Diffusion Model Based Posterior Sampling for Noisy Linear Inverse Problems

Xiangming Meng XIANGMINGMENG@INTL.ZJU.EDU.CN
Zhejiang University - University of Illinois Urbana-Champaign Institute
Zhejiang University
Haining 314400, China

Yoshiyuki Kabashima KABA@PHYS.S.U-TOKYO.AC.JP
Institute for Physics of Intelligence & Department of Physics
The University of Tokyo
Tokyo 113-0033, Japan

Editors: Vu Nguyen and Hsuan-Tien Lin

Abstract

With the rapid development of diffusion models and flow-based generative models, there has been a surge of interests in solving noisy linear inverse problems, e.g., super-resolution, deblurring, denoising, colorization, etc, with generative models. However, while remarkable reconstruction performances have been achieved, their inference time is typically too slow since most of them rely on the seminal diffusion posterior sampling (DPS) framework and thus to approximate the intractable likelihood score, time-consuming gradient calculation through back-propagation is needed. To address this issue, this paper provides a fast and effective solution by proposing a simple closed-form approximation to the likelihood score. For both diffusion and flow-based models, extensive experiments are conducted on various noisy linear inverse problems such as noisy super-resolution, denoising, deblurring, and colorization. In all these tasks, our method (namely DMPS) demonstrates highly competitive or even better reconstruction performances while being significantly faster than all the baseline methods.

Keywords: Inverse problems; diffusion models; flow-based models; image restoration.

1. Introduction

Many problems in science and engineering such as computer vision and signal processing can be cast as the following noisy linear inverse problems:

$$\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{n}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{M\times N}$ is a (known) linear mixing matrix, $\mathbf{n} \sim \mathcal{N}(\mathbf{n}; 0, \sigma^2\mathbf{I})$ is an i.i.d. additive Gaussian noise, and the goal is to recover the unknown target signal $\mathbf{x}_0 \in \mathbb{R}^{N\times 1}$ from the noisy linear measurements $\mathbf{y} \in \mathbb{R}^{M\times 1}$. Notable examples include a wide class of image restoration tasks like super-resolution (SR) Ledig et al. (2017), colorization Zhang et al. (2016), denoising Buades et al. (2005), deblurring Yuan et al. (2007), inpainting Bertalmio et al. (2000), as well as the well-known compressed sensing (CS) Candès et al. (2006); Candès and Wakin (2008) in signal processing. One big challenge of these linear inverse problems is that they are ill-posed O'Sullivan (1986), i.e., the solution to (1) is not unique (even in the noiseless case). This problem can be tackled from a

© 2024 X. Meng & Y. Kabashima.

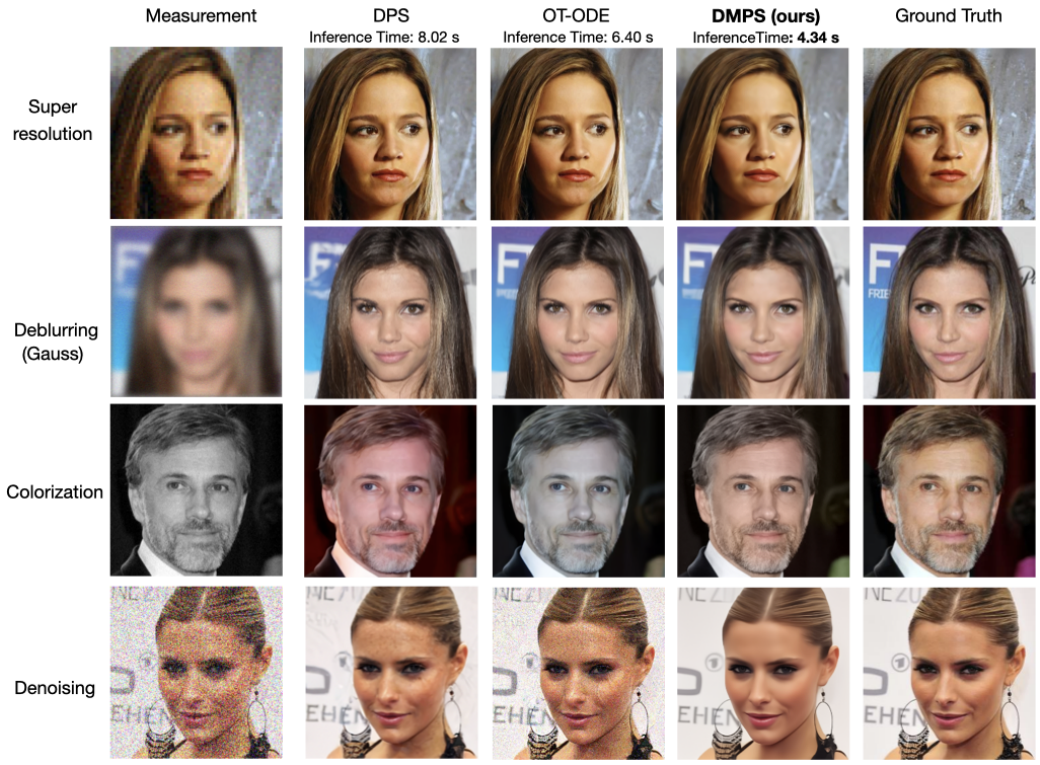


Figure 1: Typical results for different image restoration tasks on CelebA-HQ 256×256 validation set, along with the average inference time in seconds. It can be seen that our method (DMPS) achieves highly competitive or even better reconstruction performances with much less inference time. For a fair comparison, all the algorithms are run on the same flow-based model with NFE=50.

Bayesian perspective: suppose that the target signal $\mathbf{x}$ follows a proper *prior distribution* $p(\mathbf{x})$, given noisy observations $\mathbf{y}$, one can perform posterior sampling from $p(\mathbf{x}_0|\mathbf{y})$ to recover $\mathbf{x}_0$. Hence, an accurate prior $p(\mathbf{x}_0)$ is crucial in recovering $\mathbf{x}_0$. Various kinds of priors or structure constraints have been proposed, including sparsity Candès and Wakin (2008), low-rank Fazel et al. (2008), total variation Candès et al. (2006), just to name a few. However, such handcrafted priors might fail to capture the capture more rich structure of natural signals Ulyanov et al. (2018).

