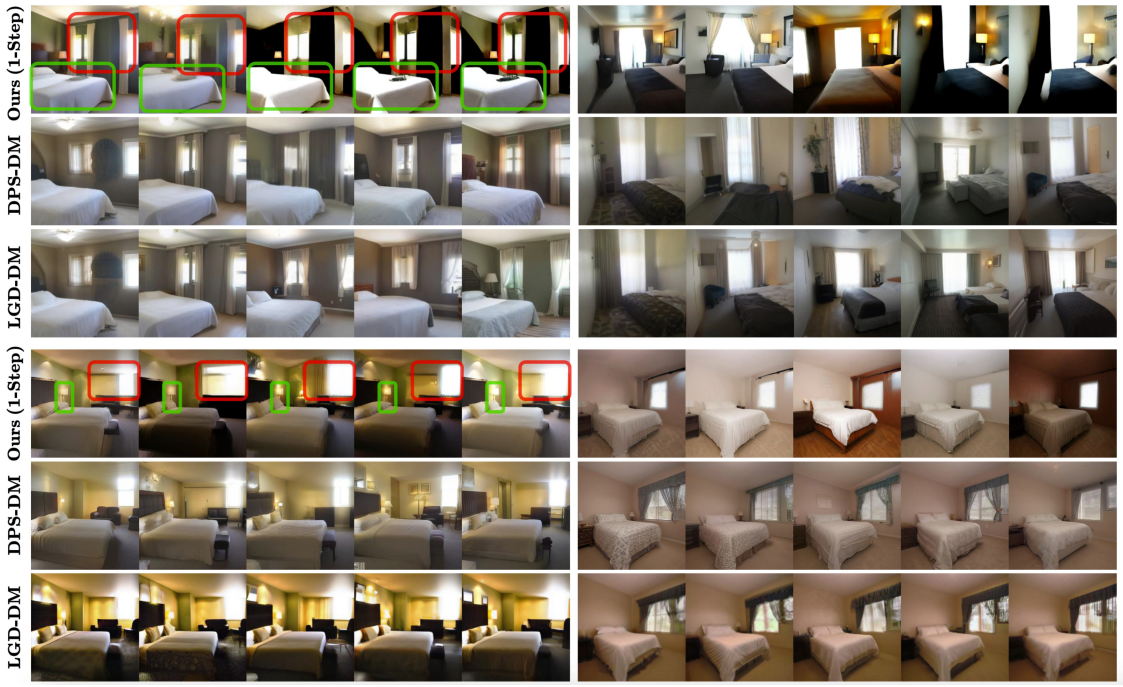

Figure 4. Posterior samples for the inpainitng (10%) (top three rows) and nonlinear deblur (bottom three rows) tasks on LSUN-Bedroom (256 x 256). Green boxes highlight low-uncertainty features and red boxes highlight highly uncertain features.

Table 3. Quantitative comparison of diversity metrics on linear and non-linear image restoration tasks on LSUN-Bedroom (256 x 256).

Method	SR(8x)		Gaussian Deblur		10% Inpainting		Nonlinear Deblur		Phase Retrieval		HDR Reconstruction	
	DS ↑	CS ↓	DS ↑	CS ↓	DS ↑	CS ↓	DS ↑	CS ↓	DS ↑	CS ↓	DS ↑	CS ↓
DPS-DM	2.14	**0.843**	2.10	0.938	2.33	0.876	2.22	0.924	2.42	**0.809**	2.25	**0.873**
LGD-DM	2.35	0.881	2.19	0.925	2.28	0.872	2.11	0.923	2.36	0.815	3.14	0.914
Ours(1-step)	**3.01**	0.879	**3.26**	0.997	**3.15**	0.869	**2.80**	0.912	**3.08**	0.914	3.09	0.927
Ours(2-step)	2.67	0.919	2.62	**0.866**	2.48	**0.864**	2.69	**0.885**	2.89	0.862	**3.23**	0.904

Bold denotes the best method, underline denotes the second best method.