With the recent advent of diffusion models Sohl-Dickstein et al. (2015); Song and Ermon (2019); Ho et al. (2020); Dhariwal and Nichol (2021); Rombach et al. (2022) and flow-based models Lipman et al. (2022); Liu et al. (2022); Albergo et al. (2023); Ma et al. (2024), there has been a surge of interests in applying them to solve the linear inverse problems with remarkable performances Kadkhodaie and Simoncelli (2020, 2021); Jalal et al. (2021a,b); Kawar et al. (2021, 2022); Chung et al. (2022b,a); Wang et al. (2022); Meng and Kabashima (2023, 2024); Pokle et al. (2023). One fundamental challenge in this field is computing the score of *noise-perturbed likelihood* $p(\mathbf{y}|\mathbf{x}_t)$, i.e., $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$, where $\mathbf{x}_t$ is a noise-perturbed version of $\mathbf{x}_0$ at time instance t defined by the forward process of DM Ho et al. (2020); Song and Ermon (2019). This is because

while $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ is easily obtained for $t = 0$ from (1), it is intractable for general $t > 0$. To address this challenge, most diffusion and flow-based methods adopt the diffusion posterior sampling (DPS) framework Chung et al. (2022a) which leverages the Tweedie's formula Robbins (1992) to obtain a posterior estimate of $\mathbf{x_0}$. While DPS and its variants achieve excellent reconstruction performances, they suffer from a big disadvantage that their inference speed is very slow due to the time-consuming gradient calculation through back-propagation.

In this paper, we take an alternative perspective and provide a simple fast solution for solving the noisy linear inverse problems with diffusion or flow-based models by proposing a closed-from approximation to the intractable function $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$. Our primary goal is to reduce the inference time of existing methods with minimal degradation, rather than to compete with state-of-the-art performance. The key observation is that, the noise-perturbed likelihood $p(\mathbf{y}|\mathbf{x}_t) = \int p(\mathbf{y}|\mathbf{x}_0)p(\mathbf{x}_0|\mathbf{x}_t)d\mathbf{x}_0$ is unavailable due to the intractability of the reverse transition probability $p(\mathbf{x}_0|\mathbf{x}_t)$, so that one can obtain a closed-form approximation of it assuming an *uninformative* prior $p(\mathbf{x}_0)$. Interestingly, such assumption is asymptotically accurate when the perturbed noise in $\mathbf{x}_t$ negligibly small. The resultant algorithm is denoted as Diffusion Model based Posterior Sampling (DMPS), one approach that applies to both diffusion and flow-based models. Compared with the seminal DPS and its variants such as PGDM, thanks to the proposed closed-from approximation, no back-propagation through the pre-trained model is needed, thus significantly reducing the inference time. To verify its efficacy, a variety of experiments on different linear inverse problems such as image super-resolution, denoising, deblurring, colorization, are conducted. Remarkably, as shown in Figure 1, in all these tasks, despite its simplicity, DMPS achieves highly competitive or even better reconstruction performances, while the running time is significantly reduced.

2. Background

Diffusion models (DM) Song and Ermon (2019); Ho et al. (2020); Dhariwal and Nichol (2021); Song et al. (2023) and Flow-based models (such as flow matching, rectified flow) Lipman et al. (2022); Liu et al. (2022); Albergo et al. (2023); Ma et al. (2024) can be seen as a unified class of probabilistic generative models that learn to turning random noise into data samples $\mathbf{x}_0 \sim p(\mathbf{x}_0)$. The forward time-dependent process $\mathbf{x}_0 \to \mathbf{x}_1 \to \cdots \to \mathbf{x}_T$ can be described as follows:

$$\mathbf{x}_t = a_t \mathbf{x}_0 + b_t \epsilon, \tag{2}$$

where a_t is a decreasing function of t, b_t is an increasing function of t, and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is an i.i.d. standard Gaussian noise. Equivalently, the forward process (2) is modeled as

$$p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; a_t\mathbf{x}_0, b_t^2\mathbf{I}). \tag{3}$$

Both diffusion models and flow-based models aim to reverse the forward process (2) and generate new samples from a distribution that approximates the target data distribution $p(\mathbf{x}_0)$.

Diffusion Models: Diffusion models reverse the forward process (2) by performing a denoising task for each step, i.e., predicting the noise ϵ from $\mathbf{x}_t$. In the seminal work of DDPM Ho et al. (2020), $a_t = \sqrt{\bar{\alpha}_t}$, $b_t^2 = 1 - \bar{\alpha}_t$, where $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, $\alpha_t = 1 - \beta_t$, and $0 < \beta_1 < \beta_1 < \cdots < \beta_T < 1$ Ho et al. (2020). Denote $\mathrm{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ as the noise approximator from $\mathbf{x}_t$, one can generate samples following the estimated reverse process Ho et al. (2020) as

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathrm{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\big) + \beta_t \mathbf{z}_t, \tag{4}$$

where $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is an i.i.d. standard Gaussian noise. Note that in the variant ADM in Dhariwal and Nichol (2021), the reverse noise variance β_t is learned as $\{\tilde{\sigma}_t\}_{t=1}^T$, which further improves the performances of DDPM.

Diffusion models are also known as score-based generated models since the denoising process is equivalent to approximating the score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ Song and Ermon (2019); Song et al. (2020). For example, for DDPM, there is a one-to-one mapping between $\mathrm{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ and $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \mathrm{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t). \tag{5}$$

Flow-based Models: Flow-based models can be viewed as a generalization of diffusion models Lipman et al. (2022); Liu et al. (2022); Albergo et al. (2023); Ma et al. (2024), which introduce a probability ODE with a velocity field Lipman et al. (2022); Ma et al. (2024):

$$\dot{\mathbf{x}}_t = \mathbf{v}(\mathbf{x}_t, t), \tag{6}$$

where $\mathbf{v}(\mathbf{x}, t)$ can be obtained as the conditional expectation $\mathbf{v}(\mathbf{x}, t) = \mathbb{E}[\dot{\mathbf{x}}_t | \mathbf{x}_t = \mathbf{x}]$. Flow-based models solve the probability ODE (6) backwards by learning the velocity field $\mathbf{v}(\mathbf{x}, t)$ using a neural network $\mathrm{v}_{\boldsymbol{\theta}}(\mathbf{x}, t)$, and a first-order ODE solver can be realized as follows:

$$\mathbf{x}_{t-1} = \mathbf{x}_t - \mathrm{v}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\Delta_t, \tag{7}$$

where Δ_t is the sampling time interval. Interestingly, the score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ can also be expressed in terms of the velocity field Ma et al. (2024)

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) = b_t^{-1} \frac{a_t \mathrm{v}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \dot{a}_t \mathbf{x}_t}{\dot{a}_t b_t - a_t \dot{b}_t}. \tag{8}$$

Previous Methods with Diffusion and Flow-based Models: The problem of reconstructing $\mathbf{x}_0$ from noisy $\mathbf{y}$ in (1) can be cast as performing *posterior* inference, i.e.,

$$p(\mathbf{x}_0|\mathbf{y}) = \frac{p(\mathbf{x}_0)p(\mathbf{y}|\mathbf{x_0})}{p(\mathbf{y})}, \tag{9}$$

where $p(\mathbf{x}_0|\mathbf{y})$ is the *posterior* distribution. Ideally, one can directly train diffusion or flow-based models using samples from $p(\mathbf{x}|\mathbf{y})$. However, such a supervised approach is neither efficient nor flexible and most previous methods adopt an unsupervised approach Jalal et al. (2021a); Chung et al. (2022a); Song et al. (2022); Pokle et al. (2023): given a pre-trained diffusion model or flow-based model, one treats it as an implicit prior $p(\mathbf{x}_0)$ and then performs posterior sampling through a reverse sampling process $\mathbf{x}_T \to \cdots \mathbf{x}_t \to \mathbf{x}_{t-1} \to \cdots \to \mathbf{x}_0$. The main challenge is thus how to incorporate information of $\mathbf{y}$ within such reverse sampling process. Interestingly, while diffusion models and flow-based models admit slightly different forms, there exists a principled way thanks to the simple relation from the Bayes' rule (9),

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t), \tag{10}$$

where $p(\mathbf{x}_t|\mathbf{y})$ is the score of posterior distribution (we call *posterior score*), which is the sum of the prior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$, and the likelihood score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$. Given a pre-trained diffusion

model or flow-based model, the prior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ can be readily obtained from the pre-trained model outputs thanks to the intrinsic connections (5) (8). However, while $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ can be readily obtained from (1) when $t = 0$, it becomes intractable in the general case for $t > 0$ Chung et al. (2022a). To see this, one can equivalently write $p(\mathbf{y}|\mathbf{x}_t)$ as

$$p(\mathbf{y}|\mathbf{x}_t) = \int p(\mathbf{y}|\mathbf{x}_0)p(\mathbf{x}_0|\mathbf{x}_t)d\mathbf{x}_0, \tag{11}$$

where from the Bayes' rule,

$$p(\mathbf{x}_0|\mathbf{x}_t) = \frac{p(\mathbf{x}_t|\mathbf{x}_0)p(\mathbf{x}_0)}{\int p(\mathbf{x}_t|\mathbf{x}_0)p(\mathbf{x}_0)d\mathbf{x}_0}. \tag{12}$$

For both diffusion and flow-based models, although the forward transition probability $p(\mathbf{x}_t|\mathbf{x}_0)$ is exactly known as (3), the reverse transition probability $p(\mathbf{x}_0|\mathbf{x}_t)$ is difficult to obtain. Consequently, the remaining key challenge is the calculation of the *noise-perturbed likelihood score* $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$. A variety of methods Jalal et al. (2021a); Chung et al. (2022a); Song et al. (2022); Pokle et al. (2023) have been proposed to approximate $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ (or equivalently $p(\mathbf{y}|\mathbf{x}_t)$) and most of them build on the seminal work DPS Chung et al. (2022a) which leverages the Tweedie's formula to obtain the posterior estimate of $\mathbf{x}_0$ Robbins (1992); Chung et al. (2022a):

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) := \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \frac{1}{a_t}\left(\mathbf{x}_t + b_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right), \tag{13}$$

where $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ is approximated by the neural network as (5) and (8) for diffusion and flow-based models, respectively. In particular, DPS uses a Laplace approximation $p(\mathbf{y}|\mathbf{x}_t) \approx p(\mathbf{y}|\hat{\mathbf{x}}_0(\mathbf{x}_t)) = \mathcal{N}(\mathbf{A}\hat{\mathbf{x}}_0(\mathbf{x}_t); \sigma_y^2\mathbf{I})$, while both PGDM Song et al. (2022) and OT-ODE Pokle et al. (2023) use an approximation $p(\mathbf{y}|\mathbf{x}_t) \approx \mathcal{N}(\mathbf{A}\hat{\mathbf{x}}_0(\mathbf{x}_t); \gamma_t^2\mathbf{A}\mathbf{A}^T + \sigma_y^2\mathbf{I})$, where γ_t is a hyper-parameter for the variance term. Nevertheless, while DPS and its variants can achieve excellent reconstruction performances, they suffer from a significant drawback: the inference speed is very slow due to the time-consuming gradient of the pre-trained model output w.r.t. $\mathbf{x}_t$ in calculating the likelihood $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$.

3. Method

In this section, we propose a fast and efficient closed-form solution for the intractable likelihood score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$, termed as noise-perturbed pseudo-likelihood score. We first derive the results of $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ under the general settings (2-3), and then apply our results in diffusion and flow-based models, respectively.

3.1. Noise-Perturbed Pseudo-Likelihood Score

To tackle the intractability difficulty of $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$, we introduce a simple approximation under the following assumption:

Assumption 1 *(uninformative prior) The prior $p(\mathbf{x}_0)$ (12) is uninformative (flat) w.r.t. $\mathbf{x}_t$ so that $p(\mathbf{x}_0|\mathbf{x}_t) \propto p(\mathbf{x}_t|\mathbf{x}_0)$, where $\propto$ denotes equality up to a constant scaling.*

Note that while the uninformative prior assumption appears crude at first sight, it is asymptotically accurate when the perturbed noise in $\mathbf{x}_t$ becomes negligible, as verified in Appendix A.

Under Assumption 1, we obtain a simple *closed-form* approximation of $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ called noise-perturbed *pseudo-likelihood score* and denote as $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)$, as shown in Theorem 1.

Theorem 1 *(noise-perturbed pseudo-likelihood score for (2)) For the general forward process (2), under Assumption 1, the noise-perturbed likelihood score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ for $\mathbf{y} = \mathbf{A}\mathbf{x_0} + \mathbf{n}$ in (1) admits a closed-form*

$$\begin{aligned}&\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \simeq \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)\\ =&\frac{1}{a_t}\mathbf{A}^T\Big(\sigma_y^2\mathbf{I} + \frac{b_t^2}{a_t^2}\mathbf{A}\mathbf{A}^T\Big)^{-1}\big(\mathbf{y} - \frac{1}{a_t}\mathbf{A}\mathbf{x}_t\big).\end{aligned} \tag{14}$$