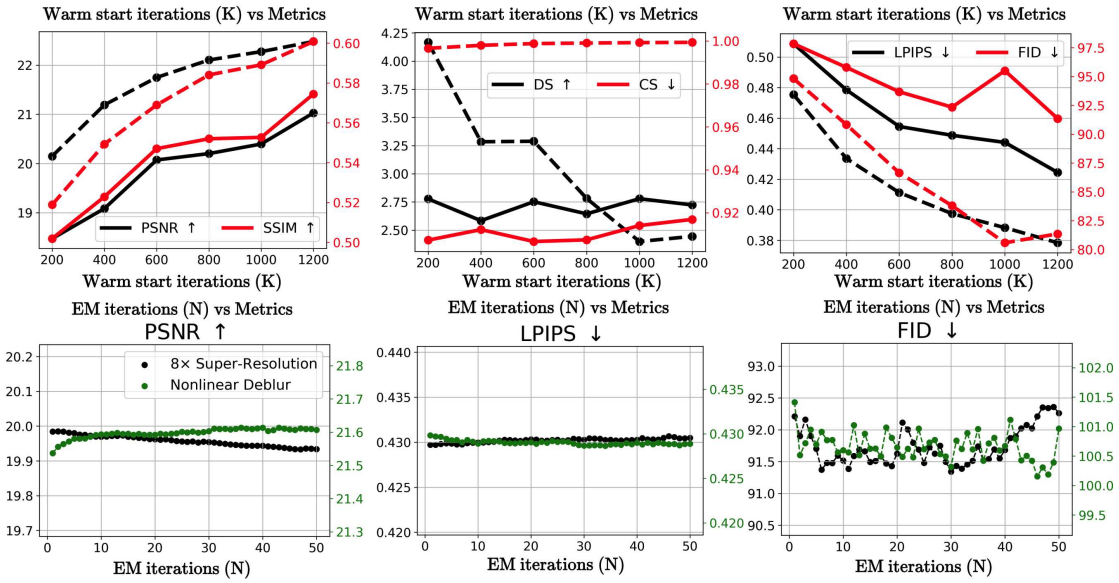

Figure 5. **Ablation study on warm-start and EM iterations. Top row:** Effect of warm-start iterations K on various metrics. Increasing K improves fidelity (PSNR, SSIM), perceptual quality (LPIPS, FID), and diversity (higher DS, lower CS). **Bottom row:** Effect of EM sampling iterations N on PSNR, LPIPS, and FID for 8× super-resolution and nonlinear deblurring. Metrics remain stable across iterations, indicating fast convergence and efficiency.

nating projection with convergence guarantees, and [5] use tilted transport for linear cases. Flow models have also been adapted, e.g., [42] extend ΠGDM [48] to CNFs. However, most require full sampling, limiting scalability. In contrast, our approach samples progressively in the noise space of one- or few-step mappings, enabling efficient posterior sampling.

Guided generation via noise space iteration. Generative models with deterministic mappings from latent noise to data—such as GANs [18], flows [6], and consistency models (CMs)[51]—enable noise optimization to guide generation via conditional signals[1–3, 16, 41, 54]. In GANs, this is used for text-to-image synthesis [16, 41] or task-specific guidance [3]. Flow-based models have adopted similar strategies for inverse problems [1, 54], for example, D-Flow [2] optimizes noise inputs to CNFs. Our method also operates in noise space but simulates Langevin dynamics for posterior sampling rather than point estimation.

9. Discussion

We have outlined an approach for posterior sampling via Langevin dynamics in the noise space of a generative model. Using a CM mapping from noise to data, our posterior sampling provides solutions to general noisy image inverse problems, demonstrating superior reconstruction fidelity to other CM methods and competitiveness with diffusion baselines. A primary limitation of our approach is the low visual quality in some posterior samples. Fidelity drawbacks can be attributed to a relatively poor approximation of the prior by CMs. Future work will focus on improving the fidelity of diverse samples, perhaps by using more accurate prior models and adaptive simulation of the SDE. Regardless, our method produces highly diverse samples, representing meaningful semantic uncertainty of data features within the posterior.

Acknowledgments

MR, YX, and XC were partially supported by National Science Foundation (NSF) DMS-2134037. YX was also partially supported by NSF DMS-2220495. XC was also partially supported by NSF DMS-2237842 and Simons Foundation MPS-MODL-00814643. JL was partially supported by NSF DMS-2309378.

References

[1] Muhammad Asim, Max Daniels, Oscar Leong, Ali Ahmed, and Paul Hand. Invertible generative models for inverse problems: mitigating representation error and dataset bias. In *ICML*, 2020. 8

[2] Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. In *ICML*, 2024. 8

[3] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. In *ICML*, 2018. 8

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 1

[5] Joan Bruna and Jiequn Han. Posterior sampling with denoising oracles via tilted transport. Available online : https://arxiv.org/abs/2407.00745, 2024. 6, 8

[6] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018. 2, 4, 8

[7] Sitan Chen, Giannis Daras, and Alex Dimakis. Restoration-degradation beyond linear diffusions: A non-asymptotic analysis for ddim-type samplers. In *ICML*, 2023. 3

[8] Xiuyuan Cheng, Jianfeng Lu, Yixin Tan, and Yao Xie. Convergence of flow-based generative models via proximal gradient descent in Wasserstein space. *IEEE Transactions on Information Theory*, 2024. 3