Proof. From Assumption 1, we have $p(\mathbf{x}_0|\mathbf{x}_t) \propto p(\mathbf{x}_t|\mathbf{x}_0)$. Recall that for the forward process (2), $p(\mathbf{x}_t|\mathbf{x}_0)$ is Gaussian (2). By completing the squares w.r.t. $\mathbf{x}_0$, an approximation for $p(\mathbf{x}_0|\mathbf{x}_t)$ can be derived as follows:

$$p(\mathbf{x}_0|\mathbf{x}_t) \simeq \mathcal{N}(\mathbf{x}_0; \frac{\mathbf{x}_t}{a_t}, \frac{b_t^2}{a_t^2}\mathbf{I}), \tag{15}$$

whereby $\mathbf{x}_0$ can be equivalently written as $\mathbf{x}_0 = \frac{\mathbf{x}_t}{a_t} + \frac{b_t}{a_t}\mathbf{w}$, where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Thus, from (1), we obtain an alternative representation of $\mathbf{y}$

$$\mathbf{y} = \frac{\mathbf{A}\mathbf{x}_t}{a_t} + \frac{b_t}{a_t}\mathbf{A}\mathbf{w} + \mathbf{n}. \tag{16}$$

After some simple algebra, the likelihood $p(\mathbf{y}|\mathbf{x}_t)$ can be approximated as $\tilde{p}(\mathbf{y}|\mathbf{x}_t)$

$$\tilde{p}(\mathbf{y}|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}; \frac{\mathbf{A}\mathbf{x}_t}{a_t}, \sigma_y^2\mathbf{I} + \frac{b_t^2}{a_t^2}\mathbf{A}\mathbf{A}^T), \tag{17}$$

where $\tilde{p}(\mathbf{y}|\mathbf{x}_t)$ is used to denote the *pseudo-likelihood* as opposed to the exact $p(\mathbf{y}|\mathbf{x}_t)$ due to Assumption 1. Using (17), one can readily obtain a closed-form solution for the noise-perturbed pseudo-likelihood score $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)$ as (14), which completes the proof. ■

As shown in Theorem 1, now we obtain a simple *closed-form* approximation for the intractable likelihood score, which is much easier to implement compared to DPS and its variants. In the special case when $\mathbf{A}$ itself is row-orthogonal, i.e., $\mathbf{A}\mathbf{A}^T$ is diagonal, the matrix inversion is trivial and (14) simply reduces to

$$[\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)]_m = \frac{\mathbf{a}_m^T\left(\mathbf{y} - \frac{1}{a_t}\mathbf{A}\mathbf{x}_t\right)}{\sigma_y^2 a_t + \frac{b_t^2}{a_t^2}\|\mathbf{a}_m\|_2^2}. \tag{18}$$

where $[\cdot]_m$ is the m-th element and $\mathbf{a}_m$ is the m-th row of $\mathbf{A}$. For general matrices $\mathbf{A}$, such an inversion is essential but it can also be efficiently implemented by resorting to singular value decomposition (SVD) of $\mathbf{A}$, as shown in Theorem 2.

Corollary 2 *(efficient computation via SVD) For the general forward process (2), the noise-perturbed pseudo-likelihood score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ in (14) of Theorem 1 can be equivalently computed as*

$$\begin{aligned}&\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \simeq \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)\\ =&\frac{1}{a_t}\mathbf{V}\mathbf{\Sigma}\Big(\sigma_y^2\mathbf{I} + \frac{b_t^2}{a_t^2}\mathbf{\Sigma}^2\Big)^{-1}\mathbf{U}^T\big(\mathbf{y} - \frac{1}{a_t}\mathbf{A}\mathbf{x}_t\big),\end{aligned} \tag{19}$$

where $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the SVD of $\mathbf{A}$ and $\mathbf{\Sigma}^2$ denotes element-wise square of $\mathbf{\Sigma}$.

Proof. The result is straightforward from Theorem 1. ■

Remark 3 *Thanks to SVD, there is no need to compute the matrix inversion in (14) for each t. Instead, one simply needs to perform SVD of $\mathbf{A}$ only once and then compute $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)$ by (19), which is quite simple since $\mathbf{\Sigma}$ is a diagonal matrix.*

3.2. DMPS: Diffusion Model Based Posterior Sampling

After obtaining the approximate results of the likelihood score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$, we can easily modify the sampling equations of the original diffusion and flow-based models from a unified Bayesian perspective. Here we introduce a simple yet universal procedure demonstrating how we can achieve this for both diffusion and flow-based models.

Step 1: Reformulate the original sampling equations for unconditional generation in the terms of the prior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$. This step requires building connections between the generative model (either diffusion or flow-based models) output with the score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$. For example, given the connections (5) (8), the original sampling equation (4) for DDPM and (7) for flow-based models can be rewritten using $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ as follows

$$\text{DDPM:}\ \ \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\big(\mathbf{x}_t + (1-\alpha_t)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\big) + \beta_t \mathbf{z}_t, \tag{20}$$

$$\text{Flow-based:}\ \ \mathbf{x}_{t-1} = \mathbf{x}_t - \big(\frac{\dot{a}_t}{a_t}\mathbf{x}_t + \frac{b_t(\dot{a}_t b_t - a_t \dot{b}_t)}{a_t}\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\big)\Delta_t, \tag{21}$$

$$\text{DDPM:}\ \ \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\big(\mathbf{x}_t + (1-\alpha_t)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)\big)\big) + \beta_t \mathbf{z}_t, \tag{22}$$

$$\text{Flow-based:}\ \ \mathbf{x}_{t-1} = \mathbf{x}_t - \big(\frac{\dot{a}_t}{a_t}\mathbf{x}_t + \frac{b_t(\dot{a}_t b_t - a_t \dot{b}_t)}{a_t}\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)\big)\big)\Delta_t, \tag{23}$$

Step 2: Replace the prior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ in the sampling equations obtained in Step 1 with the posterior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$ as (10). For example, for DDPM and flow-based models, the corresponding sampling equations (22-21) become

$$\text{DDPM:}\ \ \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\big(\mathbf{x}_t + (1-\alpha_t)(\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t))\big) + \beta_t \mathbf{z}_t, \tag{24}$$

$$\text{Flow-based:}\ \ \mathbf{x}_{t-1} = \mathbf{x}_t - \big(\frac{\dot{a}_t}{a_t}\mathbf{x}_t + \frac{b_t(\dot{a}_t b_t - a_t \dot{b}_t)}{a_t}(\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t))\big)\Delta_t, \tag{25}$$

Step 3: Replace the prior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ back in terms of the generative model outputs in the obtained sampling equations in Step 2. Subsequently, taking into account the additional terms due to the addition of likelihood score, we can easily obtain the final posterior sampling equations. For example, for DDPM and flow-based models, the corresponding sampling equations (24-25) finally

Algorithm 1 DMPS (DDPM version)

Input: $\mathbf{y}$, $\mathbf{A}$, σ_y^2, $\{\tilde{\sigma}_t\}_{t=1}^T$, λ

Initialization: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$

1 **for** $t = T$ ***to*** 1 **do**

2 Draw $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

3 $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathrm{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right) + \tilde{\sigma}_t \mathbf{z}_t$

4 $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)$
$= \frac{1}{\sqrt{\bar{\alpha}_t}}\mathbf{V}\mathbf{\Sigma}\left(\sigma_y^2\mathbf{I} + \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}\mathbf{\Sigma}^2\right)^{-1}\mathbf{U}^T\left(\mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}}\mathbf{A}\mathbf{x}_t\right)$

5 $\mathbf{x}_{t-1} = \mathbf{x}_{t-1} + \lambda \frac{1-\alpha_t}{\sqrt{\alpha_t}} \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)$

Output: $\mathbf{x}_0$

Algorithm 2 DMPS (flow-based version)

Input: $\mathbf{y}$, $\mathbf{A}$, σ_y^2, $\Delta_t = 1/T$, λ

Initialization: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$

6 **for** $t = T$ ***to*** 1 **do**

7 $\mathbf{x}_{t-1} = \mathbf{x}_t - \mathrm{v}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\Delta_t$

8 $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)$
$= \frac{1}{a_t}\mathbf{V}\mathbf{\Sigma}\left(\sigma_y^2\mathbf{I} + \frac{b_t^2}{a_t^2}\mathbf{\Sigma}^2\right)^{-1}\mathbf{U}^T\left(\mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}}\mathbf{A}\mathbf{x}_t\right)$

9 $\mathbf{x}_{t-1} = \mathbf{x}_{t-1} - \lambda \frac{b_t(\dot{a}_t b_t - a_t \dot{b}_t)}{a_t} \log \tilde{p}(\mathbf{y}|\mathbf{x}_t)\Delta_t$

Output: $\mathbf{x}_0$

become

$$\text{DDPM:}\quad \mathbf{x}_{t-1} = \underbrace{\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} s_\theta(\mathbf{x}_t, t)\right) + \beta_t \mathbf{z}_t}_{\text{original sampling equation}} + \underbrace{\frac{1-\alpha_t}{\sqrt{\alpha_t}} \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)}_{\text{additional part}}, \tag{26}$$

$$\text{Flow-based:}\quad \mathbf{x}_{t-1} = \underbrace{\mathbf{x}_t - \mathbf{v}_\theta(\mathbf{x}_t, t)\Delta_t}_{\text{original sampling equation}} \underbrace{- \frac{b_t(\dot{a}_t b_t - a_t \dot{b}_t)}{a_t} \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)\Delta_t}_{\text{additional part}}, \tag{27}$$

where the blue part is the addition terms required to incorporate into the original sampling equations to enable posterior sampling from $p(\mathbf{x}_0|\mathbf{y})$ given $\mathbf{y}$.

Following the above procedures, we obtain the resultant algorithms for DDPM and flow-based models, as shown in Algorithm 1 and Algorithm 2, respectively. For brevity, we call both algorithms as Diffusion Model based Posterior Sampling (dubbed DMPS) since flow-based models can be viewed as a generalization of diffusion models Albergo et al. (2023). In the DDPM version, the reverse diffusion variance $\{\tilde{\sigma}_t\}_{t=1}^T$ is learned as the ADM in Dhariwal and Nichol (2021). Both the two versions of DMPS algorithms can be easily implemented on top of the existing code just by adding two additional simple lines (lines 4-5 in Algorithm 1, lines 8-9 in Algorithm 2) of codes.

Remark: A scaling parameter $\lambda > 0$ is introduced in both algorithms, similar to classifier guidance diffusion sampling Dhariwal and Nichol (2021). Empirically it is found that the performances are robust to different choices of λ as shown in the supplementary material, and we fix $\lambda = 1.75$ for DMPS (DDPM version) and $\lambda = 2.0$ for DMPS (flow-based version) in all the experiments.

4. Experiments

In this section, we conduct experiments on a variety of noisy linear inverse problems to demonstrate the efficacy of the proposed DMPS method, for both diffusion models and flow-based models. The code is available at https://github.com/mengxiangming/dmps.

4.1. Experimental Setup

Tasks: The tasks we consider include image super-resolution (SR), denoising, deblurring, as well as image colorization. In particular: (a) for image super-resolution (SR), the bicubic downsampling is performed as Chung et al. (2022a); (b) for deblurring, uniform blur of size 9×9 Kawar et al. (2022) (for DDPM) and Gaussian blur (for flow-based) are used; (c) for colorization, the grayscale image is obtained by averaging the red, green, and blue channels of each pixel Kawar et al. (2022). For all tasks, additive Gaussian noise $\mathbf{n}$ with $\sigma = 0.05$ is added except the denoising task where a larger noise $\mathbf{n}$ with $\sigma = 0.5$ is added.
Dataset: Both FFHQ Karras et al. (2019) and CelebA-HQ Karras et al. (2018) are considered. More results on FFHQ-cat, LSUN-bedroom, and AFHQ-cat can be found in the supplementary material.
Pre-trained Diffusion Models: For a fair comparison, we use the same pre-trained model for all the different methods evaluated. For diffusion models, the pre-trained ADM model Choi et al. (2021) is used, available in DDPM-checkpoint. For flow-based models, we use the pre-trained rectified flow model Liu et al. (2022), which is available in flow-checkpoint, and the forward process (3) is specified as $a_t = 1 - t$, $b_t = t$.
Comparison Methods: We compare DMPS with the following methods: DPS Chung et al. (2022a), PGDM Song et al. (2022), and the OT-ODE method Pokle et al. (2023). Actually, OT-ODE can be viewed as the flow-based version of PGDM. For DPS, we also compare two versions: one is the original DDPM version, the other is the flow-based version obtained following the procedures described in Section 3.2.
Metrics: Three widely used metrics are considered, including the standard distortion metric peak signal noise ratio (PSNR) (dB), as well as two popular perceptual metrics: structural similarity index measure (SSIM) Wang et al. (2004) and Learned Perceptual Image Patch Similarity (LPIPS) Zhang et al. (2018).
GPU: All results are run on a single NVIDIA Tesla V100.

	super-resolution			**deblur**			**colorization**			**denoising**		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DMPS (DDPM, ours)	**27.63**	**0.8450**	**0.2071**	**27.26**	0.7644	0.2222	**21.09**	**0.9592**	**0.2738**	**27.81**	0.8777	0.2435
DPS (DDPM)	26.78	0.8391	0.2329	26.50	**0.8151**	0.2248	11.53	0.7923	0.5755	27.22	**0.8969**	0.2428
PGDM	27.60	0.8345	0.2077	26.65	0.7458	**0.2196**	12.15	0.8920	0.3969	27.60	0.8682	**0.2425**