[9] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. In *ICCV*, 2021. 1, 6

[10] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *CVPR*, 2022. 1, 6

[11] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *NeurIPS*, 2022. 1, 6

[12] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *ICLR*, 2023. 1, 2, 5, 6, 3, 4

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[14] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 1

[15] Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *ICLR*, 2024. 6

[16] Federico Galatolo., Mario Cimino., and Gigliola Vaglini. Generating images from caption and vice versa via clip-guided generative latent space search. In *International Conference on Image Processing and Vision Engineering*, 2021. 8

[17] Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In *CVPR*, 2020. 6

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 1, 8

[19] Monson Hayes. The reconstruction of a multidimensional sequence from the phase or magnitude of its fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(2):140–154, 1982. 4

[20] Linchao He, Hongyu Yan, Mengting Luo, , Hongjie Wu, Kunming Luo, Wang Wang, Wenchao Du, Hu Chen, Hongyu Yang, Yi Zhang, and Jiancheng Lv. Fast and stable diffusion inverse solver with history gradient update. Available online : https://arxiv.org/pdf/2307.12070, 2023. 3

[21] 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 *ICLR*, 2024. 1, 5, 6

[22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 1

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 1, 2

[24] Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010. 4

[25] Matthew Hoffman, Pavel Sountsov, Joshua V Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. Neutralizing bad geometry in hamiltonian monte carlo using neural transport. Available online: https://arxiv.org/abs/1903.03704, 2019. 2, 6

[26] Daniel Zhengyu Huang, Jiaoyang Huang, and Zhengjiang Lin. Convergence analysis of probability flow ode for score-based generative models. Available online: https://arxiv.org/abs/2404.09730, 2024. 3

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1

[28] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 1

[29] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 2, 5

[30] Bahjat Kawar, Gregory Vaksman, and Michael Elad. SNIPS: Solving noisy inverse problems stochastically. In *NeurIPS*, 2021. 1, 6

[31] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *NeurIPS*, 2022. 1, 6

[32] Kim et al. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *ICLR*, 2024. 5

[33] Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards faster non-asymptotic convergence for diffusion-based generative models. In *ICLR*, 2024. 3

[34] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. SRDiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. 1, 6

[35] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. 2

[36] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I2SB: Image-to-Image Schrödinger bridge. In *ICML*, 2023. 6

[37] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 2

[38] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 1, 6

[39] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. Available online : https://arxiv.org/abs/1411.1784, 2014. 1

[40] Erik Nijkamp, Ruiqi Gao, Pavel Sountsov, Srinivas Vasudevan, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. Mcmc should mix: Learning energy-based model with neural transport latent space mcmc. In *ICLR*, 2022. 2, 6

[41] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 8

[42] Ashwini Pokle, Matthew J Muckley, Ricky TQ Chen, and Brian Karrer. Training-free linear image inversion via flows. *TMLR*, 2023. 8

[43] Vishal Purohit, Junjie Luo, Yiheng Chi, Qi Guo, Stanley H. Chan, and Qiang Qiu. Generative Quanta Color Imaging. In *CVPR*, 2024. 1

[44] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH*, 2022. 1, 6

[45] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022. 6

[46] Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Conditional simulation using diffusion schrödinger bridges. In *UAI*, 2022. 1, 6

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 2, 6

[48] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *ICLR*, 2023. 1, 6, 8

[49] Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *ICML*, 2023. 1, 5, 6, 3

[50] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 2, 3

[51] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023. 2, 5, 8, 3

[52] Phong Tran, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *CVPR*, 2021. 5, 3

[53] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *ICLR*, 2023. 1, 6

[54] Jay Whang, Qi Lei, and Alex Dimakis. Solving inverse problems with a flow-based noise model. In *ICML*, 2021. 8

[55] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. VAEBM: a symbiosis between variational autoencoders and energy-based models. In *ICLR*, 2021. 2, 6

[56] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *ICML*, 2016. 2, 6

[57] Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 2, 6

[58] Chen Xu, Xiuyuan Cheng, and Yao Xie. Normalizing flow neural networks by JKO scheme. In *NeurIPS*, 2023. 3

[59] Xingyu Xu and Yuejie Chi. Provably robust score-based diffusion posterior sampling for plug-and-play image reconstruction. In *NeurIPS*, 2024. 6

[60] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. Available online : https://arxiv.org/abs/1506.03365, 2024. 5

[61] Jing Zhang, Jianwen Xie, Nick Barnes, and Ping Li. Learning generative vision transformer with energy-based latent space for saliency prediction. *NeurIPS*, 2021. 6