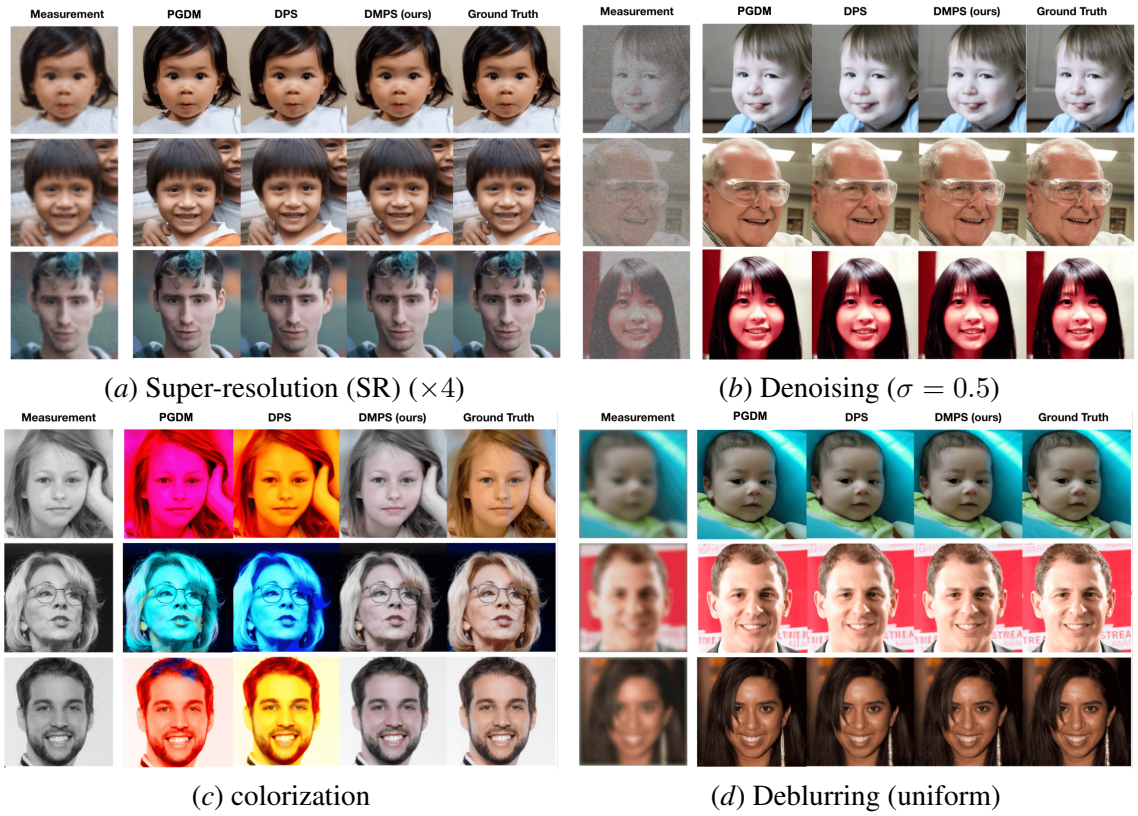
Table 1: Quantitative comparison (PSNR (dB), SSIM, LPIPS) of different algorithms for different tasks on FFHQ 256×256-1k validation dataset. The same pre-trained DDPM model is used.

Results: First is a quantitative comparison in terms of different metrics. Table 1 shows the quantitative reconstruction performances of different algorithms on diffusion models on the FFHQ dataset, and Table 2 shows the quantitative reconstruction performances of different algorithms on flow-based models with the CelebA-HQ dataset. As shown in Table 1 and Table 2, despite its simplicity, the proposed DMPS achieves highly competitive or even better performances than the baselines.

Second, we make a qualitative comparison between different algorithms for different tasks. Figure 2 shows the typical reconstructed images of different algorithms on diffusion models on the FFHQ dataset; Figure 1 shows the quantitative reconstruction performances of different algorithms

Method	super-resolution			deblur			colorization			denoising		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DMPS (Flow-based, ours)	**28.29**	**0.8011**	0.2329	**26.21**	**0.7235**	0.2637	**23.31**	**0.8861**	**0.2901**	**29.04**	**0.8166**	**0.2821**
DPS (Flow-based)	28.05	0.7754	**0.2266**	22.64	0.5787	0.3403	20.92	0.8061	0.3335	27.93	0.7465	0.2882
OT-ODE	27.71	0.7657	0.2302	25.84	0.7084	**0.2573**	21.67	0.8696	0.3094	22.76	0.3820	0.4778

Table 2: Quantitative comparison (PSNR (dB), SSIM, LPIPS) of different algorithms for different tasks on the validation set of CelebA-HQ. The same pre-trained flow-based model is used.



(*a*) Super-resolution (SR) ($\times 4$)

(*b*) Denoising ($\sigma = 0.5$)

(*c*) colorization

(*d*) Deblurring (uniform)

Figure 2: Typical results on FFHQ 256×256 1k validation set for different noisy linear inverse problems. All the algorithms are based on the same DDPM model. In all cases, the measurements are with Gaussian noise $\sigma = 0.05$, except denoising where $\sigma = 0.5$.

on flow-based models with the CelebA-HQ dataset. As shown in Figure 2 and Figure 1, in all tasks, DMPS produces high-quality realistic images which match details of the ground-truth more closely. For example, for super-resolution, please have a look at the ear stud in the first row of Figure2 (a), the hand on the shoulder in the second row of Figure 2 (a), and the background in the second row of Figure 2 (a); for denoising, please see the background door in the first row of Figure 2 (b), and the collar in the second row of Figure 2 (b), and the last row of Figure 1; for colorization, DPS tends to

Method	Inference Time [s]
DMPS (DDPM, ours)	**67.02**
DPS (DDPM)	194.42
PGDM	182.35

Method	Inference Time [s]
DMPS (flow-based, ours)	**4.45**
DPS (flow-based)	8.04
OT-DOE	6.44

Figure 3: Comparison of the inference time for different methods. Left: Results on DDPM models when NFE=1000, obtained on the SR task for FFHQ 256×256. Right: Results on flow-based models when NFE=50, obtained on the SR task for CelebA-HQ 256×256. All algorithms are run on a single NVIDIA Tesla V100.

produce over-bright images in colorization while DMPS produces more natural colored images, as shown in Figure 2 (c) and Figure 1, etc.

Finally, we evaluate the inference time of different algorithms, which is one of the key motivation of this paper. Here we would like to emphasize again that the main goal of this paper is not to compete with state-of-the-art performance but rather to provide a fast method. For fair of comparison, for both diffusion and flow-based models, different algorithms uses the same pre-trained model. Figure 3 show the the average running time for different algorithms: Left table shows the results under diffusion models when the number of function evaluation (NFE) is NFE = 1000; Right Table shows the results of different algorithms under flow-based models when NFE = 50. It can be seen that, in both versions, the inference time of the proposed DMPS method is significantly less than other methods, which is much appealing in practical applications.

5. Discussion and Conclusion

In this paper, we propose fast and effective closed-form approximation of the intractable noise-perturbed likelihood score, leading to the Diffusion Model based Posterior Sampling (dubbed DMPS). For both diffusion and flow-based models, we evaluate the effectiveness of DMPS on multiple linear inverse problems including image super-resolution, denoising, deblurring, colorization. Despite its simplicity, DMPS achieves highly competitive or even better reconstruction performances, while its inference time of DMPS is significantly faster.

Limitations & **Future Work**: While DMPS apparently reduces the inference time and achieves competitive reconstruction performances, it still suffers several limitations. First, although memory efficient SVD exists for most practical matrices $\mathbf{A}$ of practical interests Kawar et al. (2022), the SVD operation in DMPS still has some implementation difficulty for more general matrices $\mathbf{A}$. Second, it can not be directly applied to the popular latent diffusion models such as stable diffusion Rombach et al. (2022), which is widely used due to its efficiency. Addressing these limitations are left as future work.

Acknowledgements

This work was supported by NSFC No. 62306277, and the Fundamental Research Funds for the Zhejiang Provincial Universities Grant No. K20240090, The Japan Science and Technology Agency

(JST) Grant No. JPMJCR1912, and The Japan Society for the Promotion of Science (JSPS) Grant No. JP22H05117.

References

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A review of image denoising algorithms, with a new one. *Multiscale modeling & simulation*, 4(2):490–530, 2005.

Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14347–14356. IEEE, 2021.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022a.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *arXiv preprint arXiv:2206.00941*, 2022b.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Maryam Fazel, E Candes, Benjamin Recht, and P Parrilo. Compressed sensing and robust recovery of low rank matrices. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pages 1043–1047. IEEE, 2008.

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, 2021a.

Ajil Jalal, Sushrut Karmalkar, Alex Dimakis, and Eric Price. Instance-optimal compressed sensing via posterior sampling. In *International Conference on Machine Learning*, pages 4709–4720. PMLR, 2021b.

Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34:13242–13254, 2021.

Zahra Kadkhodaie and Eero P Simoncelli. Solving linear inverse problems using the prior implicit in a denoiser. *arXiv preprint arXiv:2007.13640*, 2020.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022.

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

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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. *arXiv preprint arXiv:2401.08740*, 2024.

Xiangming Meng and Yoshiyuki Kabashima. Quantized compressed sensing with score-based generative models. In *International Conference on Learning Representations*, 2023.

Xiangming Meng and Yoshiyuki Kabashima. Qcs-sgm+: Improved quantized compressed sensing with score-based generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14341–14349, 2024.

Finbarr O'Sullivan. A statistical perspective on ill-posed inverse problems. *Statistical science*, pages 502–518, 1986.

Ashwini Pokle, Matthew J Muckley, Ricky TQ Chen, and Brian Karrer. Training-free linear image inversion via flows. *arXiv preprint arXiv:2310.04432*, 2023.

Herbert E Robbins. An empirical bayes approach to statistics. In *Breakthroughs in Statistics: Foundations and basic theory*, pages 388–394. Springer, 1992.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2022.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Lu Yuan, Jian Sun, Long Quan, and Heungyeung Shum. Image deblurring with blurred/noisy image pairs. In *Proceedings of the 34th ACM SIGGRAPH Conference on Computer Graphics, 34th Annual Meeting of the Association for Computing Machinery's Special Interest Group on Graphics; San Diego, CA; United States*, 2007.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

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

Appendix A. Verification of Assumption 1

Here we provide a theoretical support of the uninformative prior assumption 1, or, equivalently, the following Gaussian approximation of the posterior $p(\mathbf{x}_0|\mathbf{x}_t)$:

$$p(\mathbf{x}_0|\mathbf{x}_t) \approx \mathcal{N}(\frac{\mathbf{x}_t}{a_t}, b_t^2\mathbf{I}), \tag{28}$$

Throughout the following derivations, we will drop any additive constants in the log (which translate to normalizing factors), and drop all terms of order $O(t)$.

Let us start with the original Bayes' formula (using the log form):

$$\log p(\mathbf{x}_0|\mathbf{x}_t) = \log p(\mathbf{x}_t|\mathbf{x}_0) + \log p_0(\mathbf{x}_0) - \log p_t(\mathbf{x}_t), \tag{29}$$

where $p_0(\mathbf{x}_0)$ and $p_t(\mathbf{x}_t)$ denote the marginal distribution of $\mathbf{x}_0$ and $\mathbf{x}_t$, respectively.

Since $p_{t-\Delta t}(\cdot) = p_t(\cdot) + \Delta t \frac{\partial}{\partial t} p_t(\cdot) + \mathcal{O}(\Delta t)$ for $|t| \ll 1$, there is

$$\log p_0(\mathbf{x}_0|\mathbf{x}_t) = \log p(\mathbf{x}_t|\mathbf{x}_0) + \log p_t(\mathbf{x}_0) + \mathcal{O}(t) - \log p_t(\mathbf{x}_t). \tag{30}$$

For (30), we perform a first order Taylor expansion of $\log p_t(\mathbf{x}_0)$ around $\mathbf{x}_t$, which yields

$$\begin{aligned}\log p_0(\mathbf{x}_0|\mathbf{x}_t) &= \log p(\mathbf{x}_t|\mathbf{x}_0) + \log p_t(\mathbf{x}_t) + \langle \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t), \mathbf{x}_0 - \mathbf{x}_t \rangle + \mathcal{O}(t) - \log p_t(\mathbf{x}_t) \\ &= \log p(\mathbf{x}_t|\mathbf{x}_0) + \langle \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t), \mathbf{x}_0 - \mathbf{x}_t \rangle + \mathcal{O}(t). \end{aligned} \tag{31}$$

Substituting $p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(a_t\mathbf{x}_0, b_t^2)$ and completing the squares, we obtain:

$$\log p(\mathbf{x}_0|\mathbf{x}_t) = -\frac{\|\mathbf{x}_t - a_t\mathbf{x}_0\|^2}{2b_t^2} + \langle \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t), \mathbf{x}_0 - \mathbf{x}_t \rangle + \mathcal{O}(t) \quad = -\frac{1}{2b_t^2}\|\mathbf{x}_0 - \boldsymbol{\mu}\|^2 + C, \tag{32}$$

where C is a constant value and the mean value $\boldsymbol{\mu}$ is:

$$\boldsymbol{\mu} = \frac{\mathbf{x}_t}{a_t} + \frac{b_t^2}{a_t^2}\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \tag{33}$$

Therefore, we obtain that the posterior distribution $p(\mathbf{x}_0|\mathbf{x}_t)$ can be approximated as a Gaussian

$$p(\mathbf{x}_0|\mathbf{x}_t) \approx \mathcal{N}(\frac{\mathbf{x}_t}{a_t} + \frac{b_t^2}{a_t^2}\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t), b_t^2\mathbf{I}) \tag{34}$$

Comparing eqs. (28) and (34), we can see that in our result (1), we further ignore the term $\frac{b_t^2}{a_t^2}\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ in the mean value. This is valid for sufficiently small t since the variance b_t^2 is sufficiently small following the special design principle in forward diffusion process. For example, for DDPM and flow-based model considered in our manuscript, $b_t^2 = 1 - \bar{\alpha}_t$, $b_t^2 = t^2$, respectively.

Reflecting on this derivation, the main idea is that for a sufficiently small t, the Bayes' rule expansion of $p(\mathbf{x}_0 \mid \mathbf{x}_t)$ (recall that this is what we need to compute the likelihood score) is primarily influenced by the term $p(\mathbf{x}_t \mid \mathbf{x}_0)$ from the forward process, regardless of the prior of $p(\mathbf{x}_0)$. As a result, the uninformative prior assumption is reasonable for sufficiently small t. In fact, this insight is exactly why in the diffusion models the reverse process and the forward process share the same functional form for sufficiently small time interval. It is worth pointing out that, the validity of the above results does not depend on the underlying distribution $p_0(\mathbf{x}_0)$, whether it being a simple Gaussian or a complex distribution as that of a face image.

A toy example: We further consider a toy example to illustrate this where the exact form of $p(\mathbf{x}_0|\mathbf{x}_t)$ in (12) can be computed exactly. Assume that $\mathbf{x}$ reduces to a scalar random variable x and the

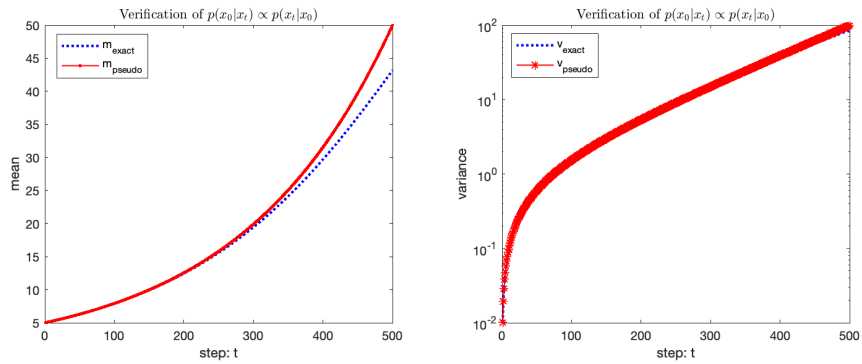


Figure 4: Comparison of the exact mean and variance of $p(x_0|x_t)$ with the pseudo mean and variance under the uninformative assumption, i.e., $p(x_0|x_t) \propto p(x_t|x_0)$.

associated prior $p(x)$ follows a Gaussian distribution, i.e., $p(x) = \mathcal{N}(x; 0, \sigma_0^2)$, where σ^2 is the prior variance. The likelihood $p(\mathbf{x}_t)|\mathbf{x}_0)$ (3) in this case is simply $p(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t))$.

Then, from (12), after some algebra, it can be computed that the posterior distribution $p(x_0|x_t)$ is

$$p(x_0|x_t) = \mathcal{N}(x_0; m_{\text{exact}}, v_{\text{exact}}) \tag{35}$$

where

$$m_{\text{exact}} = \frac{\sqrt{\bar{\alpha}_t}\sigma_0^2}{(1-\bar{\alpha}_t) + \bar{\alpha}_t\sigma_0^2}x_t, \;\; v_{\text{exact}} = \frac{(1-\bar{\alpha}_t)\sigma_0^2}{(1-\bar{\alpha}_t) + \bar{\alpha}_t\sigma_0^2}. \tag{36}$$

Under the Assumption 1, i.e., $p(x_0|x_t) \propto p(x_t|x_0)$, we obtain an approximation of $p(x|x_t)$ as follows

$$p(x_0|x_t) \simeq \tilde{p}(x_0|x_t) = \mathcal{N}(x_0; m_{\text{pseudo}}, v_{\text{pseudo}}), \tag{37}$$

where

$$m_{\text{pseudo}} = \frac{1}{\sqrt{\bar{\alpha}_t}}x_t, \;\; v_{\text{pseudo}} = \frac{1-\bar{\alpha}_t}{\bar{\alpha}}. \tag{38}$$

By comparing the exact result (36) and approximation result (38), it can be easily seen that for a fixed $\sigma_0^2 > 0$, as $\bar{\alpha}_t \to 1$, we have $m_{\text{pseudo}} \to m_{\text{post}}$ and $v_{\text{pseudo}} \to v_{\text{post}}$, which is exactly the case for DDPM as $t \to 1$. To see this, we anneal $\bar{\alpha}_t$ as $\bar{\alpha}_t = \bar{\alpha}_{\max}(\frac{\bar{\alpha}_{\min}}{\bar{\alpha}_{\max}})^{\frac{t-1}{T-1}}$ geometrically and compare $m_{\text{pseudo}}, v_{\text{pseudo}}$ with $m_{\text{exact}}, v_{\text{exact}}$ as t increase from 1 to T. Assume that $\bar{\alpha}_{\min} = 0.01$ and $\bar{\alpha}_{\min} = 0.99$, and $\sigma_0 = 25, x_t = 5, T = 500$, we obtain the results in Fig. 4. It can be seen in Fig. 4 that the approximated values $m_{\text{pseudo}}, v_{\text{pseudo}}$, especially the variance v_{pseudo}, approach to the exact values $m_{\text{exact}}, v_{\text{exact}}$ very quickly, verifying the effectiveness of the Assumption 1 under this toy example.